

Федеральное государственное бюджетное учреждение науки Институт проблем управления  
им. В.А. Трапезникова Российской академии наук «ИПУ РАН»



На правах рукописи

Ларионов Андрей Алексеевич

**Технология построения и методы исследования систем управления  
безопасностью дорожного движения на основе широкополосных  
беспроводных сетей и радиочастотной идентификации**

Специальность 05.13.15 —

«Вычислительные машины, комплексы и компьютерные сети»

Диссертация на соискание учёной степени  
кандидата технических наук

Научный руководитель:  
доктор технических наук, профессор  
Вишневецкий Владимир Миронович

Москва — 2021

## Оглавление

	Стр.
Введение . . . . .	6
<b>Глава 1. Распределенная система радиочастотной идентификации транспорта . . . . .</b>	<b>12</b>
1.1 Введение . . . . .	12
1.2 Структура системы радиочастотной идентификации транспорта . . . . .	13
1.3 Постановка задач исследования . . . . .	14
1.3.1 Исследование эффективности радиочастотной идентификации мобильных меток . . . . .	15
1.3.2 Анализ производительности опорной сети . . . . .	16
1.3.3 Экспериментальная реализация системы . . . . .	17
1.4 Радиочастотная идентификация транспортных средств . . . . .	18
1.4.1 Стандарт EPC Class 1 Generation 2 . . . . .	19
1.4.2 Обзор исследований производительности UHF RFID . . . . .	25
1.5 Передача данных по беспроводным сетям . . . . .	29
1.5.1 Механизм доступа к каналу IEEE 802.11 DCF . . . . .	29
1.5.2 Обзор исследований производительности беспроводных каналов связи . . . . .	30
1.6 Методы исследования задержек в многошаговых беспроводных сетях . . . . .	32
1.6.1 Открытые сети массового обслуживания с марковскими потоками заявок и обслуживанием фазового типа . . . . .	34
1.6.2 Методы восстановления РН-распределений и MAP-потоков . . . . .	36
1.7 Методы построения распределённых систем радиочастотной идентификации . . . . .	38
1.8 Заключение . . . . .	41
<b>Глава 2. Исследование производительности систем радиочастотной идентификации . . . . .</b>	<b>42</b>
2.1 Структура системы радиочастотной идентификации . . . . .	42
2.2 Общая схема расчёта вероятности идентификации автомобилей . . . . .	43
2.3 Анализ влияния параметров протокола на длительности раундов . . . . .	45
2.4 Моделирование радиоканала между считывателем и меткой . . . . .	48
2.4.1 Расчёт бюджета соединений . . . . .	48
2.4.2 Расчёт мощности принятых сигналов . . . . .	50
2.4.3 Расчёт вероятности битовой ошибки (BER) . . . . .	54
2.5 Результаты имитационного моделирования . . . . .	56
2.5.1 Анализ влияния частоты переключений антенн на число раундов . . . . .	57
2.5.2 Анализ вероятности идентификации транспортных средств . . . . .	58
2.6 Заключение . . . . .	62

<b>Глава 3. Аналитическая модель системы радиочастотной идентификации автомобилей</b>	<b>63</b>
3.1 Ограничения и допущения	64
3.1.1 Модельный считыватель	64
3.1.2 Модельные метки	65
3.2 Постановка задачи	66
3.3 Моделирование раундов инвентаризации	67
3.4 Вычисление оценки длительностей раундов	73
3.4.1 Размеченные сценарии и элементарные операции	73
3.4.2 Матрицы элементарных операций	76
3.4.3 Построение операций по размеченному сценарию	78
3.4.4 Расчет распределения числа активных меток	81
3.4.5 Итерационный расчет оценок длительностей раундов	82
3.5 Расчет вероятности идентификации	84
3.5.1 Определение основного процесса	85
3.5.2 Переходные вероятности основного процесса для элементарных операций	87
3.5.3 Матрицы переходных вероятностей между раундами для основного процесса	90
3.5.4 Расчет вероятности поглощения процесса	90
3.6 Результаты моделирования	92
3.6.1 Параметры модели	92
3.6.2 Анализ свойств раундов	94
3.6.3 Расчет вероятности идентификации	96
3.6.4 Исследование ошибок в оценке вероятности идентификации	98
3.7 Заключение	100
<b>Глава 4. Анализ производительности опорной беспроводной сети</b>	<b>101</b>
4.1 Моделирование многошаговой беспроводной сети с помощью тандемной сети массового обслуживания	102
4.2 Открытая тандемная сеть массового обслуживания с узлами MAP/PN/1/M	103
4.2.1 Свойства PN-распределений и MAP-потоков	103
4.2.2 Свойства системы MAP/PN/1/M	105
4.2.3 Точный расчет характеристик сети массового обслуживания	108
4.3 Расчет характеристик сети массового обслуживания методом Монте-Карло	112
4.3.1 Обработка событий	113
4.3.2 Расчет характеристик	114
4.3.3 Программная реализация	114
4.3.4 Преимущества и недостатки метода	115
4.4 Расчет характеристик сети методом понижения размерности выходящих потоков	116

	Стр.
4.4.1	Аппроксимация по среднему . . . . . 117
4.4.2	Аппроксимация по двум моментам . . . . . 118
4.4.3	Аппроксимация по трем моментам . . . . . 119
4.4.4	Сложность и применимость метода . . . . . 124
4.5	Моделирование задержки в канале . . . . . 125
4.6	Численное исследование эффективности метода аппроксимации потоков . . . 127
4.6.1	Построение набора данных для исследования . . . . . 127
4.6.2	Использование методов аппроксимации выходящего потока . . . . . 130
4.7	Численный расчет характеристик многошаговой беспроводной сети . . . . . 138
4.7.1	Характеристики каналов калибровочной сети . . . . . 138
4.7.2	Расчет межконцевых задержек в сети произвольного размера . . . . . 140
4.8	Заключение . . . . . 144
<b>Глава 5. Разработка и экспериментальное внедрение системы</b>	
	<b>радиочастотной идентификации . . . . . 145</b>
5.1	Архитектура системы управления считывателями . . . . . 145
5.2	Протоколы взаимодействия компонентов системы . . . . . 148
5.2.1	IMMP — протокол управления модулями . . . . . 149
5.2.2	SUAR — протокол подключения интерфейсов управления . . . . . 150
5.2.3	ITOP — протокол работы с RFID-адаптерами . . . . . 152
5.2.4	TFP — протокол потокового чтения меток . . . . . 155
5.3	Основные компоненты системы . . . . . 156
5.3.1	Супервизор SVR . . . . . 156
5.3.2	RFID-адаптер G2RD . . . . . 159
5.3.3	Приложение TFPD и клиент TFPC . . . . . 160
5.3.4	Утилита UAX и интерфейсы управления . . . . . 161
5.4	Реализация RFID-считывателей и системы управления . . . . . 162
5.4.1	Аппаратная реализация считывателей . . . . . 162
5.4.2	Особенности программной реализации . . . . . 163
5.5	Экспериментальные испытания . . . . . 164
5.5.1	Испытания в Казани в 2014 году . . . . . 165
5.5.2	Испытания в Казани в 2020 году . . . . . 168
5.5.3	Испытания на платном участке ЦКАД в 2021 году . . . . . 169
5.6	Заключение . . . . . 170
<b>Заключение . . . . . 171</b>	
<b>Список сокращений и условных обозначений . . . . . 172</b>	
<b>Словарь терминов . . . . . 174</b>	

	Стр.
Список литературы . . . . .	175
Список рисунков . . . . .	193
Список таблиц . . . . .	197
Приложение А. Акты о внедрении . . . . .	198

## Введение

Системы идентификации автомобилей и регистрации нарушений правил дорожного движения играют важную роль в управлении безопасностью на дорогах, способствуют снижению смертности и большого материального ущерба при авариях. Существующие системы основаны на видео- или фото-идентификации номеров транспортных средств (ТС). Работе этих систем могут препятствовать загрязнение при плохих погодных условиях или намеренное скрытие номерного знака. Из-за этого, по сведениям ГИБДД, эффективность систем идентификации автомобилей может падать ниже 50 %. Это определяет актуальность тематики диссертации, направленной на исследование и создание новых распределенных компьютерных систем безопасности на дорогах, использующих альтернативные источники идентификации транспортных средств, а именно — технологию радиочастотной идентификации (RFID). Постановлением Правительства РФ предусмотрено создание таких систем в 2022 – 2024 гг. в пилотных зонах в Москве, Санкт-Петербурге и Казани.

При использовании RFID-технологии радиометка размещается в номерном знаке или под лобовым стеклом автомобиля. Находясь в зоне действия RFID-считывателя, она передает ему свой уникальный идентификатор. В пассивных системах RFID УВЧ-диапазона 860 – 960 МГц метки не обладают собственными источниками энергии и могут быть идентифицированы считывателями на расстоянии порядка 10 – 20 метров. Эффективность радиочастотной идентификации существенно зависит как от особенностей окружения, распространения радиосигналов и характеристик антенн, так и от параметров протокола международного стандарта EPC Class 1 Generation 2 (EPC Gen2), используемого для связи между считывателем и метками. Хотя анализу производительности систем радиочастотной идентификации в различных приложениях посвящено значительное количество работ, тематика применения RFID для идентификации автомобилей остается слабо изученной. В диссертации предложены и исследованы новые математические и имитационные модели системы радиочастотной идентификации ТС, учитывающие как параметры распространения радиосигналов, так и всевозможные настройки RFID-считывателей и параметры протокола.

Для работы автоматической компьютерной системы безопасности необходимо соединить точки идентификации ТС с центрами обработки данных компьютерными сетями, по которым оперативно передается информация о распознанных номерах автомобилей. Проводные решения оказываются не всегда доступными по техническим или экономическим причинам, поэтому особый интерес представляют широкополосные беспроводные сети. Для анализа производительности беспроводных сетей часто применяются стохастические методы, но из-за высокой сложности их зачастую трудно применять для анализа больших многошаговых сетей. Одним из наиболее распространённых методов исследований являются сети массового обслуживания (СМО), в которых каналы связи и маршрутизаторы моделируются обслуживающими приборами с очередями. Хорошую точность при описании реальных систем показывают СМО с марковским входящими потоками (МАР) и распределением времени обслуживания фазового типа (РН). Хотя системы МАР/РН/1/N хорошо

исследованы, их применение для построения открытых сетей массового обслуживания, моделирующих реальные беспроводные сети, мало изучено в современной литературе. Из-за высокой сложности моделей значительный интерес представляют методы быстрого получения приближенных оценок их характеристик, разработанные в настоящей работе.

Системы управления RFID-считывателями, используемые в системах безопасности на автодорогах, обладают рядом особенностей. Во-первых, они должны обеспечивать непрерывный сбор данных с большого количества RFID-считывателей в реальном времени. Во-вторых, необходимо объединять данные о прочитанных RFID-метках с данными от камер видеофиксации, радарными и другими датчиками. Наконец, для различных служб и ведомств система должна предоставлять разные уровни доступа. Перечисленные особенности обуславливают необходимость разработки новой распределенной системы управления считывателями, предложенной в диссертации. Таким образом, высокая практическая потребность в разработке новых систем управления безопасностью дорожного движения на основе RFID-технологии и широкополосной беспроводной связи и особенности их реализации определяют актуальность и новизну диссертационной работы.

Исследованиям в области проектирования, разработки и применения технологии RFID, а также исследованиям широкополосных беспроводных сетей с использованием методов теории массового обслуживания, посвящен ряд работ, среди которых следует особо отметить работы отечественных и зарубежных учёных: В.М. Вишневский, А.Н. Дудин, П.В. Никитин, Е.А. Кучерявый, Р.В. Киричек, К.Е. Самуйлов, Ю.В. Гайдамака, В.В. Рыков, Р.Н. Минниханов, D. Lucantoni, N. Abramson, L.G. Roberts, M.F. Neuts, G. Horvath, G. Bianchi, H. Okamura, P. Buchholz, M. Telek, J.C. Strelen, L. Bodrog, D. Aldous, K.V.S. Rao, C. Floerkemeier, C. Wang, E. Vahedi, R. Nelson, T.S. Rappaport, S. Singh, J. Kriege, P. Djukic, S. Valaee, J.E. Hoag. и др.

**Объектом исследования** являются системы радиочастотной идентификации транспорта и компьютерные сети в их составе.

**Предметом исследования** являются методы анализа и алгоритмы расчета характеристик производительности систем радиочастотной идентификации, а также методы построения и исследования компьютерных сетей и систем управления распределенными системами идентификации транспорта.

**Целью** данной работы является создание комплекса моделей для анализа производительности систем радиочастотной идентификации ТС и широкополосных беспроводных сетей, а также разработка и экспериментальное внедрение системы управления, сбора и передачи данных с RFID-считывателей в центр обработки данных. В диссертационной работе сформулированы и решены следующие **задачи**:

1. Разработка и исследование комплекса аналитических и имитационных моделей для анализа и оптимизации основных характеристик систем радиочастотной идентификации транспортных средств.
2. Разработка методики оценки производительности широкополосных беспроводных сетей, использующихся для передачи данных от RFID-считывателей в центры обработки данных, на основе методов теории массового обслуживания и марковских случайных процессов.

3. Разработка архитектуры и реализация распределенной компьютерной системы управления и сбора данных с RFID-считывателей, ее экспериментальное внедрение и проведение испытаний.

**Научная новизна:**

1. Впервые предложена и исследована стохастическая модель системы радиочастотной идентификации ТС, учитывающая скорость движения RFID-меток, расположенных на номерных знаках автомобилей, а также различные сценарии проведения циклического опроса и сбора данных с меток.
2. Разработан комплекс новых аналитических и имитационных моделей для анализа вероятности идентификации ТС, учитывающих особенности логического и физического уровней протокола стандарта EPC Class 1 Gen.2, и особенности распространения радиосигналов между RFID-меткой и считывателем.
3. Предложена новая методика моделирования многошаговых беспроводных сетей с помощью тандемных сетей массового обслуживания, учитывающая особенности трафика и интерференции в каналах связи.
4. Разработан оригинальный метод вычисления оценок характеристик многофазных систем массового обслуживания большой размерности с коррелированными входными потоками и распределениями обслуживания фазового типа.
5. Разработана архитектура и реализована новая распределенная компьютерная система управления RFID-считывателями, предназначенная для организации сбора данных об идентифицированных транспортных средствах.
6. Проведена обработка экспериментальных данных, полученных при опытных внедрениях разработанной распределенной компьютерной системы радиочастотной идентификации на автодорогах в г. Казань и на ЦКАД в Московской области, показавшая высокое совпадение с теоретическими результатами диссертации.

**Область исследования.** Диссертационная работа соответствует содержанию специальности 05.13.15, а именно следующим пунктам специальности:

- Разработка научных основ создания вычислительных машин, комплексов и компьютерных сетей, исследования общих свойств и принципов функционирования вычислительных машин, комплексов и компьютерных сетей.
- Теоретический анализ и экспериментальное исследование функционирования вычислительных машин, комплексов и компьютерных сетей с целью улучшения их технико-экономических и эксплуатационных характеристик.
- Разработка научных методов и алгоритмов организации параллельной и распределенной обработки информации, многопроцессорных, многомашинных и специальных вычислительных систем.
- Разработка научных методов и алгоритмов создания структур и топологий компьютерных сетей, сетевых протоколов и служб передачи данных в компьютерных сетях, взаимодействия компьютерных сетей, построенных с использованием различных телекоммуникационных технологий, мобильных и специальных компьютерных сетей, защиты компьютерных сетей и приложений.



**Практическая значимость.** Аналитические и имитационные модели и методы, предложенные в диссертации, могут эффективно использоваться для оценки производительности систем радиочастотной идентификации автомобилей и широкополосных беспроводных сетей. Распределенная система управления считывателями и программное обеспечение, описанные в работе, использовались в трех экспериментальных внедрениях на автодорогах в г. Казань и Московской области. Практическая значимость диссертационной работы подтверждается актами о внедрении, полученными от Государственного бюджетного учреждения «Безопасность дорожного движения» (г. Казань) и ПАО «Микрон».

Результаты работы также были использованы в исследованиях, проводимых по следующим грантам:

- Контракт с Министерством образования и науки РФ № 14.514.11.4071 в рамках федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы».
- Соглашение с Министерством образования и науки РФ о предоставлении субсидии от 22.10.2014 г. № 14.613.21.0020 в рамках федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса на 2014-2020 годы».
- Грант Российского научного фонда (РНФ) № 16-49-02021.
- Грант Российского фонда фундаментальных исследований (РФФИ) № 13-07-00737.
- Грант Российского фонда фундаментальных исследований (международный проект РФФИ — БРФФИ) № 14-07-90015.
- Грант Российского фонда фундаментальных исследований (международный проект РФФИ — БРФФИ) № 16-57-00130.

**Методология и методы исследования.** Для решения задач, поставленных в диссертации, использовались методы теории вероятностей, математической статистики, теории случайных процессов, теории массового обслуживания, методы дискретно-событийного имитационного моделирования. При разработке программного обеспечения использовались методы многопоточного программирования, методы разработки распределенных систем.

**Основные положения, выносимые на защиту:**

1. Стохастическая модель системы радиочастотной идентификации ТС, учитывающая скорость движения RFID-меток, расположенных на номерных знаках автомобилей, а также различные сценарии проведения циклического опроса и сбора данных с меток.
2. Новый комплекс аналитических и имитационных моделей для анализа вероятности идентификации ТС, учитывающих особенности логического и физического уровней протокола стандарта EPC Class 1 Gen.2, и особенности распространения радиосигналов между RFID-меткой и считывателем.
3. Новая методика моделирования многошаговых беспроводных сетей с помощью тандемных сетей массового обслуживания, учитывающая особенности трафика и интерференции в каналах связи.

4. Метод вычисления оценок характеристик многофазных систем массового обслуживания большой размерности с коррелированными входными потоками и распределениями обслуживания фазового типа.
5. Архитектура и реализация новой распределенной компьютерной системы управления RFID-считывателями, предназначенная для организации сбора данных об идентифицированных транспортных средствах.
6. Результаты экспериментальных внедрений разработанной распределенной компьютерной системы радиочастотной идентификации транспортных средств на автодорогах в г. Казань и в Московской области.

**Достоверность** полученных результатов обеспечивается использованием строгих математических моделей, сравнением результатов аналитического и имитационного моделирования. Результаты анализа вероятности идентификации автомобилей согласуются с результатами, полученными в ходе реальных экспериментов.

**Апробация работы.** Основные результаты работы докладывались на следующих конференциях: 12th Annual IEEE International conference on RFID 2018 (IEEE RFID 2018; США, Орландо); 11th Annual IEEE International conference on RFID 2017 (IEEE RFID 2017; США, Финикс); международный форум Kazan Digital Week 2020 (Казань); International conference on Advances in Applied Probability and Stochastic Processes 2019 (ICAAP & SP 2019; Индия, Коттаям); 13-е и 12-е Всероссийские совещания по проблемам управления (ВСПУ 2019, ВСПУ 2014; Москва, ИПУ РАН); 20-я, 18-я, 16-я и 15-я Международные конференции им. А.Ф. Терпугова Информационные технологии и математическое моделирование (ИТММ 2021, ИТММ 2019, ИТММ 2017, ИТММ 2016; Россия); 11th IEEE International Conference on Application of Information and Communication Technologies (IEEE AICT 2017; Москва, ИПУ РАН); 21-я, 20-я, 19-я, 18-я и 17-я Международные Конференции «Распределенные компьютерные и телекоммуникационные сети: управление, вычисление, связь» (DCCN 2018, DCCN 2017, DCCN 2016, DCCN 2015, DCCN 2013; Москва); 7th International Workshop on Communication Technologies for Vehicles (Nets4Cars-Fall 2014; Санкт-Петербург); 2012 International Conference on RFID-Technology and Applications (IEEE RFID-TA 2012; Франция, Ницца); 10-я и 5-я Всероссийская конференция «Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем» (ИТТММ 2020, ИТТММ 2015; Москва, РУДН).

**Личный вклад.** Основные результаты диссертационной работы получены автором самостоятельно. В работах [186–189, 192, 199–205, 209, 210] автору принадлежит участие в постановке задач, разработка методов и алгоритмов и реализация комплекса программ. В работах [194–198] — обзор литературы и разработка численных алгоритмов. Работа [193] выполнена без соавторов.

**Публикации.** Основные результаты по теме диссертации изложены в 20 печатных изданиях, 4 из которых изданы в журналах, рекомендованных ВАК, 15 — в изданиях, индексируемых Web of Science и Scopus, 1 — в издании, индексируемом РИНЦ.

**Объем и структура работы.** Диссертация состоит из введения, 5 глав, заключения и 1 приложения. Полный объем диссертации составляет 200 страниц, включая 85 рисунков и 8 таблиц. Список литературы содержит 210 наименований.

## Глава 1. Распределенная система радиочастотной идентификации транспорта

### 1.1 Введение

Системы радиочастотной идентификации транспорта представляют реальную альтернативу распространённым системам, построенным на базе фотокамер. В таких системах транспорт оснащается радиометками с уникальными идентификаторами. Эти идентификаторы считываются специальными устройствами (считывателями), расположенными в точках контроля. Данные о прочитанных метках передаются в центры обработки по каналам связи. Такие распределённые системы могут иметь множество назначений, например — сбор платы за проезд, идентификация нарушителей правил дорожного движения, контроль пересечения транспортом заданного периметра.

Ключевыми преимуществами систем радиочастотной идентификации перед традиционными системами с фотоидентификацией транспорта является более низкая стоимость, а также то, что надёжность чтения меток не зависит от погодных условий, которые могут существенно осложнять работу камер. Кроме того, передача данных о прочитанных метках требует гораздо меньшей пропускной способности каналов связи, нежели передача фотографий или видеоснимков. Наконец, системы радиочастотной идентификации могут применяться как сами по себе, так и дополнять существующие системы фотоидентификации, повышая достоверность при работе в плохих погодных условиях, а также предоставлять дополнительные функции.

При построении распределённых систем радиочастотной идентификации транспорта необходимо решать множество задач. Во-первых, необходимо обеспечить высокую вероятность успешного чтения меток на подвижных объектах. Во-вторых, для многих приложений критична малая задержка в передаче данных с точек идентификации в центры обработки данных. Наконец, определённую техническую сложность представляет реализация самой системы, включая разработку систем управления считывателями, протоколов передачи данных и решение прочих прикладных задач.

В настоящей главе будет рассмотрена структура распределённой системы радиочастотной идентификации транспорта, будут описаны её основные компоненты. Далее будут поставлены задачи, которые решены в диссертационном исследовании. Затем будут кратко описаны технологии радиочастотной идентификации и беспроводной передачи данных, и будет приведён обзор исследований по решаемым в диссертации проблемам.

## 1.2 Структура системы радиочастотной идентификации транспорта

Распределённая система радиочастотной идентификации транспорта (см. рис. 1.1) предназначена для определения номеров движущихся транспортных средств и передачи данных о них в центр обработки данных в режиме реального времени.

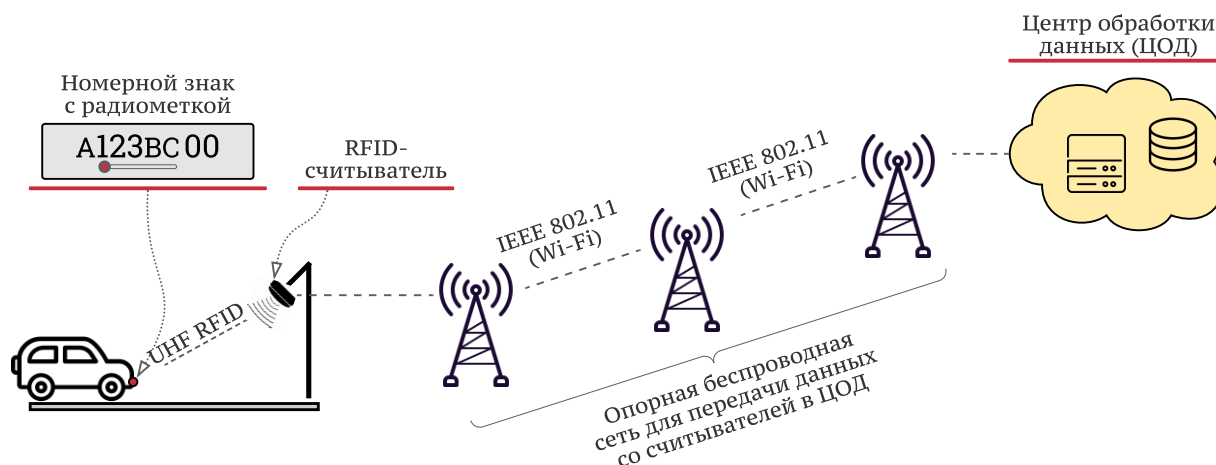


Рисунок 1.1 — Схема распределённой системы радиочастотной идентификации транспорта

Основными компонентами простейшей распределенной системы радиочастотной идентификации транспорта являются RFID-метки, RFID-считыватели, телекоммуникационная сеть и центр обработки данных. **RFID-метки** или **транспондеры** — это устройства, которыми оснащаются транспортные средства. Их назначение — передача записанного идентификатора считывателям. В зависимости от используемой технологии, могут быть активными или пассивными, то есть иметь или нет свой источник питания. Подробнее принцип работы меток стандарта EPC Class 1 Gen. 2 (ISO 18000-6C) описан в разделе 1.4. **RFID-считыватели** — активные устройства, осуществляющие чтение идентификаторов меток и их передачу в центр обработки данных. **Телекоммуникационная сеть** используется для передачи данных о метках от считывателей в центр обработки данных, а также для доступа администраторов к считывателям для их настройки, обслуживания и мониторинга. **Центр обработки данных** включает информационную систему, в которой собираются данные о прочитанных метках и состоянии работы считывателей.

В качестве примера можно привести системы для бесконтактного сбора оплаты проезда на автодорогах, например — на участках трасс М-4 «Дон», М-1 «Беларусь» и Центральной кольцевой автодороге (ЦКАД). В некоторых странах (например, в Чили, ЮАР, Азербайджане) проводились эксперименты по более масштабному внедрению RFID, для регистрации всех или определенной группы автомобилей. Подобный эксперимент проводился и в России, в республике Татарстан, его результаты будут более подробно рассмотрены в настоящей работе.

Для получения данных о проезжающем транспортном средстве оно должно быть оснащено одной или несколькими RFID-метками, причем выбранная технология радиочастотной

идентификации должна обеспечивать чтение метки на расстоянии 10–20 метров. Одной из наиболее подходящих технологий RFID, отвечающих этому требованию, является СВЧ-RFID стандарта EPC Class 1 Generation 2 [1], работающая в диапазоне 860–920 МГц. В частности, эта технология была использована в эксперименте, проведенном в городе Казань. В качестве альтернативы также можно использовать активные метки, или технологию DSRC, которая используется на уже упомянутых платных дорогах М-1, М-4 и ЦКАД.

Для передачи информации о считанных метках RFID-считыватели должны быть подключены к центрам обработки данных. Возможны три пути решения этой задачи: подключение к существующей проводной или беспроводной сети, коммутационное оборудование которой находится вблизи точек размещения считывателей; использование сотовой сети; построение отдельной сети для подключения считывателей. Использование существующих сетей оказывается не всегда возможным, особенно при развертывании системы за пределами крупных населённых пунктов. Кроме того, сети могут быть перегружены (например, если они уже используются для передачи данных с городских камер видеонаблюдения). Из-за ограниченности покрытия использование сотовой сети также не всегда возможно. В качестве примера, в настоящем исследовании будем рассматривать построение отдельной беспроводной сети. Как будет показано, для работы системы не нужно высокоскоростных каналов, и для подключения большого количества считывателей (порядка нескольких сотен) достаточно многошаговой сети, построенной на основе технологии IEEE 802.11g, оборудование которой очень недорогое и позволяет строить беспроводные соединения на расстояниях до нескольких десятков километров.

В центре обработки данных должна собираться как информация о зарегистрированных считывателями метках, так и о состоянии работы оборудования. Если в состав системы входят камеры, радары и прочее оборудование, в информационном центре можно объединять данные, полученные от различных источников. Например, объединяя результаты идентификации номерных знаков по фото с данными, полученными от считывателей, можно существенно повысить достоверность распознавания транспортного средства в условиях плохой видимости. Возможна интеграция и с системами приема оплаты или с базами розыска угнанных машин.

В ходе исследования, результаты которого приводятся в настоящей диссертационной работе, предполагалось, что для реализации распределённой системы радиочастотной идентификации транспорта используются технологии, описанные в табл. 1.

### 1.3 Постановка задач исследования

При проектировании и построении распределённой системы радиочастотной идентификации транспорта возникает целый ряд задач, относящихся как непосредственно к идентификации транспорта, так и к организации связи между считывателями и центром

Таблица 1 — Технологии, используемые в распределённой системе радиочастотной идентификации транспорта, разработанной в рамках диссертационной работы.

Компонент	Технология	Комментарии
Метки	EPC Class 1 Gen. 2	Пассивные метки в номерных знаках или на стикерах под лобовым стеклом.
Считыватели	EPC Class 1 Gen. 2	RFID-считыватели устанавливаются над дорогой, поддерживают до четырех антенн и подключаются к сети.
Сеть	IEEE 802.11	Многошаговая беспроводная сеть с каналами, в которых используется схема доступа CSMA/CA.
ЦОД		Различное программное обеспечение для получения, сохранения и анализа данных о метках.

обработки данных. Рассмотрим подробнее задачи, решению которых посвящена диссертационная работа.

### 1.3.1 Исследование эффективности радиочастотной идентификации мобильных меток

Критерием работоспособности системы является процент успешно распознанных транспортных средств. Задача осложняется тем, что метки могут двигаться с высокой скоростью (порядка 100–150 км/ч), а время, доступное для их чтения, крайне ограничено, так как считыватель может получить данные от метки лишь на небольшом расстоянии, порядка десяти метров. При различных настройках протокола EPC Class 1 Gen. 2 скорость обмена данными с меткой и вероятность успешной передачи сообщений могут меняться в очень широких пределах. Дополнительную сложность задаче придают многолучевое распространения сигналов между считывателем и меткой (как минимум, присутствие отраженного от дороги луча), наличие эффекта Доплера, а также использование метода обратного рассеяния для передачи ответов меток. Для решения задачи оценки работы системы радиочастотной идентификации удобно использовать методы имитационного моделирования, позволяющие учесть как параметры протокола, так и особенности распространения сигналов.

Отдельные аспекты работы считывателей можно исследовать с помощью методов теории случайных процессов. Хотя такой подход затрудняет учет всего объема факторов, влияющих на производительность системы, он позволяет более детально изучить отдельные закономерности, в частности — влияние периодических отключений питания и изменений

подмножеств опрашиваемых меток на вероятность успешной идентификации. Для построения аналитической модели протокола нужно формально описать компоненты, определяющие состояние системы, а также формализовать операции, которые считыватель может осуществлять над метками. После этого можно исследовать свойства этих операций и найти способы получения оценки успешного чтения меток.

Таким образом, для исследования эффективности системы радиочастотной идентификации автомобилей нужно решить следующие задачи:

1. Выявить факторы, влияющие на вероятность успешной идентификации мобильной RFID-метки.
2. Построить формальную математическую модель системы радиочастотной идентификации, исследовать свойства операций, осуществляемых считывателем над множеством мобильных меток.
3. Разработать аналитические и имитационные модели для получения оценки вероятности идентификации мобильных RFID-меток.
4. Определить параметры протокола, при которых доля успешно идентифицированных автомобилей с RFID-метками оказывается не ниже 90%.

Решению этих задач посвящены главы 2 и 3 диссертации.

### 1.3.2 Анализ производительности опорной сети

Для работы некоторых приложений важно обеспечить быструю передачу информации о прочитанных метках в центр обработки данных. Например, это необходимо для реализации бесконтактной оплаты проезда или для поиска угнанных транспортных средств. Для анализа межконцевых задержек можно использовать как имитационное моделирование, так и аналитические модели, построенные на базе теории массового обслуживания.

Модели открытых тандемных сетей массового обслуживания хорошо известны и исследованы. Основная сложность их применения к задаче поиска межконцевых задержек пакетов, передаваемых в многошаговых сетях, заключается в выборе адекватных распределений интервалов между поступлениями пакетов в сеть и распределений длительности передачи пакетов по каналам связи.

В диссертационном исследовании предполагается, что опорная сеть строится на базе беспроводных линий связи с каналами CSMA/CA. Для моделирования каналов связи используются системы массового обслуживания MAP/PН/1/N, в которых интервалы между поступлениями пакетов в сеть моделируются марковскими случайными потоками (MAP), время обслуживания пакетов имеет распределение фазового типа (PH), а размер очереди ограничен. Получение численных характеристик открытых сетей  $MAP/PН/1/N \rightarrow \bullet/PН/1/N \rightarrow \dots \rightarrow \bullet/PН/1/N$  осложняется экспоненциальным ростом пространства состояний при увеличении числа узлов в сети, и отдельной задачей является поиск эффективных способов получения численных оценок таких сетей.



Для оценки производительности опорной сети в диссертации решаются следующие задачи:

1. Поиск РН-распределений, адекватно моделирующих время обслуживания в беспроводных сетях с каналами CSMA/CA.
2. Оценка возможности использования метода аппроксимации выходящих потоков для получения численных оценок межконецевых задержек в открытых тандемных сетях массового обслуживания с узлами типа MAP/PH/1/N.
3. Получение численных оценок межконецевых задержек для многошаговой беспроводной сети с помощью аналитического и имитационного моделирования.

Решению этих задач посвящена глава 4 диссертации.

### 1.3.3 Экспериментальная реализация системы

Для того, чтобы исследовать работоспособность системы радиочастотной идентификации автомобилей, были разработаны экспериментальные RFID-считыватели, распределенная система управления считывателями и программное обеспечение для центров обработки данных. Разработанный программно-аппаратный комплекс проходил испытания в трех экспериментах. Первый эксперимент прошел в 2014–2015 годах в городе Казань. Тогда метками были оснащены около 750 автобусов, в двух точках города установлены RFID-считыватели, информация с которых передавалась в центр обработки данных в ГИБДД. За несколько зимних месяцев была собрана статистика, вероятность идентификации составила 92–95 %. Следующий эксперимент проходил в 2020 году также в Казани, его целью была проверка системы при идентификации автомобилей, движущихся со скоростями до 170 км/ч и совершающих различные маневры. Третий эксперимент начался летом 2021 года на ЦКАД, его целью была проверка применимости системы для сбора данных для оплаты проезда. Все эксперименты завершились успешно.

При разработке программного обеспечения RFID-считывателей требовалось обеспечить его высокую надежность, быстродействие, а также гибкость. В частности, было принято решение разделить потоки данных и управления (сигнализации) так, чтобы, с одной стороны, сделать невозможным изменение параметров работы считывателя каким-либо образом, кроме как через административный интерфейс, а с другой стороны — упростить подключение внешних клиентов для получения потоков считанных меток. Программное обеспечение было спроектировано таким образом, чтобы можно было в быстро заменить радиомодули RFID-считывателей без изменения кода остальных компонентов, а также обеспечить возможность выноса части компонентов за пределы считывателя.

В состав разработанного программного обеспечения входит управляющий модуль (супервайзер), адаптеры радиомодулей, интерфейсы управления (веб-интерфейс, интерфейс командной строки), а также модули для приема и обработки потоков данных о прочитанных

метках. Все программное обеспечение, за исключением веб-интерфейса, было реализовано на языках C и C++.

Для экспериментального исследования производительности системы радиочастотной идентификации были решены следующие задачи:

1. Разработка архитектуры распределённой системы управления RFID-считывателями и протоколов связи между компонентами системы.
2. Программная реализация распределённой системы управления и разработка программного обеспечения для приема, обработки и записи данных о прочитанных метках.
3. Проведение экспериментальных исследований и получение оценки вероятности успешной идентификации автомобилей при различных скоростях движения, маневрах, погодных условиях.

Решению этих задач посвящена глава 5 диссертации.

#### 1.4 Радиочастотная идентификация транспортных средств

Технология радиочастотной идентификации (RFID) является альтернативой для традиционной идентификации номеров транспортных средств по фото. Для использования этой технологии автомобили оснащаются радиометками, а в точках контроля размещаются считыватели. Когда метка проезжает мимо считывателя, она передает свой идентификатор, по которому определяется номер автомобиля. К преимуществам этой технологии относится относительно низкая стоимость оборудования, низкие требования к пропускной способности каналов связи, отсутствие необходимости в частом обслуживании и слабая зависимость от погодных условий.

Существует несколько различных технологий радиочастотной идентификации. В **активных** системах используются метки, обладающие собственным источником питания. Часто такие системы работают в диапазоне 2.4 ГГц и могут быть основаны, например, на технологиях IEEE 802.11 (WiFi) или ZigBee. Можно выделить технологию DSRC (Dedicated Short-Range Communication), которая часто используется в системах оплаты проезда по автомагистралям, в том числе и в России. Дальность действия считывателя в системах с активными метками может достигать десятков или даже сотен метров, но метки достаточно громоздки и дорогостоящи.

К другому классу относятся **пассивные** системы, в которых метки не обладают собственными источниками энергии. Для работы и передачи данных метки в таких системах используют энергию, получаемую из электромагнитного поля, создаваемого считывателем. RFID-системы LF-диапазона (125 — 134 кГц, ISO/IEC 18000-2) характеризуются малым расстоянием идентификации (порядка нескольких сантиметров) и используются, например, для чипирования животных. Системы HF-диапазона (13,56 МГц, стандарты ISO/IEC 18000-3,

ISO/IEC 15693, ISO/IEC 14443 A,B) очень широко распространены в области электронной оплаты (например, оплата проезда на общественном транспорте) и контроля доступа, характеризуются низкой стоимостью и дальностью порядка нескольких миллиметров или сантиметров. Системы UHF-диапазона (860 — 960 МГц, ISO/IEC 18000-6С, EPC Gen-2) работают на расстояниях до 15–20 метров и используются в логистике, торговле, контроле доступа и на производствах. Кроме того, метки UHF-диапазона иногда оснащаются собственным источником питания, который может использоваться для работы встроенного сенсора. В этом случае метка может передавать результаты измерений, полученные от сенсора, а питание также может использоваться для усиления отражаемых сигналов. Такие метки называются полупассивными или полуактивными.

#### 1.4.1 Стандарт EPC Class 1 Generation 2

Стандарт EPC Class 1 Generation 2 [1] (ISO/IEC 18000-6С, ГОСТ Р 58701-2019 [2]) описывает физический (PHY) и канальный (MAC) уровни системы радиочастотной идентификации пассивных и полупассивных меток. На физическом уровне стандарт описывает способы модуляции и кодирования сигналов, а на канальном уровне — протокол обмена данными между считывателем и метками. Протокол основан на Slotted ALOHA [3;4], позволяет бороться с коллизиями при нахождении нескольких меток в зоне считывателя, а также содержит средства для чтения и записи данных на метки.

Поскольку пассивные метки не имеют собственного источника питания, считыватель постоянно создает электромагнитное поле, которое используется метками для получения энергии. Периодически считыватель отключает питание, обеспечивая меткам возможность сбросить некоторые регистры и флаги. В то время, когда считыватель не осуществляет передачу своих команд, он генерирует синусоидальный сигнал (Constant Wave, CW), который метки используют в качестве несущей — для передачи ответов метки модулируют отражаемый ими сигнал CW, полученный от считывателя, изменением своего коэффициента отражения (т.е. передача данных от меток ведётся методом обратного рассеяния).

Метки не могут сами инициировать обмен данными со считывателем, они всегда передают свои сообщения в ответ на получаемые от считывателя команды. Стандарт допускает следующие основные действия с меткой: инвентаризацию, чтение, запись и блокирование памяти, а также уничтожение метки. Инвентаризация — это опрос меток, в ходе которого они передают считывателю свои идентификаторы EPCID и выполняют другие операции, которые может запросить считыватель. Блокирование памяти означает, что считыватель впоследствии не сможет записать эту область памяти. Наконец, операция уничтожения метки приводит к тому, что в будущем метка никогда не будет участвовать в инвентаризации. Протокол определяет и другие операции, включая средства для защиты доступа к данным. Производители могут реализовать дополнительные операции на своих метках. В рамках диссертационной работы интерес будут представлять только инвентаризация и чтение памяти.

## Логическая структура памяти метки

Каждая метка включает в себя четыре банка памяти (см. рис. 1.2): EPC, TID, зарезервированную память (Reserved) и пользовательскую память (User).

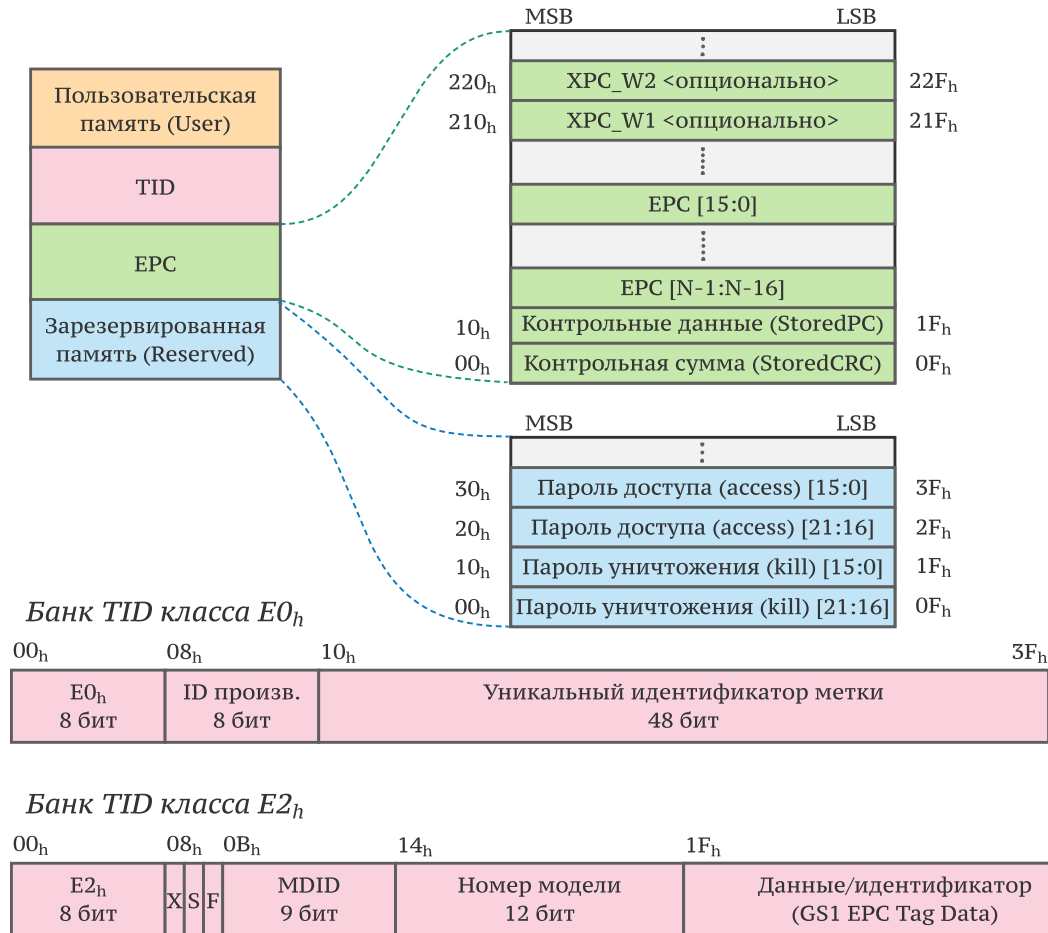


Рисунок 1.2 — Логическая структура памяти метки

Банк EPC содержит код EPC (Electronic Product Code), контрольную сумму (StoredCRC) и служебную информацию (PC, Protocol Control), в которой хранятся данные о длине поля EPC, доступности пользовательской памяти и прочую информацию. Объем памяти в поле EPC может меняться от метки к метке, распространенным значением является 96 бит. Чтобы не путать название банка памяти и значение EPC, значение в дальнейшем будем называть EPCID.

Банк Reserved хранит пароли для блокирования и уничтожения меток, если метка поддерживает соответствующие операции.

Банк TID хранит данные о производителе и модели метки, а также ее уникальный идентификатор. Главная особенность банка TID — фабричная защита от изменений значения, записанного при производстве метки, поэтому его удобно использовать в задаче надежной идентификации объектов. Первые 8 бит TID содержат идентификатор класса, который может быть равен либо E0h, либо E2h. Если идентификатор класса равен E0h, то длина банка равна 64 битам, второй октет содержит идентификатор производителя, а последние 48 бит

содержат серийный номер метки. Такие метки производства NXP использовались в эксперименте в Казани в 2014 году. Структура банка TID для класса E2<sub>h</sub> более сложная, а его размер — больше. Следующие после первого октета три бита — служебные флаги, после них — 9-битный идентификатор производителя MDID (Mask Designer Identifier), далее 12-битный номер модели и уже после него уникальный номер метки. Например, метки производства АО «Микрон», которые использовались в экспериментах 2020-го и 2021-го года, имеют класс E2<sub>h</sub> и содержат 96-битные значения.

Банк пользовательской памяти может иметь достаточно большой объем (порядка 512 бит) и используется для хранения дополнительных данных, например — подробной информации о маркированном предмете.

### Физический уровень

При передаче данных от считывателя к меткам используется амплитудную модуляцию DSB-ASK, SSB-ASK или PR-ASK. Для кодирования используется схема PIE (Pulse Interval Encoding), в которой нули и единицы кодируются символами data-0 и data-1 различной длины (см. рис. 1.3).

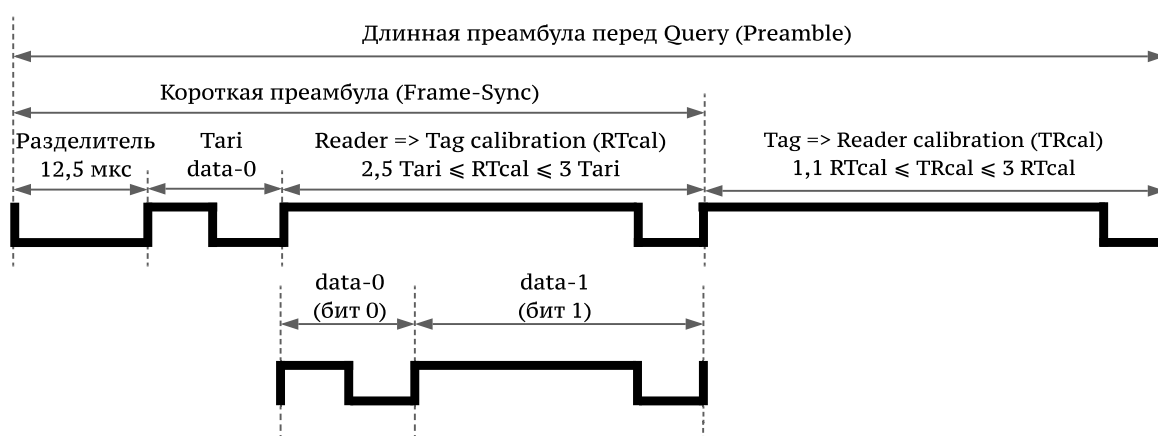


Рисунок 1.3 — Символы в схеме PIE и преамбулы команд RFID-считывателя

Передачу каждой команды считыватель начинает с синхронизирующей последовательности (преамбулы). Стандарт определяет два вида преамбул (см. рис. 1.3): длинную, называемую в стандарте преамбулой, и короткую, называемую синхронизацией кадра, (Frame-Sync). Оба вида преамбул начинаются с постоянного разделителя длительностью 12,5 мкс, за ним идут символы ноль (data-0) и RTcal (Reader-to-Tag calibration). Длительность символа data-0 называется Tari. Длительность символа RTcal равна суммарной длительности нуля и единицы, этот символ используется меткой для различения символов data-0 и data-1: если символ короче половины RTcal, то это ноль, иначе — единица. Короткая преамбула завершается передачей RTcal, а длинная содержит также символ TRcal (Tag-to-Reader

calibration), длительность которого используется меткой для расчета скорости, с которой она должна передавать свой ответ.

Длинная преамбула отличается от короткой наличием символа  $TR_{cal}$  и передается перед первой командой Query, с которой считыватель начинает опрос меток. Так как метка, не получившая Query, все равно не сможет участвовать в опросе, передавать  $TR_{cal}$  в последующих кадрах не имеет смысла и используется короткая преамбула, содержащая только ту информацию, которая нужна метке для успешного декодирования команды.

Отметим, что из-за использования P1E длительность кадров, передаваемых считывателем, зависит от числа нулей и единиц в команде. Поэтому для точного моделирования протокола необходимо учитывать результат кодирования команд.

Метка передает свой ответ, модулируя несущую, полученную от считывателя. Метка может использовать амплитудную (ASK) или фазовую (PSK) модуляцию. В качестве схемы кодирования используется либо код FM0, либо коды Миллера с 2, 4 или 8 символами на бит. Все схемы кодирования, поддерживаемые метками, имеют память, то есть вид символа, кодирующего следующий бит, определяется предыдущим битом. Перед передачей ответа метка передает преамбулу, которая может быть обычной или расширенной. Короткая преамбула равна по длительности 6 битам для кода FM0 и 10 битам для кодов Миллера, а расширенная - 18 и 22 битам соответственно. Кроме того, после каждого ответа метки следует символ, не кодирующий ни ноль, ни единицу.

Для вычисления скорости передачи символов метка вычисляет величину BLF (Backscatter Link Frequency) по следующей формуле:

$$BLF = \frac{DR}{TR_{cal}},$$

где DR (Divide Ratio) - величина, передаваемая в команде Query, и принимающая одно из двух значений: 8 или 64/3. Для вычисления битовой скорости необходимо разделить BLF на число символов на бит в используемом меткой методе кодирования ( $M=1,2,4,8$ ). В отличие от считывателей, использующих P1E, все биты в ответах меток имеют одинаковую длительность, поэтому их содержание не оказывает влияния на длительность передачи ответов.

## Логический уровень

Протокол доступа к каналу, описываемый в стандарте [1], основан на протоколе Framed Slotted ALOHA [3; 4]. Все время работы разбито на **раунды** (см. рис. 1.4), которые начинаются передачей считывателем команды Query. Эта команда, среди прочего, несет в себе параметр  $Q$ , который используется метками для расчета количества слотов в раунде как  $2^Q$ . Получив Query, метка выбирает случайный номер слота  $SN$  для передачи своего ответа от 0 до  $2^Q - 1$ . При последующем получении команд QueryRep метка уменьшает значение

счетчика  $SN$ . Когда счетчик доходит до 0, метка генерирует и передает случайное 16-битное число (RN16). Считыватель, получив это случайное число, пересылает его в команде Ack. Метка сравнивает полученное значение с отправленным. Если значения совпадают, то в ответ на Ack метка пересылает свой EPCID с контрольной суммой (PacketCRC) и контрольными данными (PC). Если же значения не совпадают, метка считает, что Ack предназначен не ей, и ничего не отвечает.

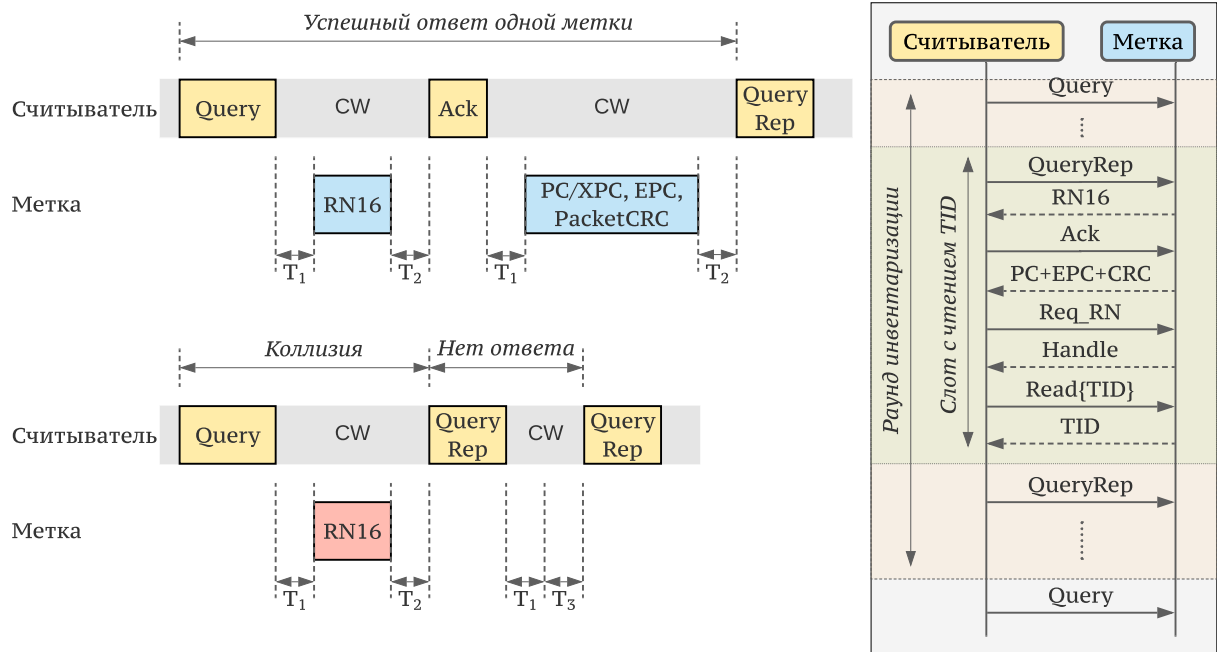


Рисунок 1.4 — Схема опроса меток и раунд инвентаризации

Стандарт определяет границы интервалов, которые должны пройти между последовательными командами и ответами:

- $T_1$ : минимальное время между концом последнего символа команды и первым символом преамбулы ответа;
- $T_2$ : минимальное время между окончанием последнего (псевдо-) бита ответа метки и первым символом преамбулы команды считывателя;
- $T_3$ : минимальный интервал между окончанием  $T_1$  и следующей командой считывателя.

Все, что происходит от передачи QueryRep до получения ответа от метки с EPCID, называется *стадией инвентаризации*. Именно в ней разрешаются коллизии — если несколько меток выбрали один слот и одновременно передали слова RN16, то считыватель, скорее всего, не сможет получить ни одно из них и либо вовсе не отправит Ack, либо отправит Ack с неправильным значением. Если же метка получила Ack с правильным значением, то коллизии не было, поэтому она может передать более длинное (по сравнению с RN16) значение EPCID. Если считыватель смог обнаружить ошибку в ответе метки, он может передать команду NACK, получив которую метка сможет определить, что ее идентификатор не был успешно передан. Также считыватель может повторно отправить Ack, чтобы метка повторила передачу EPCID.

Если после получения EPCID считывателю нужно выполнить над меткой другие действия, он начинает *стадию доступа*. В ней он запрашивает новое случайное число от метки с помощью команды Req\_RN и, после получения ответа, передает команду чтения, записи, блокирования или уничтожения метки. Если метка использует пароль для доступа, то перед передачей команды считыватель должен отправить пароль.

Помимо стадий инвентаризации и доступа, стандарт определяет *стадию выбора*. До отправки команды Query считыватель может передать команду Select, в которой задать параметры тех меток, ответы от которых он хочет получить в текущем раунде. Эту команду удобно использовать при опросе большого числа меток, например на складе. Еще одна интересная особенность стандарта — возможность динамически изменять число слотов. Если считыватель обнаруживает, что коллизии происходят слишком часто или, наоборот, слишком много пустых слотов, он может передать команду QueryAdjust с новым значением Q и запросить метки, которые еще не ответили в этом раунде, выбрать новые слоты. Подробнее про протокол взаимодействия между считывателями и метками можно прочитать в стандарте [1].

В диссертационной работе будет использоваться фиксированное значение Q и только одна операция доступа — чтение банка TID. Для ее выполнения считыватель использует команду Read, в которой отмечает нужный банк (TID), адрес начала памяти ( $00_n$ ) и число 16-битных слов, которые нужно прочитать. Если команда выполнена успешно, метка передает в ответ хранящееся значение. Пример чтения банка TID показан на рис. 1.4.

## Сессии

Сессии позволяют разграничить множество окружающих считыватель меток. Каждая метка должна поддерживать четыре сессии S0, S1, S2 и S3. Для каждой сессии метка имеет отдельный флаг, значения которого в стандарте обозначаются как A и B.

В команде Query считыватель указывает, с какой сессией работает в этом раунде, и какое значение флага сессии должно быть у меток, которые могут участвовать в опросе. После передачи EPCID метка, получив команду Query, QueryRep или QueryAdjust, должна инвертировать хранящееся у нее значение флага ( $A \rightarrow B$ ,  $B \rightarrow A$ ). Если в следующем раунде опрос будет проходить по той же сессии и тому же значению флага, эта метка участвовать в раунде не будет, так как у нее хранится инвертированное значение флага. Если считыватель не смог получить EPCID, он может передать команду NAK — в этом случае метка не изменит флаг и сможет участвовать в следующем раунде.

Сессии различаются по требованиям ко времени сохранения значения флага при потере питания, а также к начальным значениям флагов. Флаг сессии S0 всегда инициализируется в значение A. Он сбрасывается всякий раз, когда метка теряет питание. Флаг сессии S1 сбрасывается в значение A при включении метки, если со времени его изменения прошло не менее интервала, значение которого может варьироваться от 0,5 с до 5 с. Значение также сбрасывается в том случае, если метка не выключалась и с момента последнего изменения



прошло слишком много времени. Флаги сессий S2 и S3 должны сбрасываться в  $A$ , если с момента выключения метки прошло не менее 2 с. Таким образом, использование сессий S2 и S3 может предотвратить изменение флага в случае краткосрочной потери меткой энергии.

Как будет подробно исследовано в главе 3, выбор сессии и сценария опроса (всегда ли опрашивать метки по значению флага  $A$  или чередовать опросы по  $A$  и по  $B$ ) оказывает существенное влияние на вероятность успешной идентификации метки.

### 1.4.2 Обзор исследований производительности UHF RFID

Технология радиочастотной идентификации [5; 6] используется во множестве приложений, от учета товаров на складах, до поиска автомобилей на парковке с помощью дронов [7], позиционирования мобильных объектов [8–10], расчета скорости движущихся автомобилей [11–13], и даже выявления признаков усталости водителя [14] (в последнем случае метки предлагается размещать с двух сторон на шапке), или оперативного поиска потерявшихся в городе людей (например, с болезнью Альцгеймера или иными расстройствами) с помощью RFID-считывателей на припаркованных автомобилях [15].

Задача идентификации транспорта является одним из основных применений RFID [16]. Первые системы сбора платы за проезд, использующие RFID, появились в США в 1991 году [17]. Использование RFID для идентификации автомобильного и железнодорожного транспорта исследовалось в работах [12; 18–38]. В частности, в работе [26] Unterhuber и соавторы рассматривают систему идентификации быстрого транспорта (автомобилей и поездов) и утверждают, что наиболее критичный параметр, определяющий вероятность идентификации — число раундов, в которых успевает принять участие метка. Авторы предлагают простую модель для оценки максимального числа раундов, используют реалистичные модели антенных систем и получают результаты, доказывающие, что UHF RFID можно применять для идентификации очень быстро движущихся автомобилей и поездов. Unterhuber и соавторы также отмечают, что из-за многолучевого распространения и эффекта Доплера может существовать несколько отрезков, на которых возможна идентификация (аналогичные данные получены в моделях, представленных в диссертации). Вместе с тем, в работе [26] не рассматриваются коллизии или идентификация меток по TID. Стоит также выделить предыдущую работу Unterhuber и соавторов [25] и работу Magais и соавторов [39], в которых представлен анализ оптимального угла наклона антенн при идентификации транспорта.

Использование RFID в рамках интеллектуальных транспортных систем и «умных городов» — одно из перспективных направлений исследований [28; 37; 38]. Например, Pawłowicz и соавторы в работе [28] рассматривают задачу идентификации транспортного потока в контексте концепции «умного города», предлагают очень простую аналитическую модель и сравнивают данные, полученные с ее помощью, с результатами макетного моделирования.

Процесс чтения движущихся меток, в общем случае, не является надежным — метка может быть потеряна из-за высокой скорости движения, слабого сигнала или коллизий. Для

того, чтобы предсказать, будет ли идентифицирована метка, и извлечь дополнительную информацию из данных, полученных от считывателя, можно использовать методы машинного обучения. В работе [29] Jo и соавторы используют метод опорных векторов (SVM) для предсказания того, будет ли идентифицирован движущийся объект, в зависимости от скорости, угла наклона антенны и положения метки. А в работе [12] Choу и соавторы используют искусственные нейронные сети для вычисления скорости движения автомобилей по мощности принятого RFID-считывателем сигнала. В работе также отмечено, что основная задача — борьба с нарушениями скоростного режима, а метки желательно размещать в номерах автомобилей.

Применение пассивных и активных RFID-технологий на дорогах не ограничивается простой идентификацией транспорта. Так, в работах [13; 40–44] предлагается использовать RFID для чтения данных с дорожных знаков или с дорожного полотна. В этом случае метки располагаются на статичных объектах, а считыватели — в автомобилях. А в работе [45] Porter и Kim описывают результаты эксперимента, проведенного в штате Орегон (США), в ходе которого RFID-метки использовались для сбора и передачи данных о пройденном автомобилем пути для дальнейшего расчета налога на топливо при заправке — авторы совместно с управлением транспорта штата Орегон исследовали возможность гибкого расчета топливного сбора на основе пройденного автомобилем расстояния. Отметим, что в этой работе использовался RFID в диапазоне 2,4 ГГц с доступом к каналу CSMA/CA. В работах [37; 46; 47] предлагается использовать информацию, полученную при чтении меток, для анализа трафика и траекторий передвижения, позиционирования и поиска угнанных автомобилей. Отметим, что в работе [47] предполагается использование активных RFID в диапазоне 2,4 ГГц.

Некоторые исследователи предлагают устанавливать пассивные RFID-метки в дорожное покрытие и использовать их для измерения скорости движения или передачи данных автомобилям, оборудованным RFID-считывателем. Cheng и соавторы [43] предлагают использовать такие дорожные метки для позиционирования автомобилей. Jing и соавторы [41] также предлагают размещать метки на полосах движения и использовать их не только для передачи статических данных автомобилям (местоположение, направление движения, ограничение скорости и пр.), но и для записи автомобилями данных о текущей обстановке; эти данные затем могут быть получены другими проезжающими автомобилями, как от меток, так и в результате передачи данных по беспроводной сети между машинами. В работе [13] Jing и соавторы приводят простой расчет минимального расстояния между метками, при котором водители с агрессивной манерой вождения не смогут ускориться и замедлиться так, чтобы их мгновенная скорость существенно превышала ограничение, но средняя скорость оставалась допустимой.

Отметим, что протокол, лежащий в основе канального уровня стандарта EPC Gen-2 [1] — Framed Slotted ALOHA [3]. Исходный протокол ALOHA был представлен в работе [4]. В этой же работе, на основе предположения о пуассоновском потоке передаваемых пакетов, была получена оценка пиковой пропускной способности сети как  $1/(2e) \approx 0,186$ . Позднее, в работе [3] был описан способ повышения пропускной способности в два раза за счет разбиения времени на слоты (Slotted ALOHA).

Ряд работ посвящен имитационному моделированию систем UHF RFID [30; 41; 48; 49]. Так, в работе [48] была предложена модель JiST, реализованная на языке Java; в работе описывается дизайн системы моделирования и ее применение. К сожалению, симулятор JiST давно не обновлялся и, судя по всему, в настоящее время практически не используется. В работе [49] сделан фокус на особенности распространения сигналов, моделировании каналов передачи данных и прочих низкоуровневых особенностях RFID-системы, однако практически не учитываются аспекты верхнего (логического) уровня. Авторы работы [30] предлагают использовать комбинацию симулятора OMNeT++ для моделирования логики и MATLAB для моделирования физического уровня, в качестве примера рассматривают модель UHF RFID. В работе [41] также упоминается имитационная модель RFID, разработанная в системе моделирования OMNeT++.

Для того, чтобы адекватно моделировать процесс обмена сообщениями, учитывая потери из-за ненадежности каналов связи между считывателем и меткой, требуется детально учитывать особенности антенных систем, чувствительность меток и считывателя и свойства распространения сигналов. Эти вопросы изучались во многих работах, в том числе в [50–60]. Работы [50; 51] предлагают модели распространения сигналов, основанные на технике трассировки лучей. Авторы работы [50] рассматривают особенности распространения сигналов внутри помещений со множеством отражений, однако акцентируются на стационарных объектах и игнорируют эффект Доплера. В работе [51] авторы моделируют систему с движущимися транспортными средствами, однако не учитывают параметры логического уровня и фокусируются на оценке энергетических параметров в сравнении с данными, полученными из эксперимента. В работах [52; 55] предложена точная модель для расчёта бюджета соединений. В работе [55] Никитин и Рао рассматривают возможные конфигурации антенн считывателей, приводят типичные значения SNR для различных компонентов антенных систем считывателя, демонстрируют существенные различия в расчете затухания сигнала для однолучевой и многолучевой моделей, приводят формулы для расчета бюджета соединений, а также приводят результаты измерений для различных меток. Кроме того, авторы отмечают, что основной вклад в шум при приеме ответов считывателем вносит не термический шум, а утечка излучаемого сигнала. В диссертационной работе этот факт будет использоваться при расчете SNR.

Влияние коллизий и расчет BER рассматривается в работе Lazago и соавторов [61]. Авторы предлагают для расчета BER использовать не обычную модель канала с аддитивным Гауссовым шумом (AWGN), а модель, полученную из AWGN усреднением по соотношению SNR, которое предполагается случайным, имеющим распределение Рэлея.

Значительное число работ посвящено анализу производительности систем радиочастотной идентификации. Авторы пользуются эмпирическими методами [62], используют аппарат марковских случайных процессов как с идеальным каналом [63–69], так и с каналом с ошибками [70], а также используют иные аналитические подходы для анализа производительности [71–74]. В ряде работ также сравнивается производительность протокола Frame Slotted ALOHA с альтернативными протоколами и расширениями [75; 76]. В работе [62] M. Buettner пришел к выводу, среди прочего, что в ряде случаев при опросе выгоднее не пересылать

повторно команды АСК при потере ответа от метки, а лучше выключиться и попробовать в следующем раунде или использовать иную стратегию опроса. Аналогичный подход будет использоваться в диссертационном исследовании. Отметим также работу [69], в которой с помощью дискретной цепи Маркова исследуется эффект потери метками питания. Авторы описывают систему в виде двумерной цепи, моделируя изменение числа меток, которые продолжают участвовать в опросе, и число меток, уже передавших свои данные. В результате проведенного анализа, авторы находят оценки вероятности идентификации к концу раунда опроса и пропускную способность системы. В отличие от представленных в диссертационной работе моделей, потери пакетов возникают только вследствие коллизий, не учитываются сессии, и моделирование производится по слотами в одном раунде, а не по раундам. Также стоит выделить работу [28], в которой предлагается очень простая модель для оценки вероятности потери метки, то есть ее проезда области чтения без идентификации. Авторы делят область чтения на сегменты фиксированной длины и, полагая, что передавшие свой идентификатор метки не участвуют в последующих раундах, предлагают простой метод расчета. Как и в главе 3 диссертационного исследования, авторы [28] рассматривают изменения системы между раундами, в том числе — поступления и выходы метки из области чтения. В то же время, степень детализации модели в диссертационном исследовании существенно выше.

Модель системы радиочастотной идентификации транспорта, описанная в главе 2, в значительной степени опирается на результаты, описанные в работах Никитина и Рао [55], Lazaro [61], Griffin [52] и других. Новизна предложенной в диссертации модели заключается в совместном моделировании как особенностей радиопередачи (включая эффект Доплера и многолучевое распространение), так и антиколлизийного протокола в единой имитационной модели и получении из нее оценок вероятности идентификации автомобилей по паре меток. Кроме того, в большинстве работ не рассматривается идентификация по PID, которая исследуется в диссертационной работе.

Аналитическая модель, описанная в главе 3, также имеет ряд отличий от существующих исследований. Во-первых, переходы марковской цепи происходят на границах раундов, а не слотов, и учитывают сбросы питания и смены флагов опроса меток. Во-вторых, модель учитывает потери произвольных пакетов из-за ненулевой битовой ошибки. В-третьих, модель состоит из двух марковских процессов, первый из которых нужен для расчета числа меток, участвующих в раундах, а второй — для оценки вероятности идентификации. В-четвертых, матрицы переходных вероятностей строятся в виде произведения матриц отдельных операций (проведение опроса, инвертирование флага опроса, сброс питания, добавление или удаление метки из области чтения). Из-за этого марковские процессы оказываются неоднородными и тяжелее в анализе, но предложенный метод становится очень гибким. В дальнейшем метод можно расширить на более сложные операции.

## 1.5 Передача данных по беспроводным сетям

Исторически, одним из первых методов доступа к каналу стал CSMA/CA, который был взят за основу уже в первых версиях стандарта IEEE 802.11. Базовый механизм доступа, описанный в этом стандарте и основанный на CSMA/CA, называется DCF (Distributed Coordination Function). Хотя современные версии стандарта используют гораздо более совершенные методы конкурентного доступа, позволяющие учитывать QoS, производить групповые передачи, подтверждать группы пакетов вместо отдельных, объединять каналы и пр., так или иначе в их основе лежит DCF. Ввиду своей простоты, мы будем использовать его в качестве модельного образца конкурентного доступа к каналу.

### 1.5.1 Механизм доступа к каналу IEEE 802.11 DCF

Метод доступа CSMA/CA и механизм DCF хорошо известны и подробно описаны в различных статьях и книгах (например, см. работу Bianchi [77]), а наиболее подробное описание можно найти в стандарте IEEE 802.11. Вкратце опишем основы механизма.

Когда станции требуется передать новый пакет, в первую очередь она должна убедиться, что канал свободен. Если канал остается свободен в течение интервала DIFS (DCF Interframe Space), то станция выбирает случайное число слотов ожидания в интервале от 0 до  $CW-1$  ( $CW$  - Contention Window). Значение  $CW$  при первой попытке передачи равняется  $CW_{min}$ . Каждый слот имеет фиксированную длительность  $\sigma$ . Если в течение выбранного числа слотов канал остается свободным, то станция начинает передачу. В противном случае, если канал стал занят в течение очередного слота, станция останавливает отсчет, ждет освобождения канала, прослушивает его еще в течение DIFS, и после этого возвращается к отсчету слотов. Если передача прошла успешно, принимающая станция должна после интервала SIFS (Short Interframe Space, всегда короче DIFS) передать подтверждение ACK. Если подтверждение не было получено, передающая станция удваивает значение  $CW$  и повторяет попытку передачи. По достижении  $CW$  значения  $CW_{max}$  увеличение окна прекращается. Механизм такого ожидания передачи называется *backoff*. Возможные события при передаче показаны на рис. 1.5.

В действительности, механизм DCF имеет ряд дополнительных возможностей, позволяющих увеличить эффективность передачи. Во-первых, если передаваемые данные имеют достаточно большой размер, перед их передачей станция может отправить кадр RTS (Request to Send), в котором передается ожидаемое время занятости канала. В ответ на RTS получатель передает CTS (Clear to Send), которым подтверждает готовность вести прием. Этот механизм позволяет решить проблему скрытых станций. Во-вторых, после успешной передачи станция может начать отсчет *backoff* (так называемый *post-backoff*) и, если новый пакет придет от сетевого уровня до того, как отсчет закончится, ему потребуется ждать передачи

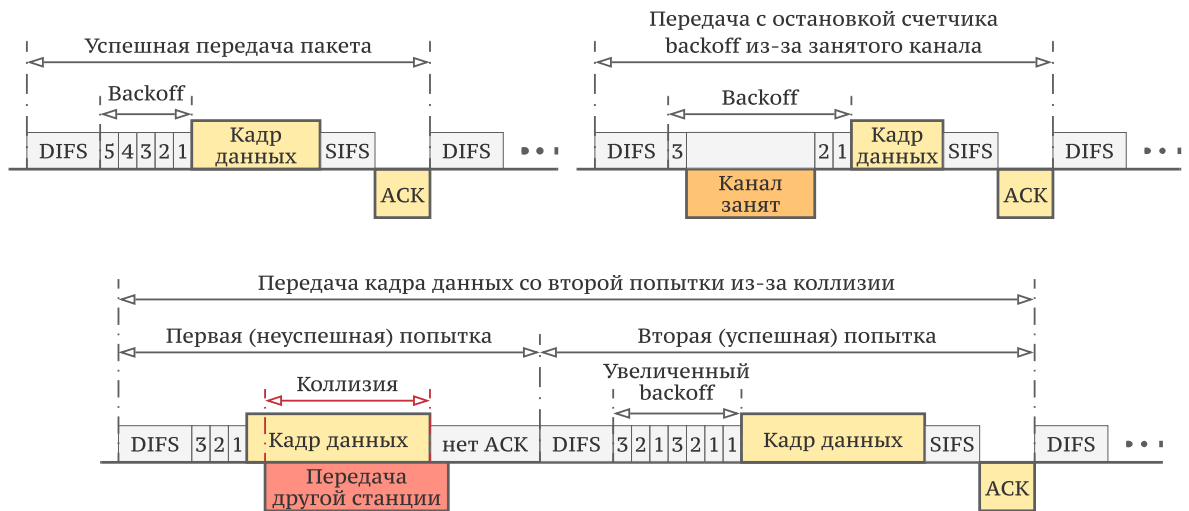


Рисунок 1.5 — Организация доступа к каналу при использовании простейшей версии механизма DCF

меньше времени. Количество попыток передачи пакета ограничено, и при его достижении пакет, вообще говоря, отбрасывается как недоставленный. Стоит также отметить, что при передаче больших пакетов может использоваться дефрагментация, и отдельные фрагменты могут передаваться без дополнительных отсчетов backoff. В диссертационной работе эти механизмы не учитываются, однако в дальнейшем их можно учесть для повышения точности моделирования. Что касается post-backoff и передачи группы кадров получившей доступ к каналу станцией (EDCF), для их адекватного учета нужно использовать более сложные методы описания времени передачи, например — марковские процессы обслуживания (MSP).

### 1.5.2 Обзор исследований производительности беспроводных каналов связи

Из всех характеристик беспроводных каналов, в диссертационном исследовании основной интерес представляет время, необходимое для передачи одного сообщения. Для оценки длительности передачи сообщений, а также для оценки пропускной способности канала часто используются марковские или полумарковские случайные процессы. В таких моделях состояние случайного процесса описывает состояние канала (например, «канал занят, ждать ещё два слота» или «ведётся передача»), и производится оценка либо доли времени, которое процесс находится в определенных состояниях, либо времени, нужного для попадания в некоторое состояние. Использованию марковских и полумарковских процессов для анализа производительности беспроводных сетей на основе CSMA/CA посвящено большое количество работ, одной из ключевых является статья Bianchi [77], в которой была предложена марковская цепь, моделирующая работу станции сети IEEE 802.11 под управлением механизма доступа DCF. Состояния цепи соответствуют слотам ожидания backoff и попыткам передачи. Важным допущением модели является постоянство условной вероятности коллизии при передаче, а также насыщенный режим работы. С помощью предложенной цепи

рассчитывается пропускная способность сети в насыщенном режиме. Полумарковские процессы также часто используются для описания работы беспроводной сети. Например, в работе [78] предложен полумарковский процесс, моделирующий работу канального уровня сети IEEE 802.15.4-2006. Вместе с полумарковским процессом рассматривается процесс восстановления, моделирующий физический уровень. Предложенная авторами аналитическая модель используется для оценки пропускной способности сети, а кроме этого — оценки интервалов между успешными передачами и распределения длительности ожидания в backoff.

Так как DCF является наиболее базовым механизмом доступа, используемым в сетях IEEE 802.11, анализ длительности передачи пакетов в каналах DCF подробно рассматривался в большом числе работ (например, [79–89]). Отметим, что многие из этих работ основаны на марковском процессе, описанном Bianchi [77]. Так, Chatzimisios и соавторы в работе [79] описывают простейшую модель для расчета среднего времени передачи пакетов, основанную на модели Bianchi, предполагая ограниченное количество попыток передачи. В работе [80] также представлена аналитическая модель для вычисления задержек в передаче пакетов в ad-hoc сети IEEE 802.11 под управлением DCF в нагруженном режиме; помимо самой задержки, авторы анализируют сценарии использования сети для передачи голосовых и обычных данных. Авторы [81] рассматривают работу ad-hoc сети в насыщенном режиме под управлением механизма DCF, получают аналитические выражения для математического ожидания и дисперсии времени обслуживания пакета, учитывая при этом ограниченное количество попыток передачи. Авторы также дают выражение для производящей функции моментов времени передачи пакета и исследуют функцию распределения этого времени. В работе [82] рассматривается задержка, учитывающая также нахождение пакета в очереди. В работе [83] Dong и соавторы изучают работу ad-hoc сети в ненасыщенном режиме, предполагая экспоненциальное распределение интервалов времени между поступлениями пакетов, и, помимо анализа времени обслуживания на отдельных узлах, рассматривают задержку при передаче по многошаговым маршрутам, учитывая наличие скрытых станций. Также проблеме измерения времени передачи при наличии скрытых станций посвящена работа [84]. В работе [85] представлен не только делательный анализ времени передачи пакетов в ad-hoc сети в ненагруженном режиме, но и рассматривается случай передачи группы пакетов при использовании механизма EDCA из IEEE 802.11e. Felemban и Ekici в работе [86] анализируют время передачи пакетов в одношаговых ad-hoc сетях, уточняя модель Bianchi за счет введения дополнительных марковских цепей в backoff-состояниях; ими также рассматривается случай ненасыщенной работы сети (при этом предполагается, что пакеты поступают в пуассоновском потоке). Также ненагруженный режим рассматривается в работе [90]. Еще один детальный анализ производительности ad-hoc сетей, включая длительности передачи пакетов, представлен в работе [88]. Следует также выделить работу [89], в которой время передачи пакета описано как PH-распределение.

Ряд исследований рассматривают и сравнивают различные схемы доступа, используемые в сетях IEEE 802.11. Можно выделить работу [91], в которой рассматриваются режимы доступа EDCA и HCF, определенные в стандарте IEEE 802.11e и обсуждается их применимость к передаче данных мультимедиа. Также можно выделить работу [92], авторы которой

приводят описание как основных методов доступа DCF, PCF, EDCA, HCCA, так и дополнения, вводимые расширениями IEEE 802.11ac и IEEE 802.11ad (следует отметить, что статья вышла в 2012 году, когда расширения IEEE 802.11ac и IEEE 802.11ad еще находились на стадии разработки). Анализ производительности методов доступа EDCA [93–99] и HCCA [100–102], введенных в стандарте IEEE 802.11e, широко рассмотрен в литературе. В работе [93] с помощью аппарата марковских цепей и систем M/G/1 анализируется пропускная способность канала связи при использовании метода EDCA, а также полная задержка пакетов, включая время ожидания в очереди. В работе [94] также предлагается использовать марковскую цепь для анализа производительности, а предложенная авторами модель учитывает блочные подтверждения и агрегацию кадров. Еще один пример марковской цепи, используемой для анализа EDCA, приведен в работе [95]. Авторы работы [96] анализируют задержки и пропускные способности каналов под управлением EDCA при передаче разных типов трафика. В работе [98] приводится большая аналитическая модель, позволяющая анализировать производительность EDCA в ненагруженном режиме. Следует также отметить работу [99], в которой авторы предлагают для анализа DCF и EDCA использовать модель PH/PH/1. В работе [100] авторы предлагают модель для анализа производительности канала связи при передаче голоса с использованием метода доступа HCCA и данных с использованием метода доступа EDCA. В работе [101] предлагается модель массового обслуживания MAP/PH/1 с приоритетами и периодами отдыха для анализа метода доступа HCCA. Авторы работы [102] также используют марковскую систему массового обслуживания с отдыхами для моделирования HCCA.

В более поздних работах исследуются нововведения в протоколы MAC-уровня, появившихся в дополнениях IEEE 802.11ac и IEEE 802.11ad. Так, в работе [103] анализируется производительность канала IEEE 802.11ac с использованием MIMO, учитывающая особенности как канального, так и физического уровня; нужно отметить, что в качестве модели канала используется, фактически, модель Bianchi [77]. В работе [104] приводится сравнение производительности IEEE 802.11n и IEEE 802.11ac, акцент сделан на исследовании механизмов агрегации кадров. Авторы работы [105] предлагают марковскую модель для анализа производительности миллиметровых каналов IEEE 802.11ad, учитывая такие особенности канала, как разделение пространства вокруг базовой станции на сектора.

## 1.6 Методы исследования задержек в многошаговых беспроводных сетях

Способы получения оценок задержек и пропускной способности сетей можно разбить на три группы: стендовые, имитационные и аналитические. Стендовые измерения проводятся на реальной сети или ее макете. Эти измерения наиболее точные, но очень ресурсозатратные, трудоемкие и не всегда осуществимые. Имитационные измерения проводятся с помощью различных систем имитационного моделирования, в которых сеть моделируется с помощью набора алгоритмов. Обычно эти алгоритмы воспроизводят работу моделируемых протоколов



с высокой точностью. Наиболее популярные открытые системы имитационного моделирования NS-3 и OMNeT++ содержат реализации большинства распространенных сетевых протоколов, в том числе IEEE 802.11 (WiFi) и IEEE 802.3 (Ethernet). Преимущество имитационного моделирования — высокая точность, сопоставимая с измерением на стенде, и возможность получить множество различных характеристик с одного запуска. Главный недостаток — высокая вычислительная сложность. Наконец, аналитические методы позволяют найти оценки исследуемых параметров с помощью математических моделей. Чаще всего эти математические модели абстрагируются от всего, что не связано непосредственно с исследуемыми характеристиками. В зависимости от конкретной модели, оценки возможно получить гораздо быстрее, чем с помощью имитационного моделирования. Однако, из-за высокой степени абстракции, ошибки могут быть достаточно высокими.

Для аналитической оценки времени, которое требуется пакету для прохождения по сети, чаще всего используется аппарат теории массового обслуживания. В рамках этого подхода интервалы между поступающими в сеть пакетами и времена передачи пакетов в каналах связи моделируются случайными распределениями, а память в интерфейсах коммутаторов — очередями, в которых пакеты ожидают начала обслуживания. От того, насколько удачно выбраны случайные распределения, зависит точность полученных результатов. Помимо оценки задержек, модели массового обслуживания позволяют оценить необходимую емкость очередей и оценить вероятность потери пакетов из-за их переполнений.

Выбор модели входящего потока, то есть распределений интервалов между поступлениями пакетов в сеть, зависит от передаваемого сетью трафика. В общем случае, сетевой трафик является коррелированным [106; 107]. Для представления такого трафика можно использовать модель марковских коррелированных потоков (Markovian Arrival Process, MAP).

Время передачи пакета в канале связи складывается из ожидания начала передачи, непосредственно времени передачи пакета, а также времени ожидания и получения подтверждения доставки (если протокол реализует подтверждения). Если канал ненадежен и в нем возможны потери, то приходится также учитывать неуспешные попытки передачи. Таким образом, выбор модели распределения времени обслуживания зависит от схемы доступа, которая используется в канале связи, окружения и размеров передаваемых пакетов. Для моделирования времени передачи можно использовать распределения фазового типа (Phase-Type, PH), или более общую коррелированную модель — марковские процессы обслуживания (Markovian Service Process, MSP). В диссертационном исследовании будем использовать PH-распределения, но многие выводы можно обобщить и на случай MSP.

### 1.6.1 Открытые сети массового обслуживания с марковскими потоками заявок и обслуживанием фазового типа

Для анализа характеристик многошаговых телекоммуникационных сетей используются модели открытых тандемных сетей массового обслуживания [108;109]. Такие модели состоят из обслуживающих устройств с очередями (узлов), последовательно соединенных друг с другом (см. рис. 1.6). Для простоты, будем считать, что весь внешний трафик поступает на первый узел, и передается последовательно через все промежуточные узлы. В более общем случае в сети может присутствовать кросс-трафик, то есть поступления пакетов снаружи на промежуточные узлы, а маршрутизация может быть сложнее, например — пакеты могут передаваться вперед через несколько узлов или даже назад, образуя циклы.

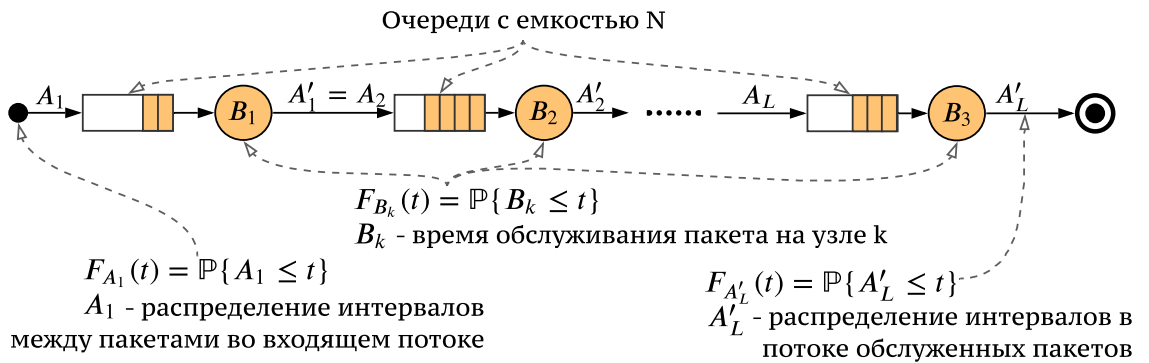


Рисунок 1.6 — Открытая сеть массового обслуживания с линейной топологией

Если в качестве распределения времени обслуживания использовать PH-распределения, а в качестве модели входящего трафика — MAP-потоки, то выходящий поток обслуженных пакетов из узла MAP/PH/1/N будет также MAP [110]. Благодаря этому свойству рассчитывать характеристики открытой сети с линейной топологией с узлами MAP/PH/1/N можно с помощью простой итерационной процедуры, которая будет рассматриваться в главе 4.

PH-распределения являются обобщением экспоненциальных распределений и могут использоваться для приближения любого непрерывного распределения на множестве неотрицательных чисел (см., например, [111]). Случайная величина  $B$  имеет PH-распределение, если ее можно выразить как время, проведенное в некоторой непрерывной цепи Маркова с генератором  $\tilde{S}$ , до попадания в поглощающее состояние. PH-распределение задается квадратной матрицей  $S \in \mathbb{R}^{V \times V}$  и вектором начального распределения  $\bar{\tau} \in \mathbb{R}^V$ , для которых должно выполняться:

$$\tilde{S} = \begin{bmatrix} S & -S\mathbf{1} \\ \mathbf{0} & 0 \end{bmatrix} \text{ — инфинитезимальная матрица,} \quad (1.1)$$

$$\forall i = \overline{1, V} : 0 \leq \tau_i \leq 1, \quad \sum_{i=1}^V \tau_i = 1.$$

Функция распределения случайной величины  $B \sim PH(S, \bar{\tau})$  имеет вид  $F_B(x) = \mathbb{P}\{B < x\} = 1 - \bar{\tau}e^{Sx}\mathbf{1}$  [112].

Марковские входные потоки (МАР) были впервые описаны в работах Neuts [113]. В ряде последующих исследований [106;107;114] было показано, что использование МАР-потоков позволяет достаточно точно моделировать сетевой трафик. МАР-поток  $A \sim MAP(D_0, D_1)$  является обобщением РН-распределений и позволяет учесть корреляцию между значениями с лагом  $k$ . Распределения интервалов между событиями в МАР-потоке описываются РН-распределениями с одинаковыми матрицами  $S \equiv D_0$ , но меняющимися начальными векторами  $\bar{\tau}_i$ , зависящими от того, в каком состоянии находился процесс при появлении предыдущего события. МАР-поток определяется с помощью двух матриц, сумма которых  $D = D_0 + D_1 \in \mathbb{R}^{W \times W}$  — инфинитезимальный генератор марковской цепи с непрерывным временем, управляющей переходами между состояниями процесса. Матрицы удовлетворяют ограничениям:

$$\begin{aligned} \forall i : \sum_{j=1} \{D\}_{ij} = 0, \quad \forall i \neq j : \{D\}_{ij} \geq 0, \quad \forall i : \{D\}_{ii} \leq 0, \\ \forall i, j : \{D_1\}_{ij} \geq 0, \quad \forall i \neq j : \{D_0\}_{ij} \geq 0, \quad \forall i : \{D_0\}_{ii} \leq 0. \end{aligned} \quad (1.2)$$

Более подробно свойства МАР-потоков и РН-распределений, которые потребуются для расчета характеристик тандемных сетей массового обслуживания, будут описаны в главе 4.

Помимо МАР-потока, иногда рассматриваются его обобщения на случай группового поступления пакетов (Batch МАР, ВМАР) [109;110;115] или поступлений пакетов с разными приоритетами (Marked Markovian Arrival Process, ММАР) [116–119]. В обоих случаях вместо двух матриц  $D_0$  и  $D_1$  рассматривается набор матриц  $D_i, i = \overline{0, K}$ , где матрицы  $D_i$  при  $i > 0$  соответствуют поступлению группы  $i$  пакетов (для ВМАР) или пакета  $i$ -го типа (для ММАР).

Одна из основных проблем, возникающих при получении численных характеристик сетей массового обслуживания с узлами МАР/РН/1/Н — экспоненциальный рост пространства состояний, из-за которого точный расчет возможен только для задач небольшой размерности. В связи с этим, для практического использования этой модели важно найти эффективные методы расчета характеристик тандемной сети массового обслуживания.

Исследованиям сетей массового обслуживания с МАР-потоками и РН-распределениями времени обслуживания посвящено множество работ. Например, в [120–123] предлагаются методы вычисления характеристик сетей массового обслуживания на основе методов декомпозиции и представления искомых характеристик в форме произведений. В ряде работ акцент сделан на тандемных сетях массового обслуживания и изучении их характеристик [124–127], а также исследованию выходных потоков из систем массового обслуживания с марковскими входными потоками [128; 129].

### 1.6.2 Методы восстановления РН-распределений и МАР-потоков

Методы поиска РН-распределений и МАР-потоков в диссертационном исследовании будут применяться для решения двух задач: для поиска распределений, моделирующих время передачи пакетов в каналах связи, и для поиска МАР-потоков меньшей размерности, аппроксимирующих потоки обслуженных пакетов на выходе из очередного узла тандемной сети.

Для поиска РН-распределения или МАР-потока можно использовать два подхода: принцип максимального правдоподобия и метод моментов. В первом случае необходима выборка значений распределения, на которой максимизируется функция правдоподобия. Существует ряд реализаций, основанных на методе максимизации ожидания (ЕМ-процедуре) для МАР-потоков [130–133] и РН-распределений [134–139]. Главное ограничение метода — высокая вычислительная сложность, что ограничивает его использование в итерационных алгоритмах, где его необходимо применять многократно. Из-за этого в диссертационном исследовании этот подход не будет использоваться, хотя он оказывается достаточно удобным при поиске МАР-потоков и РН-распределений при моделировании реальных входящих данных, полученных, например, из сетевого дампа.

Второй подход основан на методе моментов. Он применяется как для РН-распределений [111; 130; 140–145], так и для МАР-потоков [145–150]. В первом случае РН-распределение строится по известным значениям первых моментов  $m_1, m_2, \dots, m_K$  случайной величины, во втором также могут учитываться значения коэффициентов корреляции  $l$ -го порядка (с лагом  $l$ )  $\rho_1, \rho_2, \dots, \rho_L$ .

Сложность применения метода моментов заключается в том, что точные области существования РН-распределений и МАР-потоков заданной размерности  $W$ , то есть множество значений моментов  $m_1, m_2, \dots, m_K$  и коэффициентов корреляции  $\rho_1, \rho_2, \dots, \rho_L$  при сколь угодно больших  $K$  и  $L$ , для которых существует РН-распределение или МАР-поток размерности  $W$ , имеют сложный вид [140; 141; 146–148]. Другими словами, при известных значениях моментов и коэффициентов корреляции сложно не только найти РН-распределение или МАР-поток минимально возможного порядка, но и даже определить этот порядок. При этом, хорошо известны грубые оценки, связывающие размерность и значения моментов. Например, в работе [151] доказано, что среди всех РН-распределений заданного размера наименьшее значение коэффициента вариации достигается на распределении Эрланга, что позволяет грубо оценить снизу размерность искомого РН-распределения при известных первом и втором моменте. Есть методы, позволяющие находить распределения малого порядка, но не для любых наборов значений моментов. Например, в работе [142] исследуется область существования ациклических РН-распределений второго порядка АРН(2), а в работе [147] Vodrog и соавторы исследуют область существования МАР-потоков второго порядка. Telek и Horvath [146] описывают минимальные формы РН-распределений и МАР-потоков (марковская, жорданова и прочие формы), а также преобразования между ними. Авторы предлагают алгоритм оптимизации матриц  $D_0$  и  $D_1$ , позволяющий минимизировать целевую функцию, зависящую

от  $B^{-1}D_0B$  и  $B^{-1}D_1B$  посредством поиска матрицы преобразования  $B$ . Предложенный метод позволяет улучшить МАР-поток, который был получен любым другим методом.

Если ограничения на порядок существенны, то в общем случае приходится искать такой МАР-поток  $X \sim \text{MAP}(D_0, D_1)$  (РН-распределение  $Y \sim \text{PH}(S, \bar{\tau})$ ), чтобы выполнялись ограничения (1.2) (для РН-распределения — (1.1)), и достигался минимум функции ошибки при заданных весах  $\bar{w}$ :

$$\mathcal{L}_{\text{МАР}}(X; \bar{m}, \bar{\rho}, \bar{w}) = \sum_{k=1}^K w_k (m'_k - m_k)^2 + \sum_{l=1}^L w_{K+l} (\rho'_l - \rho_l)^2 \rightarrow \min,$$

$$\mathcal{L}_{\text{РН}}(Y; \bar{m}, \bar{w}) = \sum_{k=1}^K w_k (m'_k - m_k)^2 \rightarrow \min.$$

Здесь  $m'_k$  —  $k$ -й момент МАР-потока  $X$  или РН-распределения  $Y$ , а  $\rho'_l$  — коэффициент корреляции  $X$  с лагом  $l$ .

При поиске РН-распределений чаще всего накладываются ограничения на структуру матрицы  $S$ , то есть поиск ведется по некоторому подсемейству распределений. Есть реализации метода моментов [111; 130; 141; 144], которые позволяют находить распределения по произвольному набору моментов. В работе Johnson и Taafe [111] предлагают искать РН-распределения по заданным трем моментам в виде гиперэрланговского распределения с двумя цепочками Эрланга одинакового порядка. Bobbio и соавторы [141] предлагают метод поиска ациклических РН-распределений АРН( $n$ ) минимального порядка  $n$  также по первым трем моментам; кроме того, авторы описывают границы значений первых трех моментов для распределений АРН( $n$ ) при известном  $n$ . В работе [130] Horvath и соавторы предлагают метод поиска по произвольному числу первых моментов специального подкласса РН-распределений — обобщенных гиперэрланговских распределений. Метод позволяет находить множество допустимых РН-распределений, из которых можно выбрать наиболее подходящее по той или иной метрике. Casale и соавторы [150] предлагают метод поиска МАР-потоков по первым трем моментам и коэффициентам корреляции как композиции МАР-потоков меньших размеров (метод композиции через кронекерово произведение, Kronecker Product Composition, КРС).

Если задача состоит в поиске МАР-потока  $X'$  с теми же характеристиками, что и МАР-поток  $X$ , но имеющего меньший порядок, то можно использовать и другие подходы. Например, простой метод сокращений пространства состояний основан на идее о том, что некоторые состояния МАР-потока большой размерности избыточны, и их можно удалить. Этот метод применим для систем МАР/РН/1 и МАР/РН/1/ $N$ , в которых коэффициент загрузки достаточно мал. Например, авторы работы [152] рассматривают выходной поток (обслуженных пакетов) системы МАР/МАР/1 и предлагают удалять блоки матриц  $D_0, D_1$ , соответствующие числу пакетов в системе свыше заданного уровня  $N$ .

В диссертационной работе для поиска РН-распределений и МАР-потоков будет использоваться метод моментов, а именно — простой метод поиска РН-распределений по первым трем моментам в виде распределений гиперэрланга с двумя цепями одинаковых порядков,

предложенный в работе Johnson и Taafe [111]. При поиске MAP-потока этот метод будет использоваться для расчета матрицы  $D_0$ , а поиск матрицы  $D_1$  будет производиться отдельно по заданным коэффициентам корреляции, как предложено в работе Horvath и соавторов [149]. Как показали численные эксперименты, эта комбинация методов просто реализуется, работает достаточно хорошо и стабильно и позволяет находить распределения за короткое время, хотя порядки могут быть несколько больше, чем в альтернативных методах. Время работы особенно критично, так как в численном эксперименте требовалось рассчитать характеристики нескольких сотен сетей с узлами MAP/PN/1/N.

## 1.7 Методы построения распределённых систем радиочастотной идентификации

С момента появления технологии RFID ведутся исследования в области создания систем управления считывателями и промежуточного программного обеспечения (Middleware). Хороший обзор ранних подходов и практически реализованных систем приведен в работе Al-Jaroodi и соавторов [153], где они рассматривают общие требования к промежуточному программному обеспечению для RFID и анализируют семь различных систем с точки зрения надёжности, масштабируемости, балансировке нагрузки, возможности управления данными и безопасности. В более позднем обзоре Razzaque и соавторы [154], рассматривая RFID уже в качестве неотъемлемой части Интернета вещей (Internet of Things, IoT), расширяют список требований, предъявляемых к промежуточному программному обеспечению, рассматривая требования к программным интерфейсам, необходимость реализации программного обеспечения в рамках сервисно-ориентированной модели Software-as-a-Service (SaaS) и Everything-as-a-Service (XaaS).

Для подключения RFID-считывателей к системам управления используются стандартные протоколы, которые поддерживаются большинством производителей. Протокол Low Level Reader Protocol (LLRP) [155] определяет низкоуровневый интерфейс между считывателем и контроллером. Этот протокол позволяет выполнять произвольные операции доступа и инвентаризации, настраивать параметры радиоинтерфейса считывателя, получать данные о состоянии радиоканала и диагностические данные о работе считывателя. Протокол LLRP поддерживается большинством существующих считывателей. Стандарт определяет возможность работы протокола по каналам TLS. Помимо LLRP, организация EPCglobal опубликовала несколько стандартов [156–159], которые можно использовать при разработке считывателя и системы управления. Стандарт «Reader Management 1.0.1 (RM)» [156] определяет модель системы и MIB (Management Information Base) для сбора данных о состоянии работы считывателя через протокол SNMP. Стандарт «Discovery, Configuration, and Initialization (DCI) for Reader Operations» [157] позволяет считывателю, контроллерам и LLRP-клиентам находить друг друга в сети, производить аутентификацию между кон-

троллерами и считывателями, управлять работой считывателей, загружать новые образы программного обеспечения и выполнять прочие служебные функции. Стандарт «Application Level Events (ALE) Standard» [158;159] описывает рекомендации по разработке промежуточного программного обеспечения. Кроме стандартов EPCglobal, существует спецификация интерфейса между считывателем и контроллером от альянса RAIN RFID [160]. Эта спецификация более высокоуровневая, по сравнению с LLRP. Она описывает основные действия, включая конфигурацию и получение статуса считывателя, настройки радиопотокола, доступ к меткам. Спецификация определяет сообщения, передаваемые между считывателем и контроллером, в формате JSON.

В целом ряде ранних исследовательских работ [161–167] рассматриваются детали архитектуры и подходов к построению промежуточного программного обеспечения, причем материал некоторых из них и на сегодняшний день представляет интерес. Например, Ноаг и Thompson в работе [161] выделяют два уровня архитектуры: уровень агентов и уровень RFID-приложений. Предложенная архитектура допускает подключение различного оборудования (считыватели, принтеры меток, базы данных и пр.) через специальных модулей-агентов, использующих для обмена сообщениями унифицированный транспортный уровень, на котором реализуются как односторонние обмены сообщениями, так и многоадресные соединения для связи типа «публикация-подписка». Архитектура системы на основе механизма подписок также рассматривается в работе Yu и Lai [162]. В работе [163] описывается двухуровневая архитектура промежуточного программного обеспечения RFID с фильтрацией и агрегацией данных, а в работе [164] предлагается построение промежуточного программного обеспечения с использованием адаптеров, предоставляющих унифицированный интерфейс для подключения считывателей. Авторы [166] предлагают модель системы управления считывателями и промежуточным программным обеспечением, направленную на повышение отказоустойчивости. Наконец, ещё один ранний проект открытого промежуточного программного обеспечения для RFID (Accada) был предложен в работе Floerkemeier и соавторов [167]. В нем авторы поднимают ряд проблем, от различия между требованиями пользователей вплоть до многолучевого распространения сигналов, и предлагают оригинальные решения, например — использование виртуальной памяти для отложенной записи данных на метки или механизм прогнозирования длительности нахождения меток в области чтения.

В более поздних исследованиях разработчики систем управления и промежуточного программного обеспечения рассматривают RFID в более широком контексте интернета вещей (IoT). В работе [168] предложена пятиуровневая модель, в которой данные со считывателя поступают в облако и обрабатываются при помощи различных микросервисов. Aazam и Nuh в работе [169] рассматривают системы управления и взаимодействия с устройствами (в частности, с RFID-считывателями и метками) в контексте интернета вещей (IoT), интернета всего (Internet of Everything, IoE) и туманных вычислений (Fog Computing). Например, в работе [170] Schreiber с соавторами предлагают архитектуру промежуточного программного обеспечения, позволяющего единообразно работать с RFID-считывателями и сенсорами. Помимо детального описания архитектуры, авторы также предлагают SQL-подобный язык,

позволяющий пользователю строить запросы к системе, не вдаваясь в специфику того или иного устройства. В более ранней работе [171] Cho и соавторы рассматривают распределенную систему, позволяющую не просто идентифицировать объекты, но связывать данные о них с показателями сенсоров.

Хотя в большинстве работ промежуточное программное обеспечение включает в себя функции как для получения и обработки данных о прочитанных метках, так и функции управления и мониторинга состояния считывателей, некоторые авторы допускают их разнесение. Например, Kamoun в работе [172] предполагает, что системы управления считывателями лучше не рассматривать в контексте промежуточного программного обеспечения, оставив последнему функции по работе с данными. В работе [173] производится анализ и выбор метрик производительности промежуточного программного обеспечения для RFID. Авторы рассматривают в качестве метрик зависимости загрузки системы (использование процессоров и памяти) и времени отклика на запросы клиентов от числа подключенных считывателей, числа подключенных приложений (клиентов), а также количества читаемых каждым считывателем меток. Также авторы описывают дизайн системы анализа производительности промежуточного программного обеспечения для RFID и приводят краткое описание собственной реализации.

Многие исследования направлены на разработку программного обеспечения для специализированных применений RFID. Например, в работе [174] описывается архитектура информационной системы, в которой один терминал (клиент) подключается ко множеству считывателей (серверов), для применения на автомобильном заводе. Статья [175] посвящена описанию системы отслеживания местонахождения людей с помощью имплантированных RFID-чипов под кожу. В работе [176] описана информационная система для управления библиотечным фондом с использованием RFID, авторы [177] описывают архитектуру и опыт реализации RFID-системы на фабрике по выпуску одежды, а в работе [178] предлагается использовать RFID для систем обеспечения безопасности перевозки школьников в автобусах. В статье [179] рассматривается использование RFID в области фармацевтики для контроля за качеством лекарств и отслеживанием контрафакта. Работы [180; 181] посвящены системам отслеживания багажа в аэропортах с помощью RFID.

Помимо технической стороны разработки информационных систем с использованием RFID, есть ряд вопросов этики и безопасности, поскольку данные на метках находятся в тесной связи с людьми и бизнесом. Хорошую подборку и анализ как преимуществ внедрения RFID, так и проблем, связанных с этим (в частности, кражей данных и отношением людей к сбору данных о себе), можно найти в ранней работе [182]. В работе [175] предлагается использовать блокчейн для распределенного контроля доступа к меткам. Бельский и соавторы [183] исследуют безопасность систем RFID и приводят описание моделей угроз. Авторы работы [184] описывают требования к распределенной системе управления мобильными RFID-считывателями, уделяя особое внимание вопросам безопасности. Также стоит выделить работу Fouladgar и Afifi [185], в которой они предлагают безопасный протокол делегирования из доверенной центральной информационной системы прав на идентификацию меток считывателям.



## 1.8 Заключение

В главе была описана общая структура системы радиочастотной идентификации транспорта, разработке и исследованию которой посвящена диссертационная работа. Выделены ключевые компоненты системы: RFID считыватели, метки, телекоммуникационная сеть и центр обработки данных; для каждого из компонентов поставлены задачи, решаемые в диссертационном исследовании.

После постановки задач исследования, в главе были описаны технологии, используемые в распределенной системе радиочастотной идентификации, и математические методы, используемые для их исследования. По всем упомянутым технологиям и методам был приведен обзор литературы и актуальных научных исследований. Также в общих чертах были выделены отличия и новизна предложенных в диссертации методов и подходов от существующих в литературе.

## Глава 2. Исследование производительности систем радиочастотной идентификации

В настоящей главе исследуется возможность использования технологии пассивной радиочастотной идентификации стандарта EPC Gen-2 [1] для идентификации автотранспорта, движущегося со скоростями порядка 60—150 км/ч. Сначала, в разделе 2.1, будут описаны компоненты системы радиочастотной идентификации автомобилей, а в разделе 2.2 также в наиболее общей форме приведены зависимости, определяющие вероятность успешной идентификации подвижной метки. Эта вероятность зависит от числа раундов, в которых метка успевает принять участие, а также от вероятности идентификации в каждом из раундов. Для анализа факторов, от которых зависит число раундов, в разделе 2.3 будут исследованы зависимости длительностей команд, ответов и раундов от различных параметров протокола, а для анализа вероятности идентификации метки в каждом из раундов в разделе 2.4 будет приведена модель радиоканала, учитывающая особенности многолучевого распространения сигналов вблизи дороги, зависимость мощности отраженных сигналов от материала поверхности дороги, а также влияние эффекта Доплера, и будут получены оценки вероятностей битовых ошибок (BER) при передаче ответов меток. С учетом сделанных расчетов будет уточнено определение области, в которой метки имеют возможность передать свои идентификаторы считывателю, и будет сделан аналитический расчет вероятности идентификации метки. В разделе 2.5 кратко описывается имитационная модель и приводятся результаты моделирования системы радиочастотной идентификации автомобилей. Раздел 2.6 завершает главу. В следующей главе будет приведена аналитическая модель, позволяющая более точно оценить влияние сбросов питания и смены флагов опроса на вероятность идентификации.

Результаты, представленные в главе, были опубликованы в журнале IEEE Journal on RFID [186], а также в сборниках, индексируемых Scopus/WoS [187–189].

### 2.1 Структура системы радиочастотной идентификации

Система радиочастотной идентификации автомобилей состоит из RFID-считывателя и RFID-меток (радиометок), размещенных на автомобилях (см. рис. 2.1). Метки и считыватель работают по стандарту EPC Class 1 Generation 2 [1]

Метки могут размещаться на регистрационных номерных знаках, под лобовым стеклом или на корпусе автомобиля. В настоящей работе рассматривается случай, когда метки расположены на номерах, поскольку этот вариант был использован в проведенном в г. Казань эксперименте. Кроме того, размещение меток в номерных знаках обладает рядом преимуществ: расположение знаков относительно дорожного полотна варьируется в малых пределах, зона номерного знака не загорожена никакими проводящими поверхностями, на-



Рисунок 2.1 — Структура системы радиочастотной идентификации автомобилей

конец это решение можно в дальнейшем легко стандартизировать и масштабировать за счет монтирования меток в выпускаемые номерные знаки.

RFID-считыватель размещается над дорогой (например, на тех же опорах, на которых сегодня размещаются камеры), на высоте приблизительно 5 метров; считыватель может оснащаться от 1 до 4 антенн. Таким образом, один считыватель может читать метки в передних и задних номерах на двухполосной дороге.

Связь между меткой и считывателем возможна при соблюдении двух условий: метка получает достаточно энергии для работы и декодирования команд от считывателя (чувствительность современных меток имеет порядок  $-18$  дБм), и отраженный модулированный меткой сигнал поступает на считыватель на достаточно высокой мощности, чтобы быть успешно распознанным и декодированным (при отсутствии коллизий, уровень принимаемого сигнала для современных считывателей должен быть не ниже минус 80 дБм). На практике это означает, что дальность связи составляет порядка 8 – 15 метров. Однако, из-за особенностей распространения сигнала вблизи автодороги, область, в которой метка может получить достаточно энергии, может иметь достаточно сложный вид и состоять из нескольких непересекающихся областей, общая ширина которых может составлять 2 – 3 метра. Подробнее эти эффекты будут рассмотрены в последующих разделах.

## 2.2 Общая схема расчёта вероятности идентификации автомобилей

Производительность системы определяется как доля успешно идентифицированных автомобилей. Полагая, что успешного чтения хотя бы одной метки достаточно для идентификации автомобиля, вероятность идентификации можно описать следующим образом:

$$\mathbb{P}\{A_x\} = 1 - \prod_{\forall t \in \mathfrak{S}_x} (1 - \mathbb{P}\{A_t\}),$$

где  $A_x$  — событие идентификации автомобиля  $x$ ,  $A_t$  — событие идентификации метки  $t$ , а  $\mathfrak{S}_x$  есть множество меток, размещенных на автомобиле  $x$ . Вероятность идентификации метки  $\mathbb{P}\{A_t\}$  зависит от длительности раунда инвентаризации, вероятности успешной передачи ответа и числа меток в области чтения:

$$\mathbb{P}\{A_t\} = 1 - (1 - (1 - \mathbb{P}\{A_c\})\mathbb{P}\{A_r\})^{N_r},$$

где  $A_c$  — событие возникновения коллизии,  $A_r$  — событие успешной передачи меткой ее идентификатора и  $N_r$  — число раундов, в которых метка успевает принять участие.

Пусть  $N_t$  — число меток в области чтения и  $Q$  — параметр антиколлизийного протокола. [1]. Тогда вероятность возникновения коллизии для данной метки определяется следующей формулой:

$$\mathbb{P}\{A_c\} = 1 - (1 - 2^{-Q})^{|N_t|-1}.$$

Число раундов можно грубо оценить как:

$$N_r \approx \frac{L}{v} \cdot \frac{1}{\tau},$$

где  $L$  — общая длина участка дороги с хорошими условиями для чтения (т.е. на этом участке метка получает достаточно энергии для работы и битовая ошибка достаточно низка),  $\tau$  — средняя длина раунда при заданном числе меток и параметрах протокола,  $v$  — скорость движения метки. В действительности количество раундов представляет собой случайную величину, зависящую от множества параметров, включая количество меток в области чтения, настройки протокола, непосредственный выбор метками слотов для ответов (определяет количество коллизий, пустых слотов и слотов с ответами). Также число раундов зависит от результатов попыток передачи меткой её идентификатора, а также от того, сбрасывал ли считыватель питание между раундами — влияние этих факторов оказывается очень существенным и исследуется подробнее в следующей главе диссертации.

Согласно [55], при отсутствии коллизий основным источником ошибок при расчете  $\mathbb{P}\{A_r\}$  является высокий BER при приеме ответов от меток. Пусть  $r$  — ответ метки,  $\mathfrak{R}$  — множество всех ответов метки, передача которых необходима для успешной идентификации. Обозначая битовую длину ответа как  $|r|$ , вероятность успешной передачи всех ответов  $\mathbb{P}\{A_r\}$  метки можно вычислить по формуле:

$$\mathbb{P}\{A_r\} = \prod_{r \in \mathfrak{R}} (1 - B)^{|r|}.$$

Множество  $\mathfrak{R}$  обязательно содержит ответ RN16 на команду, с которой начался слот, в котором метка передаёт ответ (Query, QueryRep или QueryAdjust) и ответ на команду ACK (PC+EPC+CRC). Если для идентификации метки также используется 64-битное значение из банка TID, то  $\mathfrak{R}$  также включает ответы на Req\_RN и Read.

Из приведенных формул видно, что для максимизации вероятности идентификации автомобиля необходимо увеличивать вероятность успешной передачи меткой ответов и число раундов, и одновременно уменьшать вероятность коллизий. Однако эти требования вступают в противоречие: более низкий BER достигается при более надежных и медленных настройках протокола, а вероятность коллизии снижается при росте числа слотов, но все это ведет к увеличению длительности раунда. В последующих разделах будет разработан комплекс моделей, позволяющий проанализировать вероятность идентификации метки при различных

допущениях, а в конце главы будут приведены результаты расчета вероятности идентификации автомобиля при различных настройках протокола и скорости движения, полученные с помощью детализированной имитационной модели.

### 2.3 Анализ влияния параметров протокола на длительности раундов

Считыватель использует модуляции DSB-ASK, SSB-ASK и PR-ASK, а в качестве схемы кодирования используется PIE. Из-за использования PIE (Pulse Interval Encoding) длительность передачи команды зависит от её содержания, так как длительности символов `data_0` и `data_1` отличаются в 1,5–2 раза. Перед каждой командой Query считыватель передает преамбулу, а перед остальными — более короткую синхронизирующую последовательность.

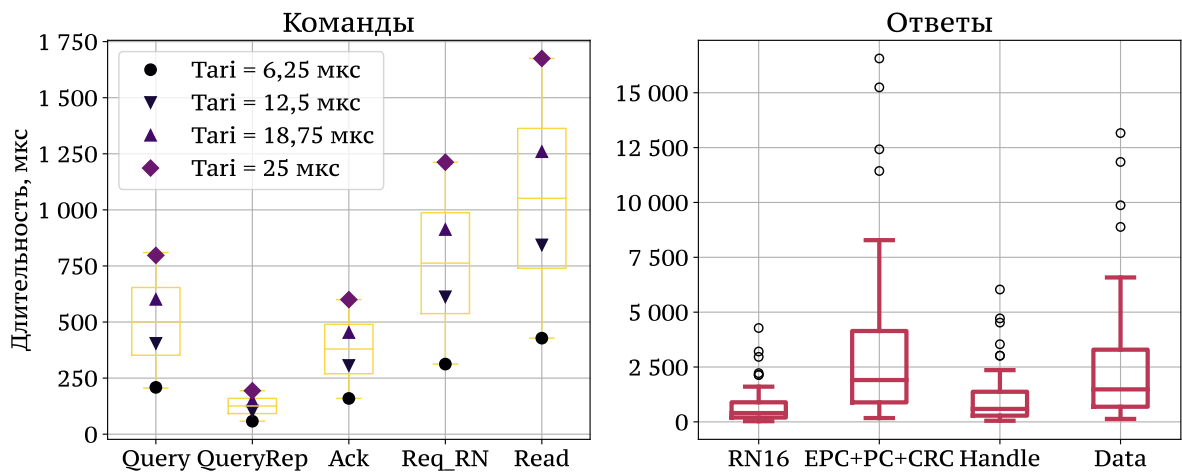


Рисунок 2.2 — Разброс длительностей сообщений между считывателем и меткой.

На рис. 2.2 показан разброс длительностей команд считывателя и ответов меток для разных настроек протокола. Длительности команд могут различаться в четыре раза, так как все символы, передаваемые считывателем, пропорциональны длительности  $T_{ari}$ . В действительности разброс длительностей команд еще больше — при построении графика предполагалось, что отношение длительностей  $RT_{cal}$ ,  $TR_{cal}$  и  $T_{ari}$  фиксировано ( $RT_{cal} = 2,75 T_{ari}$ ,  $TR_{cal} = 2,05 RT_{cal}$ ), а стандарт определяет их в диапазоне  $2,5 T_{ari} \leq RT_{cal} \leq 3 T_{ari}$  и  $1,1 RT_{cal} \leq TR_{cal} \leq 3 RT_{cal}$ .

Разброс длительностей ответов меток даже больше, чем разброс длительностей команд. Это обусловлено тем, что длительности ответов зависят не только от  $T_{ari}$ , но и от параметра DR, способа кодирования ответа (1, 2, 4 или 8 символов на бит) и того, используется ли расширенная преамбула. На рис. 2.3 показаны зависимости длительностей ответов меток с передачей EPCID, PC и CRC от различных параметров протокола, полученные при фиксированном отношении  $T_{ari}$ ,  $RT_{cal}$  и  $TR_{cal}$ . Можно видеть, что, длительность ответа может меняться на два порядка.

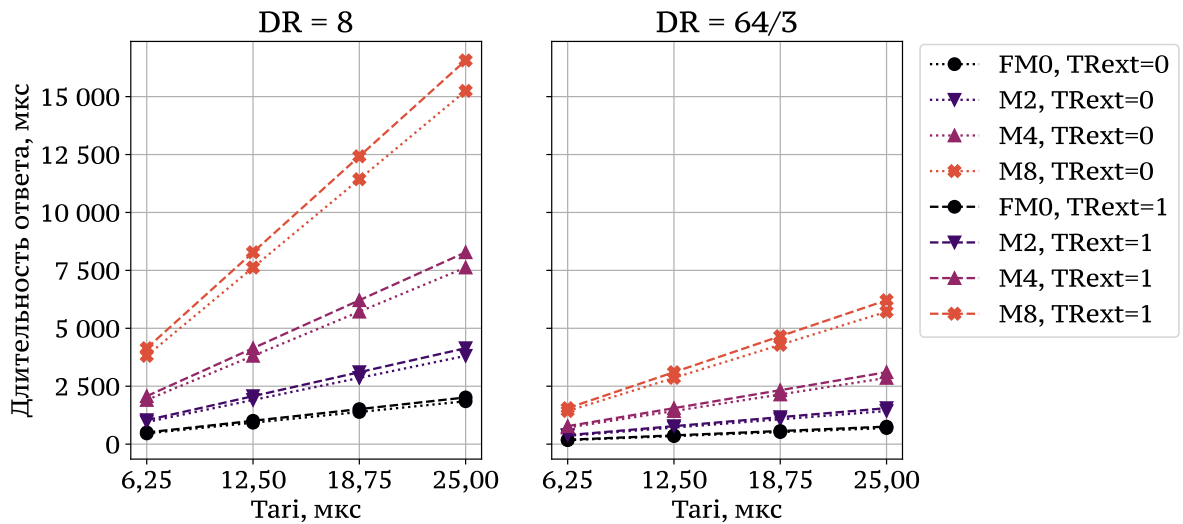


Рисунок 2.3 — Зависимость длительностей ответов EPC+PC+CRC от параметров протокола.

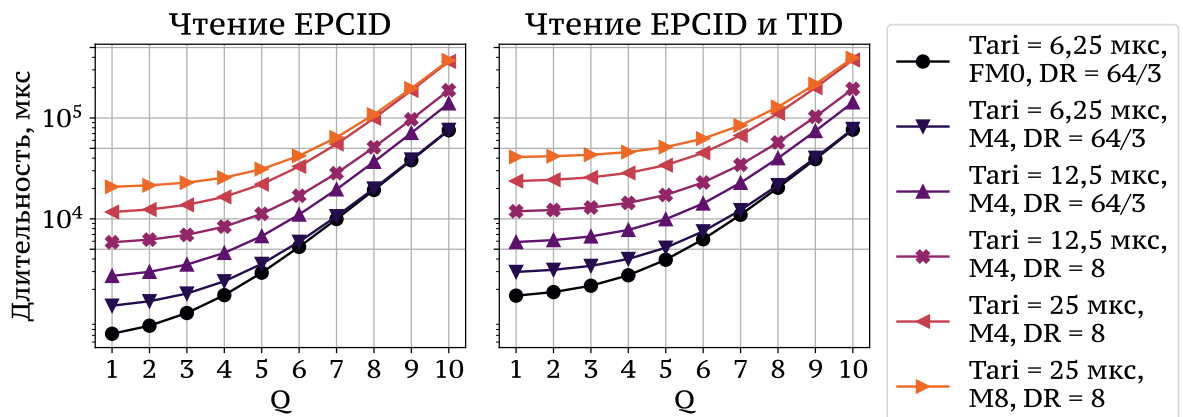


Рисунок 2.4 — Зависимость максимальной длительности раунда от значения  $Q$ .

При идентификации метки может использоваться только EPCID или комбинация EPCID и значений из других банков, например — TID. Первый вариант менее надежен, однако существенно быстрее второго — на чтение банков памяти требуется значительно больше времени, так как при этом необходимо осуществить дополнительный обмен сообщениями. На длительность идентификации также влияет, например, использование команды Select для настройки флагов меток и выбора подмножества меток для опроса. В дальнейшем будем рассматривать два самых простых сценария: идентификацию меток только по EPCID (в этом случае взаимодействие ограничивается инвентаризацией) и по комбинации EPCID + TID. Кроме того, будем предполагать, что считыватель не использует команды QueryAdjust и не изменяет выбранное значение параметра  $Q$ .

Из-за ошибок при передаче ответов, в общем случае метке может потребоваться несколько попыток для передачи своего идентификатора. Максимальное число раундов, в которых может принять участие метка, ограничено областью, в которой она получает достаточно энергии, и скоростью ее движения. Более подробно число раундов, с учетом коллизий и ошибок, анализируется в главе 3, а сейчас рассмотрим простейший модельный случай, когда в раунде передачу ведет единственная метка, и ей удается передать все свои сообщения. То есть в раунде нет коллизий, а ошибки при передаче ответов если и происходят, то

только на последних ответах. Длительность раунда  $\tau$  для этого модельного случая можно рассчитать по формуле (см. рис. 1.4):

$$\tau = t_1 + (2^Q - 1)t_0,$$

где

$$t_0 = T_{\text{QueryRep}} + T_1 + T_3 \text{ — пустой слот;}$$

$$t_1 = \begin{cases} t_1^{(\text{epc})} & \text{— слот с ответом при идентификации по EPCID;} \\ t_1^{(\text{tid})} & \text{— слот с ответом при идентификации по EPCID и TID;} \end{cases}$$

$$t_1^{(\text{epc})} = T_{\text{Query}} + 2 * (T_1 + T_2) + T_{\text{RN16}} + T_{\text{ACK}} + T_{\text{EPCID}};$$

$$t_1^{(\text{tid})} = t_1^{(\text{epc})} + 2 * (T_1 + T_2) + T_{\text{Req\_RN}} + T_{\text{Handle+TRead}} + T_{\text{Data}}.$$

Здесь  $T_1$  — интервал между концом команды и началом ответа,  $T_2$  — интервал между концом ответа и следующей командой,  $T_3$  — интервал между окончанием  $T_1$  и новой командой (ожидание ответа), а  $T_{\text{msg}}$  — длительность команды или ответа msg. Для простоты не учитываем задержку распространения сигнала, которая из-за небольшого расстояния крайне мала. На рис. 2.4 показаны зависимости длительностей раундов, рассчитанные по этим формулам, от значения Tag и величины  $Q$ .

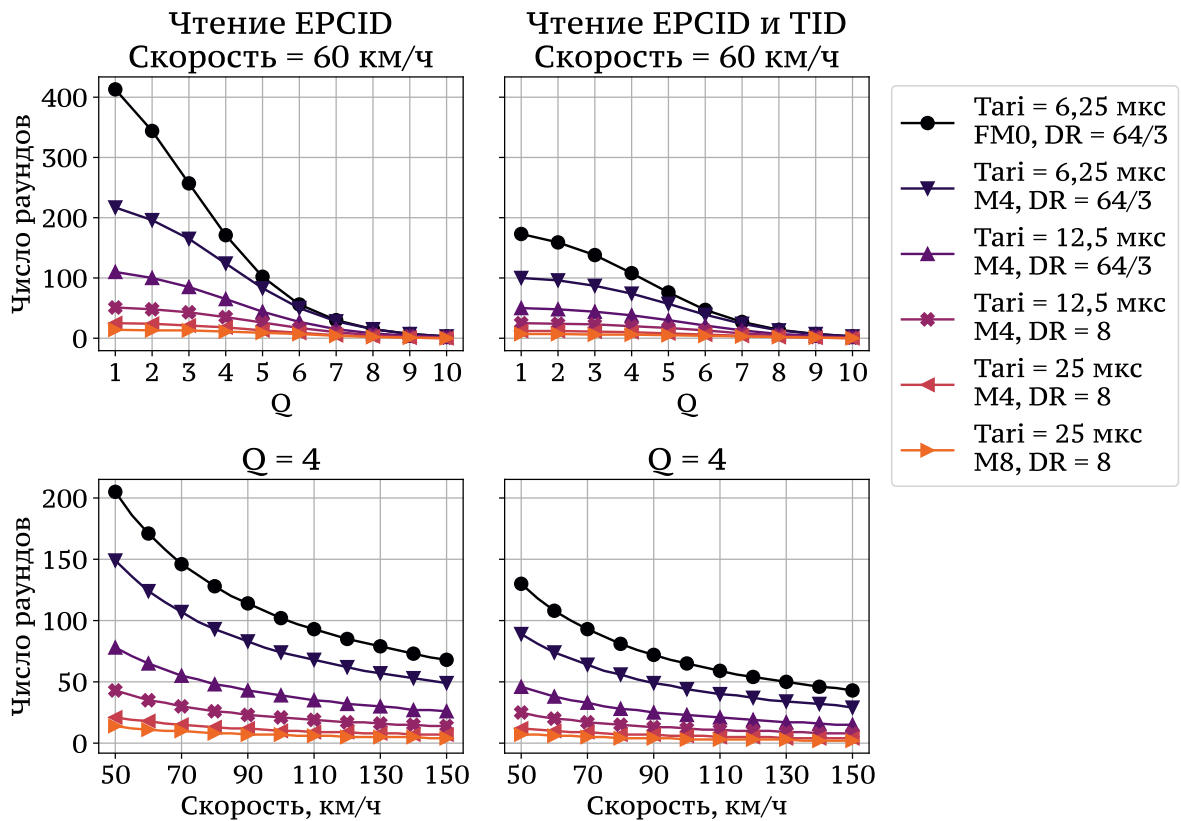


Рисунок 2.5 — Зависимость максимального числа раундов, в которых принимает участие метка, от значения  $Q$ .

Зная длительность раунда, скорость движения метки  $v$  и размер  $L$  области, в которой метка получает достаточно энергии, можно рассчитать максимальное число раундов, в которых метка может передать свой идентификатор, как  $N_r \approx \lfloor L/(\tau v) \rfloor$ . На рис. 2.5 показана

зависимость максимального числа раундов при  $L = 5$  метров от значения  $Q$  при  $v = 60$  км/ч, а также от скорости при  $Q = 4$ . При увеличении  $Q$  и росте скорости число раундов значительно снижается, причем тем быстрее, чем больше  $T_{\text{tag}}$  и  $M$ . В действительности число раундов будет еще меньше из-за переключения считывателя между антеннами, сбросах питания и возможного наличия других меток. Кроме того, метки, ошибшиеся при передаче EPCID и не получившие после этого команды NACK, не смогут принять участие в следующих раундах, если опрос будет проходить в той же сессии по тому же флагу. Более подробно эти вопросы будут исследованы в следующей главе.

## 2.4 Моделирование радиоканала между считывателем и меткой

Для передачи данных от метки используется модуляция обратного рассеяния, метка (не оснащенная источником питания) не может усилить отраженный сигнал. Из-за этого мощность сигнала, принятого на стороне считывателя, значительно ниже, чем мощность на стороне метки. По этой причине битовая ошибка (BER) оказывается выше на стороне считывателя, то есть ошибки с гораздо большей вероятностью появляются при приеме ответов от метки, чем команд от считывателя. Значение BER на стороне считывателя оказывает существенное влияние на успешность чтения меток, поэтому это значение должно быть аккуратно рассчитано.

Для анализа BER рассчитаем бюджет соединения и получим значение сигнал-шум (SNR). Результаты этого расчета для стационарного считывателя и метки, размещенной на переднем номере автомобиля, движущегося со скоростью 60 км/ч, приведены на рис. 2.7. На горизонтальной оси всех графиков показано расстояние между передатчиком и приемником, на вертикальной — время, прошедшее с момента достижения волной приемника. Как будет показано далее (см. формулу (2.2) для вычисления потерь на распространении сигнала), выбор значения скорости движения существенно влияет на скорость изменения величины затухания в заданной точке.

### 2.4.1 Расчёт бюджета соединений

Пусть  $P_t^{(r)}$  — излучаемая мощность считывателя,  $G^{(r)}$  — усиление антенны считывателя. Тогда эффективная изотропная излучаемая мощность (EIRP) составит  $EIRP = P_t^{(r)}G^{(r)}$ . Здесь и далее надстрочный индекс используется для обозначения устройства: (r) для считывателя и (t) для метки; подстрочный индекс используется для обозначения направления передачи (прием или передача сигнала). При распространении сигнала от считывателя к метке сигнал испытывает затухание  $A_{pl}^{(d)}$ , зависящее от состояния радиоканала и взаимно-



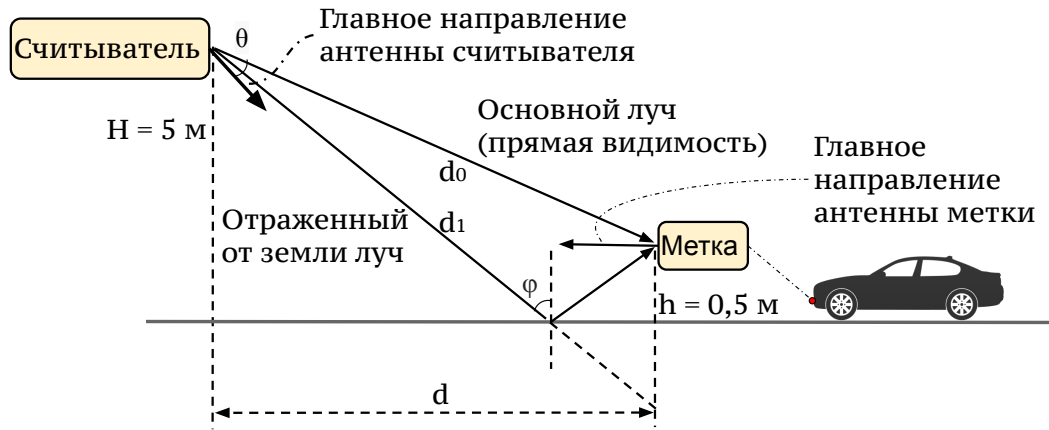


Рисунок 2.6 — Схема системы радиочастотной идентификации автомобилей и ее геометрические параметры

го расположения считывателя и метки друг относительно друга. Антенны устройств могут иметь различную поляризацию, что ведет к дополнительным потерям  $A_{pol}$ . Пусть усиление антенны метки равно  $G^{(t)}$ . Тогда мощность сигнала, принимаемого меткой, равна:

$$P_r^{(t)} = P_t^{(r)} G^{(r)} A_{pl}^{(d)} A_{pol} G^{(t)}.$$

Если эта мощность меньше чувствительности метки  $P_s^{(t)}$ , то метка не включится и не сможет взаимодействовать со считывателем. В противном случае метка сможет передавать свои ответы за счет модулирования отраженного сигнала, мощность которого будет равна  $P_t^{(t)}$ . Поскольку при этом возникают дополнительные энергетические потери (например, из-за модуляции), равные  $A_{bs}$ , то мощность принятого и отраженного сигналов связаны как  $P_t^{(t)} = P_r^{(t)} + A_{bs}$ .

Для мощности сигнала, принятого от метки считывателем, можно написать следующее соотношение:

$$P_r^{(r)} = P_r^{(t)} G^{(t)} A_{bs} A_{pl}^{(r)} A_{pol} G^{(r)}.$$

В общем случае потери на прямом (от считывателя к метке) и обратном (от метки к считывателю) путях могут отличаться. Это происходит из-за различия в поляризации (обычно на считывателях используются антенны с круговой поляризацией, а на метках — с линейной) и, как следствие, в коэффициенте отражения от земли.

В дальнейших расчетах бюджета соединения будут использованы значения, приведенные в таблице 2, типичные для оборудования, применяемого в идентификации автомобилей. При расчете также следует учитывать потери в кабеле, соединяющем антенну со считывателем. Далее считается, что антенны считывателя имеют круговую поляризацию, а меток — линейную, таким образом потери на поляризации составляют -3 дБ.

На рис. 2.7 показан расчет мощностей сигналов, принятых и переданных меткой, принятого считывателем сигнала, а также величина затухания. Все кривые убывают немонотонно из-за изменений с расстоянием в коэффициентах отражения от земли, а также из-за влияния диаграмм направленности антенн. Особенно сильно диаграмма направленности влияет

Таблица 2 — Параметры, использованные при расчете бюджета соединения

Параметр	Значение
Мощность, излучаемая считывателем, $P_t^{(r)}$	31,5 дБм
Усиление антенны считывателя, $G^{(r)}$	8 дБи
Усиление антенны метки, $G^{(t)}$	2 дБи
Чувствительность метки, $P_s^{(t)}$	-18 дБм
Потери на поляризации, $A_{pol}$	-3 дБ
Потери на модуляции на метке, $A_{bs}$	-10 дБ
Потери в кабеле, $A_{bs}$	-2 дБ

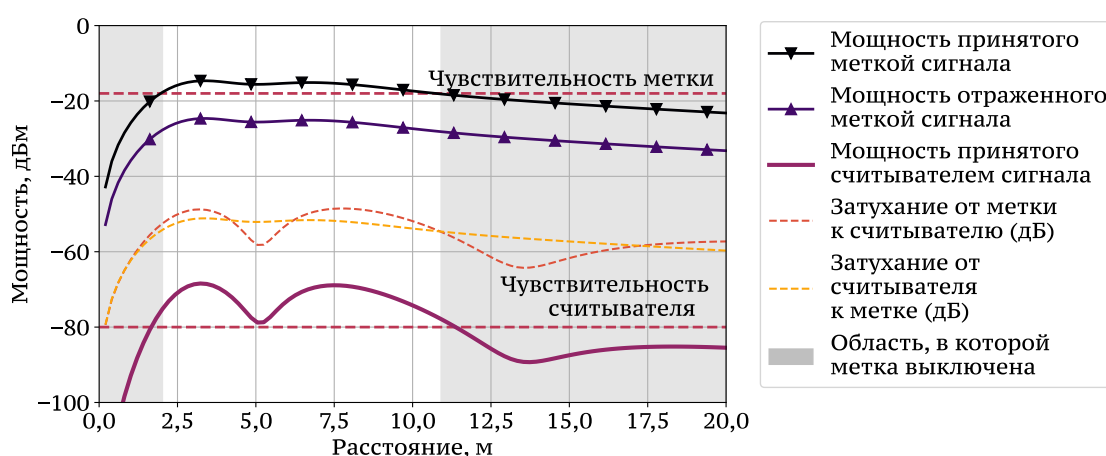


Рисунок 2.7 — Расчет бюджета соединения в зависимости от расстояния между считывателем и меткой.

на сигнал при близком расположении метки и считывателя, так как даже небольшое изменение расстояния ведет к ощутимому изменению угла падения. Для того, чтобы связь между расстоянием от считывателя до метки и временем при постоянной скорости оставалась линейной, в качестве расстояния рассматривается дистанция от опоры, на которой размещен считыватель, до метки, измеренная вдоль дороги (см. величину  $d$  на рис. 2.6).

#### 2.4.2 Расчёт мощности принятых сигналов

При движении меток и считывателей с ненулевой скоростью друг относительно друга сигналы оказываются подверженными эффекту Доплера, проявляющемуся в сдвиге частоты сигнала на величину, называемую доплеровским сдвигом, и равную:

$$\alpha = 1 - \frac{v \cos \psi}{c + v \cos \psi},$$

где  $c$  — скорость света,  $\mathbf{v}$  — скорость приёмника относительно передатчика, в роли приёмника может выступать как метка, так и считыватель (при этом их скорости будут иметь одинаковую величину и противоположные знаки),  $\psi$  — угол между волновым вектором  $\mathbf{k}$  и направлением движения,  $f_c$  — несущая частота.

Для узкополосного сигнала в RFID-канале смещённую частоту можно приблизить выражением  $\alpha f \approx f - \frac{\mathbf{v}}{c} \cos \psi \cdot f_c = f - \nu$ , где  $\nu$  — доплеровский сдвиг:

$$\nu = \frac{\mathbf{v}}{c} f_c \cos \psi = \frac{1}{2\pi} (\mathbf{v}, \mathbf{k}).$$

При распространении сигнала он может отражаться от дороги, инженерных конструкций, других автомобилей и прочих объектов, то есть в моделируемой системе имеет место многолучевое распространение сигнала. При этом возникает множество копий сигнала  $s(t)$ , каждая из которых испытывает собственное затухание  $h_i$ , задержку  $\tau_i$  и доплеровский сдвиг  $\nu_i$ , поскольку каждая копия распространяется по собственному пути, с которым, помимо длины, связаны углы передачи и приема сигнала. Пусть в сигнале присутствует прямая компонента (Line-of-Sight, LoS) сигнала и  $N$  отраженных компонент (или лучей). Без потери общности рассуждений будем считать, что LoS-компонента описывается нулевым лучём, а отраженные компоненты — лучами  $1 \dots N$ . Принятый сигнал определяется следующей формулой:

$$r(t) = \sum_{i=0}^N h_i s(t - \tau_i) e^{j\nu_i t}. \quad (2.1)$$

Затухание  $h_i$  может быть рассчитано как  $h_i = \lambda / (4\pi d_i) R_i \Gamma_i$ , где  $\lambda$  — длина волны,  $d_i$  — длина пути для  $i$ -го луча,  $R_i$  — коэффициент затухания, вызванного отражением сигнала, для  $i$ -го луча (для основной, LoS-компоненты полагаем равным единице:  $R_0 = 1$ ),  $\Gamma_i = \Gamma_i^{(r)} \Gamma_i^{(t)}$  — ослабление сигнала, определяемое диаграммами направленности передающей и принимающей антенн.

Рассмотрим в качестве передаваемого сигнала постоянную волну (синусоидальный сигнал) единичной мощности. После подстановки формул, описанных выше, в выражение для  $r(t)$  (2.1), получим затухание при распространении сигнала как значение мгновенной мощности на приемнике:

$$A_{pl} = |r(t)|^2 = \left( \frac{\lambda}{4\pi} \right)^2 \left| \sum_{i=0}^N \frac{R_i \Gamma_i}{d_i} e^{-jk(d_i - vt \cos \psi_i)} \right|^2. \quad (2.2)$$

Расчёт  $R_i$ ,  $\Gamma_i$ ,  $d_i$  и  $\psi_i$  требует определения путей распространения сигнала по разным лучам. Для статического окружения, при пренебрежении отражениями от движущихся автомобилей, пути могут быть вычислены простыми аналитическими методами. При более сложном динамическом окружении (а также при моделировании отражающих объектов сложной формы) потребуется использовать техники трассировки лучей. Далее, для упрощения расчетов, будем учитывать только две компоненты: прямую (LoS) и отраженную от земли (NLoS) компоненты. Для отраженной NLoS-компоненты коэффициент затухания  $R_1$

равен коэффициенту отражения от земли. Далее предполагаем, что антенна считывателя размещена на высоте 5 м, а метки — 0,5 м над дорогой. Остальные лучи моделируются явно, как случайные компоненты, обуславливающие использование распределение Рэлея при вычислении BER (подробнее об этом — в следующем подразделе).

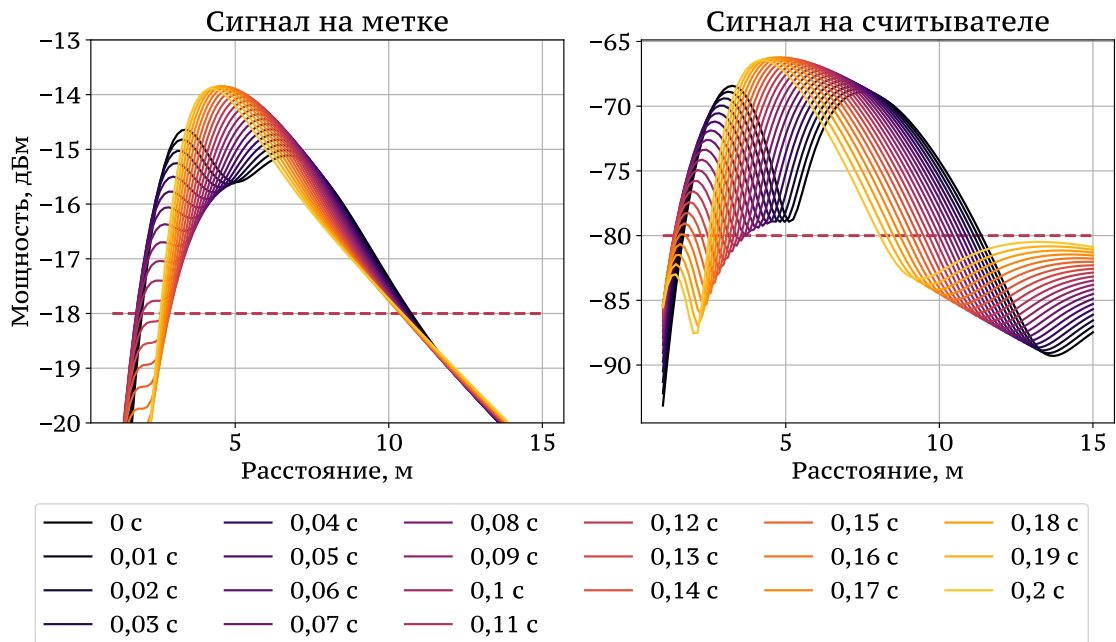


Рисунок 2.8 — Зависимость мощности сигнала, принятого считывателем, от расстояния между считывателем и меткой  $d$  и временем, прошедшим с начала приема меткой сигнала от считывателя

Наличие эффекта Доплера делает канал зависимым от времени (см. [190]), поскольку, как видно из формулы (2.1), величина сдвига для каждого луча определяется индивидуально и зависит от времени, прошедшего с начала получения сигнала. Особенность RFID-системы состоит в том, что считыватель не перестает передавать сигнал (постоянный, который метка может модулировать для передачи своего ответа, или информационный) ни после передачи своей команды, ни после получения ответа. Считыватель может прекращать передачу сигнала, например, для сброса флагов сессий меток, при переключении частот или в том случае, если по законодательству он должен периодически освобождать радиоканал. Однако, такие прерывания происходят относительно редко, внутри одного или нескольких раундов сигнал не прерывается. Поэтому значение  $t$  в (2.1) определяется как время, прошедшее со включения считывателя, за вычетом длительности распространения сигнала от считывателя до метки. Хотя метка в это время может еще находиться далеко, радиоканал уже будет существовать. Учитывая, что в момент включения считывателя метки расположены на разных расстояниях от считывателя, получаем, что значение  $t$  в одной и той же точке для разных меток будет разным, а соответственно разными будут и значения  $r(t)$ . На рис. 2.8 показана зависимость мощностей сигналов, принятых меткой и считывателем, от расстояния  $d$ , измеренного вдоль дороги. Различные кривые соответствуют времени, прошедшему с начала приема меткой сигнала от считывателя.

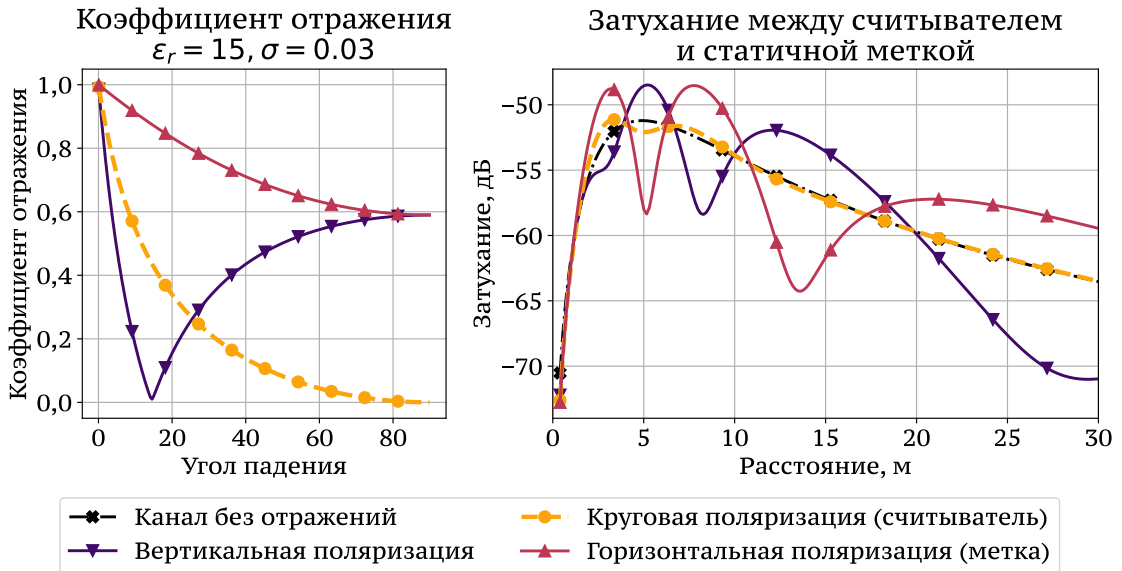


Рисунок 2.9 — Коэффициент отражения и затухание сигнала для параллельной, перпендикулярной и круговой поляризаций

На рис. 2.9 показаны результаты расчета затухания сигнала в зависимости от расстояния для разных видов поляризации: параллельной, перпендикулярной и круговой. Различие обусловлено тем, что коэффициент отражения от земли  $R_1$  для NLoS-компоненты сигнала принимает разные значения в зависимости от выбранной поляризации. Коэффициент отражения определен следующей формулой [18]:

$$R = \frac{\sin \varphi - \sqrt{C}}{\sin \varphi + \sqrt{C}},$$

где  $\varphi$  — угол падения,  $C = \eta - \cos^2 \varphi$  для горизонтально поляризованной компоненты и  $C = \frac{\eta - \cos^2 \varphi}{\eta^2}$  для вертикально поляризованной;  $\eta = \epsilon_r(f) - j60\lambda\sigma(f)$ , где  $\epsilon_r(f)$  — относительная диэлектрическая проницаемость поверхности для заданной частоты  $f$ ,  $\sigma(f)$  — проводимость.

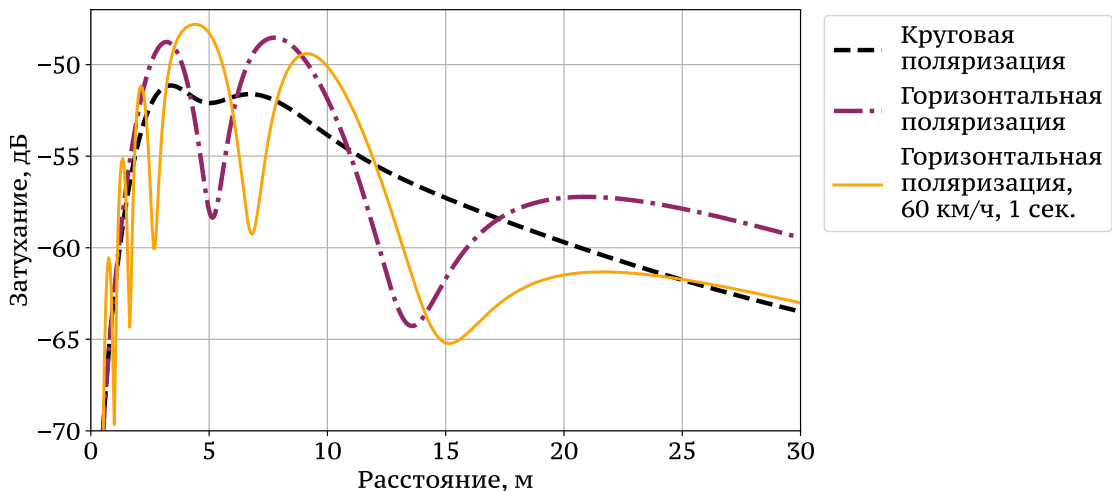


Рисунок 2.10 — Зависимость коэффициента отражения от угла падения для параллельной, перпендикулярной и круговой поляризации при разных значениях относительной диэлектрической проницаемости  $\epsilon$  и проводимости  $\sigma$

Относительная диэлектрическая проницаемость и проводимость сильно зависят от влажности поверхности. Проводимость  $\sigma$  меняется в диапазоне от 0.00014 См/м для очень сухой земли до 5 См/м для морской воды. Относительная диэлектрическая проницаемость  $\epsilon_r$  меняется в диапазоне от 3 до 70. В дальнейшем будем использовать значения проводимости  $\sigma = 0,03$  См/м и диэлектрической проницаемости  $\epsilon_r = 15$ , характерные для сухой дороги с твердым покрытием.

Одновременный учет коэффициента отражения и эффекта Доплера делают зависимость затухания от расстояния достаточно сложной. На рис. 2.10 показан пример расчета величины затухания для круговой и горизонтальной поляризаций в момент времени  $t = 0$  и через 1 секунду после включения считывателя.

Наконец, для расчета мощностей сигналов необходимо учитывать диаграммы направленности антенн. Будем считать, что у метки простая дипольная антенна, и её диаграмма направленности определяется как:

$$\Gamma(\theta) = \left| \frac{\cos \frac{\pi}{2} \sin \theta}{\cos \theta} \right|. \quad (2.3)$$

Для простоты, будем считать, что диаграмма антенны считывателя также определяется с помощью (2.3). Для более точного расчета можно использовать более подходящие диаграммы, например — патч-антенны [191].

### 2.4.3 Расчёт вероятности битовой ошибки (BER)

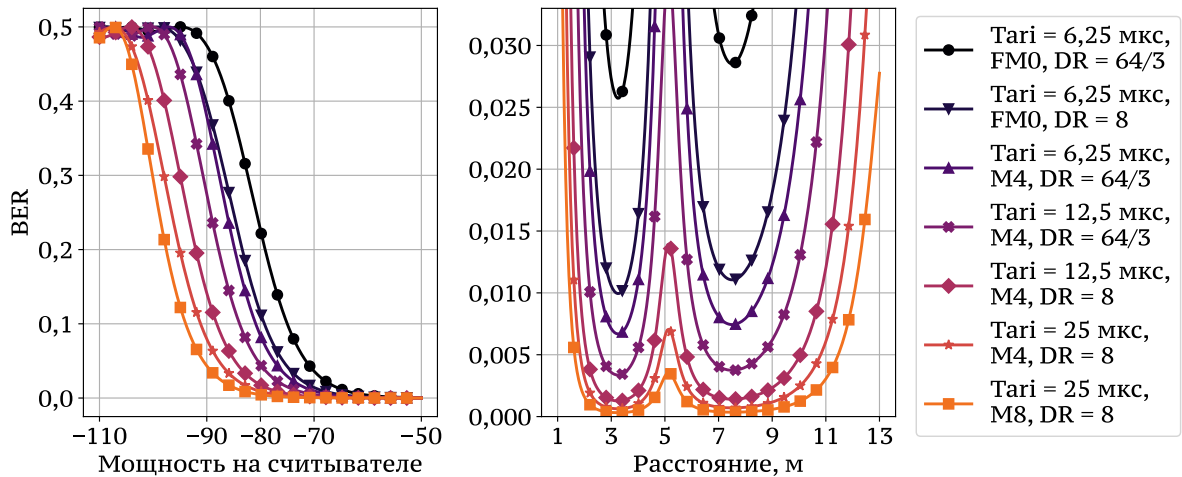


Рисунок 2.11 — Вероятность битовой ошибки (BER) для разных параметров  $T_{\text{ari}}$  и  $M$

Рассмотрим радиоканал с аддитивным гауссовским шумом (AWGN). Вероятность ошибки в передаче бита (BER) можно выразить следующей формулой:

$$P_{er} = 2Q(\gamma)[1 - Q(\gamma)], \quad (2.4)$$

где  $Q(\cdot)$  — Q-функция;  $\dot{\gamma} = mE_s/N_0 \cos^2 \varphi_s$ ;  $m$  — число символов на бит (порядок) в коде Миллера,  $E_s$  — энергия одного символа,  $N_0/2$  — спектральная плотность белого шума и  $\varphi_s$  — разность между реальной и принятой фазами сигнала. Соотношение  $E_s/N_0$  может быть выражено как  $\gamma T_s B$ , где  $\gamma$  отношение сигнал-шум (SNR),  $T_s$  — длительность символа и  $B$  — ширина полосы. Ошибку синхронизации фазы будем приближенно моделировать как  $\cos^2 \varphi_s \approx 1/\gamma T_{pr} B$ , где  $T_{pr}$  — длительность преамбулы [186].

Использование гауссовской модели оказывается слишком оптимистичным, так как при расчете мощности принятого сигнала учитываются не все лучи. Далее будет использоваться выражением, получаемым усреднением  $P_{er}$  в формуле (2.4) по  $\dot{\gamma}$ , которую считаем случайной величиной, имеющей распределение Рэлея [61]:

$$P_{er} = \frac{1}{2} - \frac{1}{\sqrt{1 + \frac{2}{\dot{\gamma}}}} + \frac{2}{\pi} \frac{\arctan \sqrt{1 + \frac{2}{\dot{\gamma}}}}{1 + \frac{2}{\dot{\gamma}}}.$$

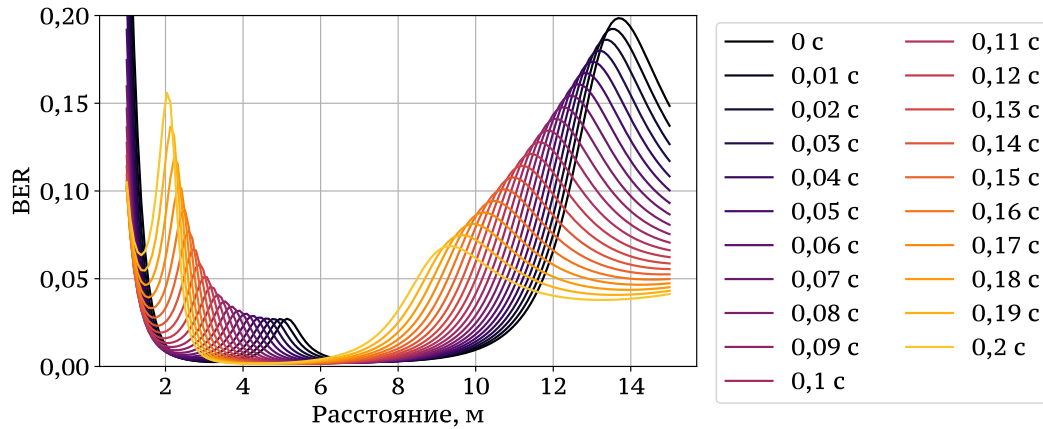


Рисунок 2.12 — Изменение вероятности битовой ошибки (BER) из-за эффекта Доплера.

На рис. 2.11 показана зависимость BER от выбранного типа кодирования ответа метки. Как и следовало ожидать, при росте числа символов на бит  $m$  и уменьшении BLF (при увеличении  $T_{ar}$  и уменьшении DR) BER снижается. Так, при  $T_{ar} = 6,25$  мкс, коде FM0 ( $m = 1$ ) и DR = 64/3 BER не опускается ниже 0,025, в то время как при  $T_{ar} = 25$  мкс, коде Miller-8 ( $m = 8$ ) и DR = 8 BER оказывается ниже 0,001 в области размером около 5 метров. При этом число попыток передать свой идентификатор у метки, то есть число раундов, зависит от тех же параметров противоположным образом (см. рис. 2.5).

Для успешной передачи EPCID метке нужно передать без ошибок RN16 (16 бит) и ответ на АСК, содержащий EPCID, PC (16 бит) и CRC (16 бит). Полагая длину EPCID, равной 96 битам, в общей сложности метка должна передать 144 бита. При BER = 0,01 вероятность успешной передачи составит 23,5%, при BER = 0,005 — 48,8%, а при BER = 0,001 — 86,5%. Учитывая, что метка может принять участие только в небольшом количестве раундов (см. рис. 2.5), нужно выбирать такие параметры протокола, при которых область с низким BER будет достаточно большой, и при этом метка должна успеть совершить несколько попыток

передачи идентификатора. Отметим, что из-за немонотонной зависимости BER от расстояния, такая область разбивается на «окна», а из-за эффекта Доплера (см. рис. 2.12) эти «окна» смещаются с течением времени от включения считывателя.

## 2.5 Результаты имитационного моделирования

Для проведения численного эксперимента была разработана имитационная модель на языке Python. Исходный код модели, документация и собранные данные можно найти в Git-репозитории на GitHub: <https://github.com/larioandr/thesis-rfidsim>.

Разработанная дискретно-событийная имитационная модель — достаточно гибкая и простая, позволяет легко вносить изменения для адаптации к другим экспериментам. Состояние модели изменяется при наступлении различных событий — начала и окончания приема сигнала, окончания ожидания сообщения, обновления положения меток, таймаута для переключения антенны считывателя, генерации нового автомобиля с меткой и прочих. Для того, чтобы учесть изменение мощностей сигналов при движении метки, затухания и мощности пересчитывались периодически при обновлении положения метки (каждую миллисекунду модельного времени), и записывались для дальнейшего определения минимального уровня сигнала и расчета BER. Более подробное описание модели можно найти в репозитории. Главный недостаток модели — высокая вычислительная сложность. Так, для получения численных результатов, представленных в диссертационном исследовании, потребовалось около восьми часов выполнения модели на четырехядерном сервере. Модель имеет более 50 параметров, в том числе:

- настройки протокола: интервалы  $T_{ari}$ ,  $R_{Tcal}$ ,  $T_{Rcal}$ , метод кодирования ответов меток ( $M$ ), коэффициент  $DR$ , использовать ли расширенную преамбулу ( $TR_{ext}$ );
- параметры мобильности меток: скорость движения, частота генерации;
- параметры окружения: число и ширина полос движения, длина автомобилей, относительная диэлектрическая проницаемость и проводимость дорожного покрытия, уровень шума;
- параметры считывателя: мощность передатчика, потери в кабеле, поляризация, усиление, диаграмма, высота и угол наклона антенн, порядок выбора флагов сессии для опроса;
- параметры меток: высота установки, поляризация, усиление и диаграмма антенн, потери на отражении сигнала, длина EPCID и TID, чувствительность;
- параметры канала: учитывать ли эффект Доплера, модель для расчета BER.

В ходе численного эксперимента были получены оценки идентификации автомобилей, на которых были установлены RFID-метки. Автомобиль считался идентифицированным, если была успешно прочитана хотя бы одна метка в переднем или заднем номере. Кроме того, для более глубокого понимания работы системы, было исследовано число раундов, в которых успевает принять участие движущаяся метка.



### 2.5.1 Анализ влияния частоты переключений антенн на число раундов

Точное число раундов, в которых успевает принять участие метка, зависит как от настроек протокола (схема кодирования ответов метки, выбор параметра  $Q$  и пр.), так и от алгоритма выбора сессии и частоты смены антенн. Так как вероятность прочесть и EPCID, и TID в одном раунде относительно мала, существенным оказывается то, в скольких раундах метка успевает принять участие, т.е. сколько у нее будет попыток передать свои идентификаторы.

В начале раунда инвентаризации считыватель выбирает сессию ( $S_0, S_1, S_2, S_3$ ) и значение её флага ( $A$  или  $B$ ). Метка отвечает только в том случае, если хранимый ею флаг выбранной сессии совпадает с переданным в команде Query. После рассылки своего EPCID в ответе на АСК, метка инвертирует значение хранимого флага сессии ( $A \rightarrow B, B \rightarrow A$ ). Для каждой сессии стандарт определяет интервал времени после выключения питания, в течение которого метка должна сохранять значение флага сессии, а после которого, если питания так и не появилось, — возвращать значение  $A$ . В дальнейшем будем рассматривать сессию  $S_0$ , для которой эта процедура самая простая — значение  $A$  должно быть установлено сразу после сброса питания, независимо от длительности потери энергии.

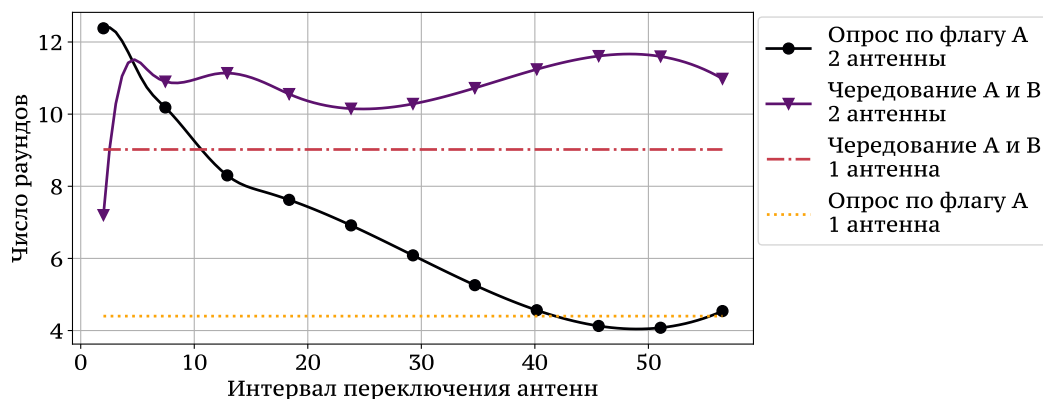


Рисунок 2.13 — Число раундов, в которых участвует метка, в зависимости от длительности работы на одной антенне, для разных стратегий выбора флага сессии.

Если считыватель использует одну антенну без периодического выключения питания, число раундов, в которых участвует метка, зависит только от стратегии выбора флага сессии. На рис. 2.13 приведены результаты симуляции, которые показывают, что число раундов при опросе меток только со значением флага сессии  $S_0 = A$  (самая нижняя горизонтальная линия) и число раундов при смене значений запрашиваемых флагов каждый раунд отличаются в шесть раз. В первом случае число раундов, в которых принимает участие метка, определяется только числом включений метки (выключения обусловлены приемом сигнала считывателя на уровне, меньшем чувствительности метки). При периодической смене флагов метка имеет возможность повторно участвовать в опросе и без выключения, поэтому число раундов оказывается значительно больше, а тем самым повышается вероятность того, что хотя бы один раз метка успешно передаст свои идентификационные данные. Небольшие

колебания на приведенном графике обусловлены случайным характером процесса чтения меток, а также случайностью длительности раундов, зависящей от числа занятых и пустых слотов. Также следует отметить, что на число раундов может влиять выбор параметра  $Q$  (график соответствует значению  $Q = 2$ ), а также использование операции Select, которая в представленном исследовании не использовалась.

Если считыватель использует несколько антенн и периодически между ними переключается, число раундов также зависит от интервала между переключениями антенн. На рис. 2.13 показаны результаты, полученные исходя из того, что считыватель размещен над однополосной дорогой и оборудован двумя антеннами, направленными в противоположные стороны (для чтения переднего и заднего номеров). Можно видеть, что периодическая смена флагов сессии также позволяет увеличить число раундов, хотя и меньше, чем при использовании одной антенны. Можно заметить, что сильнее всего выбор интервала переключения антенн влияет на число раундов при малых значениях интервала. Это связано, в первую очередь, с тем, что при этом переключение происходит каждый раунд или быстрее, метка может не успеть передать свой EPCID до потери энергии (здесь предполагается, что переключение антенн не связано с границами раунда, что является некоторым упрощением). Это связано, в первую очередь, с тем, что периоды работы на одной антенне быстро становятся сопоставимыми с длительностью проезда зон с уровнем сигнала от считывателя, превышающим чувствительность меток. При увеличении мощности сигнала или чувствительности меток зависимость должна оказываться более сильной.

Следует отметить, что при использовании других сессий (например, S2), результаты оказываются ближе к случаю отсутствия переключения антенн для малых интервалов, так как при малых интервалах между переключениями (меньших, чем время сохранения флага сессии) метка может «не заметить» потерю энергии и сохранить значение флага.

На число раундов также влияет длительность работы считывателя до его выключения и время нахождения в выключенном состоянии. Все расчёты в этом и следующем разделе сделаны в предположении, что считыватель выключается каждые 2 секунды на 100 миллисекунд.

### 2.5.2 Анализ вероятности идентификации транспортных средств

С помощью имитационной модели была изучена вероятность идентификации меток, расположенных на номерах автомобилей, движущихся по однополосной дороге. Считыватель был оборудован двумя антеннами, размещенными над дорогой в противоположных направлениях, для чтения переднего и заднего знаков.

Для выбора параметра  $Q$  было рассчитано среднее число меток, участвующих в одном раунде. Результаты этого расчета приведены в табл. 3, где  $N_v$  — среднее число автомобилей вблизи считывателя,  $T_v$  — средний интервал времени между появлениями автомобилей,  $N_t$  — среднее число меток, участвующих в раунде, а  $N_t^{(1)}$  — среднее число меток в тех раундах, в которых участвует хотя бы одна метка. Из приведенных результатов видно, что даже

при очень интенсивном потоке, когда в зоне действия считывателя оказывается более пяти машин, из-за неравномерности уровня сигнала большая часть меток оказывается выключенными, и можно считать, что в каждом раунде участвует не более одной метки. Фактически это означает, что вероятностью коллизий можно пренебречь и выбирать значение  $Q$  сколь угодно малым, так как его увеличение лишь добавит пустых слотов. Учитывая, что длительность пустого слота достаточно мала и для перестраховки на тот случай, когда помимо меток в номерах в зону действия попадут другие метки (расположенные, например, на предметах под стеклом автомобиля или на самом автомобиле), можно установить  $Q=2$ . В дальнейшем это значение будет использовано для получения остальных результатов.

Таблица 3 — Среднее число меток, участвующих в раунде, при движении автомобилей со скоростью 60 км/ч

$T_v$ , sec	$N_v$	$N_t$	$N_t^{(1)}$
0.5	5.67	0.1000	1.0
1.0	3.38	0.0357	1.0

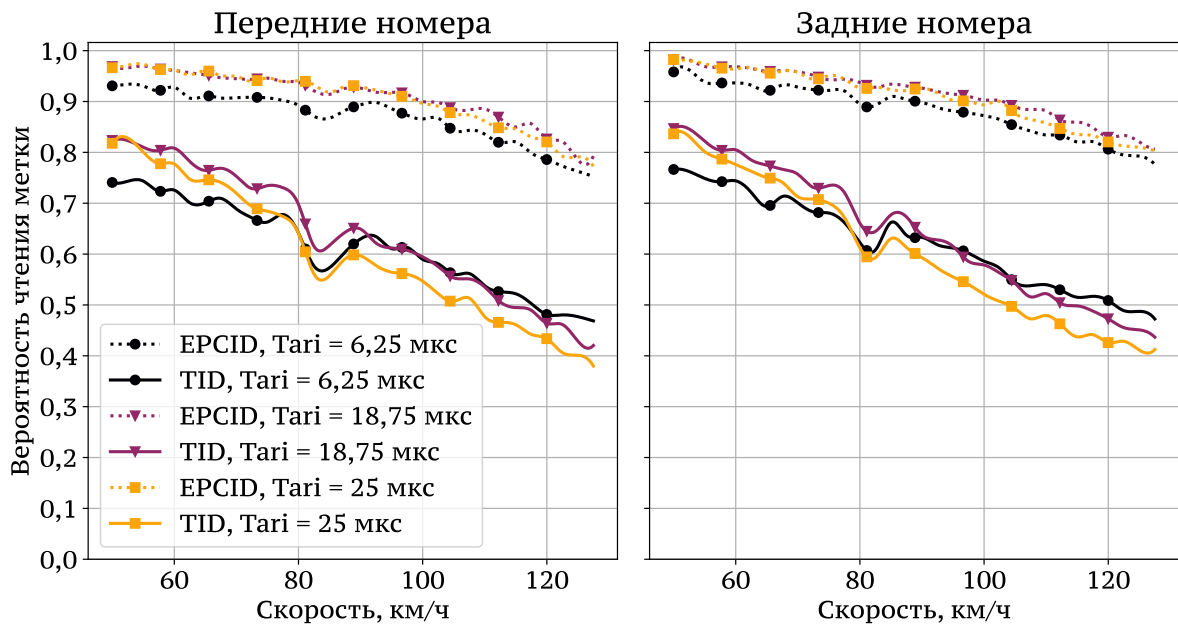


Рисунок 2.14 — Вероятность успешного чтения меток в номерах при различных значениях  $T_{\text{tag}}$  и кодировании ответов  $M = 4$

На рис. 2.14 и 2.15 показаны результаты расчёта вероятности идентификации меток в номерах в зависимости от скорости движения автомобилей при различных значениях интервала  $T_{\text{tag}}$  (см. рис. 2.14) и различных значениях  $M$  (число символов на бит в ответах меток, см. рис. 2.15). Предполагалось, что антенны считывателя имеют усиление 8 дБи, потери в кабеле составляют 2 дБ, метки используют расширенные преамбулы, а значение  $DR=8$ . Считыватель работал в сессии  $S_0$ , инвертировал значение флага сессии в каждой команде Query и переключал антенны каждые 100 мс. Из приведенных графиков видно, что вероятность

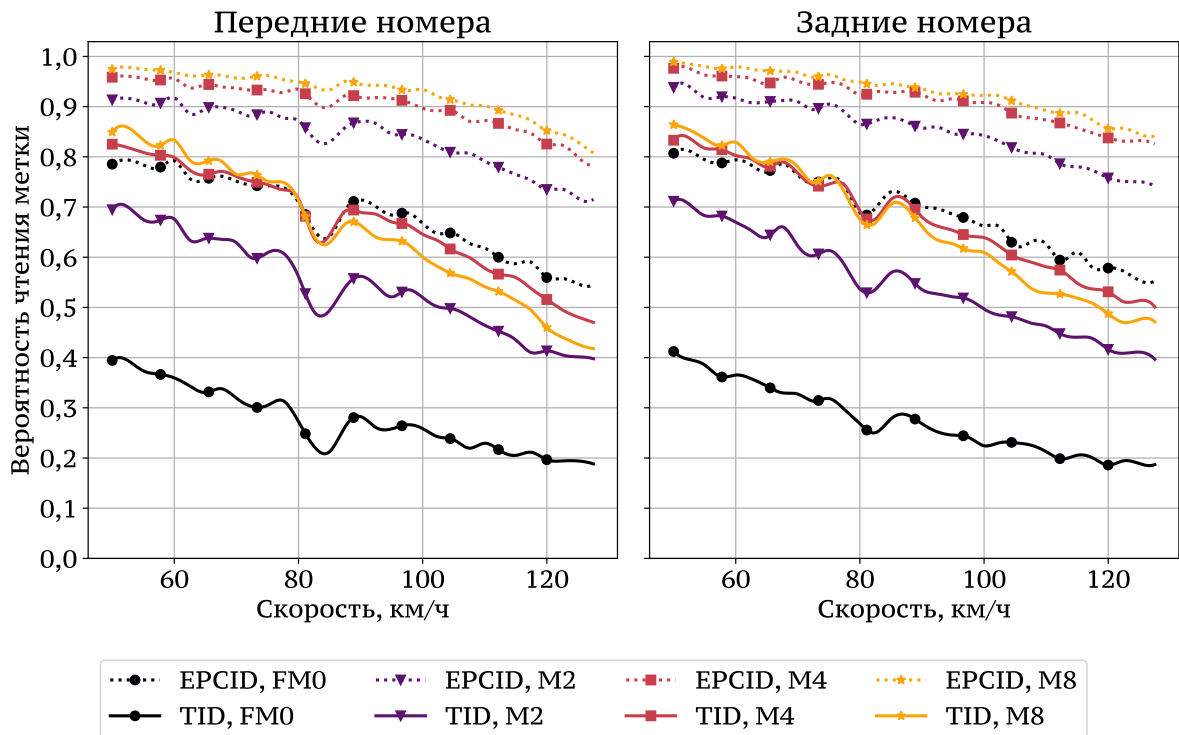


Рисунок 2.15 — Вероятность успешного чтения меток в номерах при различных значениях  $M$  и  $T_{ar1} = 12,5$  мкс

чтения меток падает с ростом скорости. При этом вероятность чтения EPCID остаётся высокой при достаточно больших скоростях свыше 120 км/ч, чего нельзя сказать про чтение TID. Важно отметить, что выбор больших значений  $T_{ar1}$  не обязательно ведёт к увеличению вероятности успешной идентификации (см. рис. 2.14), также как и выбор больших значений  $M$  (см. рис. 2.15). Более того, при чтении TID выбор самых больших значений  $T_{ar1} = 25$  мкс и  $M = 8$  даёт результаты хуже, чем выбор менее надёжных значений  $T_{ar1} = 18,75$  мкс и  $M = 4$ , хотя выбор ещё меньших значений также снижает вероятность успешного чтения. Объяснить эту закономерность можно тем, что выбор слишком больших значений ведёт к увеличению длительности раундов и, как следствие, снижению вероятности успешной идентификации хотя бы в одном раунде, хотя и повышает вероятность успешного чтения TID в одном раунде. Дальнейшее же уменьшение значений  $M$  и  $T_{ar1}$  ведёт к слишком сильному уменьшению вероятности успешной передачи в одном раунде. Этот вывод подтверждается также тем, что выбор оказывается менее существенным при чтении только EPCID, так как и увеличение длительности раунда там оказывается менее значительным, и влияние вероятности успешной передачи ответов метки меньше зависит от BER (так как в этом случае ответов, которые должна передать метка, в два раза меньше).

Также было проведено численное исследование влияния эффекта Доплера, см. рис. 2.16. Как видно из приведённых результатов, эффект Доплера оказывает существенное влияние на вероятность чтения меток, особенно при чтении TID на высоких скоростях — в этом случае эффект Доплера может снижать вероятность успешного чтения метки на 10–20 %.

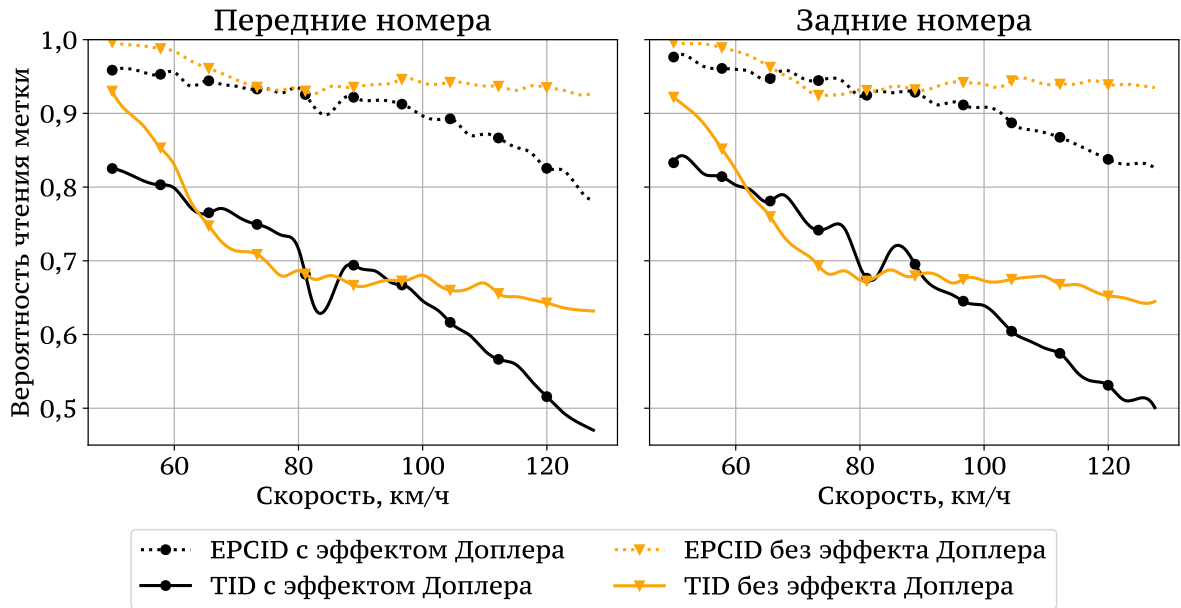


Рисунок 2.16 — Влияние эффекта Доплера на вероятность успешного чтения метки при  $M = 4$ ,  $T_{ar1} = 12,5$  мкс

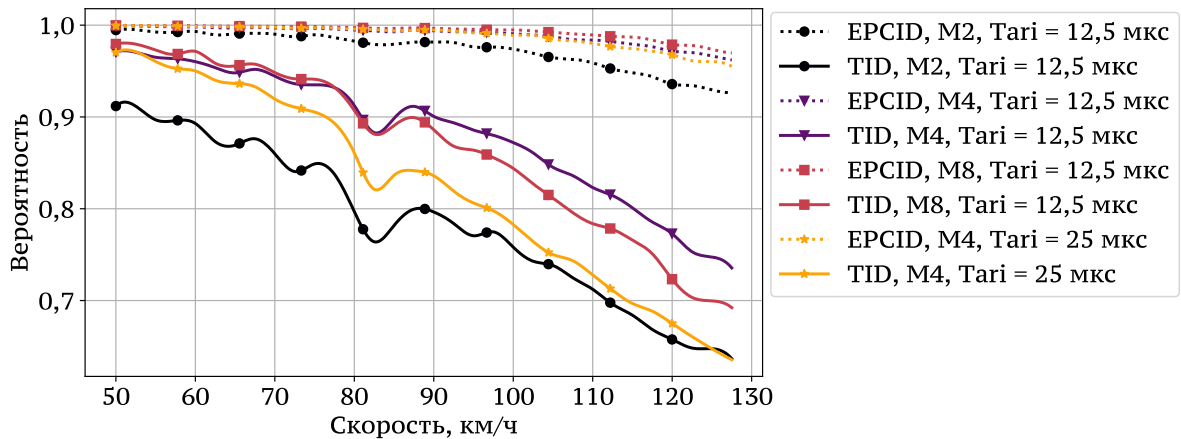


Рисунок 2.17 — Вероятность идентификации автомобиля с метками на переднем и заднем номере, идентифицируемого по любой из меток

Результаты расчета вероятности идентификации автомобилей показаны на рис. 2.17. При проведении расчётов считалось, что автомобиль может быть идентифицирован по любой метке, т.е. для успешной идентификации было достаточно единожды прочесть метку в переднем или заднем номере. Полученные результаты совпадают с данными, полученных в ходе эксперимента в городе Казань (см. главу 5), в котором использовалось значение  $M=4$  и  $T_{ar1} = 12,5$  мкс, а вероятность идентификации составила около 92–95%, в зависимости от точки идентификации, при идентификации меток по EPCID и TID. Из приведенных результатов видно, что эти значения дают наилучшую вероятность идентификации быстро движущихся автомобилей. Также можно видеть, что значения  $M = 8$  и  $T_{ar1} = 25$  мкс дают хорошие результаты на небольшой скорости, но затем сильно деградируют и на высоких скоростях дают вероятность, близкую к случаю использования гораздо менее надежных параметров  $M = 2$  и  $T_{ar1} = 12,5$  мкс. Как отмечалось ранее, это связано с тем, что при высоких ско-

ростях длительность раунда становится слишком большой и у метки остаётся мало шансов передать свои идентификаторы.

## 2.6 Заключение

В главе были представлены следующие результаты.

1. Рассмотрены основные особенности системы радиочастотной идентификации автомобилей. На основе анализа протокола радиочастотной идентификации EPC Gen2 выделены факторы, влияющие на её производительность.
2. Предложена имитационная модель, учитывающая различные сценарии идентификации автомобилей (идентификация только по значению EPCID или по паре EPCID и TID), параметры протокола EPC Gen2 (длительности символов в командах считывателя, способы кодирования ответов меток, выбор сессий и значений их флагов, число слотов в раунде), настройки считывателя (периодичность смены антенн и отключения питания), а также особенности распространения сигналов (многолучевое распространение, зависимость коэффициента отражения от материала дорожного покрытия, эффект Доплера).
3. С помощью имитационной модели были получены зависимости вероятности идентификации как отдельных номеров, так и автомобилей, от скорости движения, а также от выбранных настроек протокола EPC Gen2.

### Глава 3. Аналитическая модель системы радиочастотной идентификации автомобилей

В предыдущей главе с помощью имитационной модели были получены результаты, показывающие, что периодическая смена значений флага инвентаризации сессии и переключение питания между антеннами приводят к увеличению количества раундов, в которых метка принимает участие (см. рис. 2.13). При ошибке в передаче идентификатора считыватель не передает ни повторные команды АСК, ни команды NACK, а переходит сразу к следующему слоту или раунду. Из-за этого метка, не зная о произошедшей ошибке, инвертирует хранимое значение инвентаризационного флага текущей сессии. Например, если флаг опроса в текущем раунде был равен  $A$ , то метка инвертирует флаг  $A \rightarrow B$ . Метка принимает участие в следующем раунде в одном из двух случаев: если считыватель сменил значение флага опроса (в нашем примере с  $A$  на  $B$ ), или если считыватель продолжает опрос со значением флага  $A$ , но перед этим сбросил питание на достаточно продолжительное время. В последнем случае не имеет значения, просто ли считыватель выключил передатчик, или же переключился на другую антенну – лишь бы метка была выключена достаточно долго.

Имитационная модель, описанная ранее в предыдущей главе – очень сложная, поддерживает очень много параметров и требует значительного времени для получения статистически устойчивых результатов. Для более детального изучения зависимости числа раундов от переключений флагов опроса и сбросов питания при была разработана аналитическая модель, основанная на неоднородных марковских случайных процессах. С помощью этой модели можно описать произвольный сценарий сбросов питания и переключения значений флагов сессий, учесть законы движения меток и оценить вероятность идентификации.

Аналитическая модель строится на ряде допущений. Для их четкого определения в разделе 3.1 приводится описание модельного считывателя и меток, а после этого в разделе 3.2 – формальная постановка задачи. Далее, в разделе 3.3 приводится описание раунда инвентаризации, вводятся формулы для оценки длительности раунда. Кроме того, приводится распределение вероятностей случайной величины, описывающей число меток, успешно передающих RN16 без коллизий. В разделах 3.4 и 3.5 опишем модель как пару случайных процессов, переходные вероятности которых зависят от сценария, позволяющих найти оценки длительностей отдельных раундов (раздел 3.4), и рассчитать вероятность идентификации метки (раздел 3.5). После этого приводятся выражения для расчета переходных вероятностей процессов, зависящих от заданного сценария. В конце, в разделе 3.6, приведем численные результаты, а в разделе 3.7 сделаем выводы.

Результаты, представленные в главе, были опубликованы в сборнике трудов конференции IEEE RFID'2018, индексируемом в Scopus и WoS [192]. Кроме того, доклады по полученным результатам были представлены на международных конференциях ВСПУ 2019 [193], DCCN'2019 (Москва) и ICAAP&SP'2019 (Коттайам, Индия).

### 3.1 Ограничения и допущения

Для построения аналитической модели введём ряд необходимых допущений и ограничений на считыватель и метки, а также опишем приближенный способ расчета распределения длительностей раундов.

#### 3.1.1 Модельный считыватель

В определении **модельного считывателя** мы накладываем ограничения на область чтения, битовые ошибки в принятых ответах меток, длительности команд, схему раунда инвентаризации. Наиболее существенные ограничения касаются области чтения и битовых ошибок.

Будем считать, что метки движутся по прямой. **Областью чтения** модельного считывателя будем считать фиксированный интервал, в каждой точке которого метка имеет достаточно энергии для включения и приема команд от считывателя. Вне области чтения метки не работают, с единственным исключением: если метка, покидающая область чтения, уже принимает участие в раунде инвентаризации, считается, что она сможет передать свои ответы в любом из слотов.

Если при передаче ответа от метки произошла коллизия с ответом другой метки, считыватель не может принять ни один из ответов. Если же при передаче ответа коллизии не произошло, вероятность **битовой ошибки (BER)** в принятом ответе постоянна<sup>1</sup> и равна  $\beta$ . Возникновение ошибок в каждом бите будем полагать независимым, т.е. вероятность успешного приёма  $b$  бит от одной или нескольких меток есть  $b^{1-\beta}$ .

Модельный считыватель не изменяет значения параметров  $Q$ ,  $T_{ar}$ ,  $M$ ,  $TR_{ext}$ , а также калибровочных интервалов  $TR_{cal}$  и  $RT_{cal}$ . Число  $Q$  задает двоичный логарифм числа слотов, то есть число слотов  $N_s = 2^Q$ . Значение  $T_{ar}$  — это длительность символа data-0, равная 6,25, 12,5, 18,75 или 25,0 мкс. Значение  $M$  задает способ кодирования ответов меток, оно принимает значение 0 для кодирования FM0 и 1, 2 или 3 для кодов Миллера длиной 2, 4 или 8 символов соответственно.  $TR_{ext}$  — это флаг, задание которого означает, что метка должна использовать расширенную преамбулу перед своими ответами. Наконец, калибровочные интервалы  $TR_{cal}$  и  $RT_{cal}$  нужны для корректного разделения нулей и единиц, и для вычисления меткой скорости передачи ответов. Также считыватель всегда использует одну и ту же сессию для опроса меток; для определенности и простоты будем считать, что считыватель всегда использует сессию  $S_0$ .

**Длительности команд** будем считать фиксированными, игнорируя вариации, связанные с использованием кодировки P1E и полями, неизвестными заранее, — например,

<sup>1</sup>Это наиболее существенное ограничение, которое позволяет игнорировать точное расположение меток в области чтения, а также изменчивость канала во времени из-за эффекта Доплера.



случайным числом RN16 в команде АСК или значением контрольных сумм. Длительности команд будем обозначать как  $T\{\text{msg}\}$ , где msg – название команды. Интервалы<sup>2</sup> времени между концом команды и началом ответа метки ( $T_1$ ), между концом ответа метки и началом следующей команды ( $T_2$ ), а также между последовательными командами, если ответа от метки не было ( $T_1 + T_3$ ), считаются постоянными.

Считыватель не использует команды Select, NACK, QueryAdjust, а также повторные команды АСК. Для чтения банков памяти используется только команда Read, передаваемая после получения ответа на команду Req\_RN. Команда Read всегда направлена на один и тот же банк памяти и запрашивает чтение одного и того же числа слов.

В каждый момент модельный считыватель может выполнять одну из **трех операций**: производить опрос меток в раунде инвентаризации, сбрасывать питание или изменять флаг опроса. Операции выполняются последовательно, ненулевую длительность имеют только операции инвентаризации и сброса питания.

1. **Операция сброса питания** занимает время  $T_1$ , на это время считыватель отключает питание. Отключение и последующее включение питания происходят мгновенно. После включения считыватель готов сразу начать инвентаризацию с флагом Target, равным A.
2. **Операция смены флага инвентаризации** осуществляется мгновенно, в результате её выполнения инвертируется флаг сессии, по которому производится опрос меток (поле Target команды Query).
3. **Операция инвентаризации** заключается в выполнении раунда опроса меток, начинается с передачи команды Query и завершается вместе с последним тактом.

Из-за малого расстояния между считывателем и меткой длительность распространения сигнала не превышает 0,05 микросекунды, поэтому при оценке длительности раундов инвентаризации будем его игнорировать.

### 3.1.2 Модельные метки

Будем считать, что любой **модельной метке** всегда достаточно энергии в области чтения для включения и успешного приёма команд считывателя без ошибок. Также предполагаем, что метка при включении всегда сбрасывает значение хранимого флага в A, то есть длительность хранения флага всегда меньше, чем время отключения считывателя. Модельные метки не учитывают содержимое полей PC и слов XPC\_W1 и XPC\_W2 при формировании ответов на команды АСК. Для определенности будем предполагать, что в этих ответах метки передают PC, EPC и CRC. Длину EPCID считаем одинаковой для всех меток.

<sup>2</sup>Диапазоны интервалов  $T_1$ ,  $T_2$  и  $T_3$  определены в табл. 6.16 стандарта EPC Gen.2. В тексте используются стандартные обозначения этих интервалов; в тех случаях, когда смысл  $T_i$  может быть неверно истолкован, будет приведено дополнительное пояснение.

**Длительности ответов меток** будем считать известными и фиксированными. Будем обозначать их так же, как и длительности команд:  $T_{\text{msg}}$ , где  $\text{msg}$  — название ответа. Размеры ответов в битах будем обозначать как  $|\text{msg}|_b$ .

Также будем считать, что метки не используют паролей, считывателю не нужно передавать команду Access при чтении банков памяти. Команда Kill не рассматривается, и метка не может быть в «убитом» состоянии, когда при включении она сразу попадает в состояние KILLED и не отвечает на любые команды.

### 3.2 Постановка задачи

Будем считать, что метки движутся по одной прямой, а область чтения, в которой метки получают достаточно энергии для работы, является интервалом  $(0, L)$ . Пусть функция  $x(t)$  определяет закон движения меток так, что  $x(0) = 0$ ,  $\exists T_L : x(T_L) = L$  и  $\forall t \in (0, T_L) : x(t) \in (0, L)$ , а  $\forall t > T_L : x(t) > L$ . Метки появляются в известные моменты времени  $a_1, a_2, a_3, \dots$  в точке  $x = 0$ , их движение определяется функциями  $x_i(t) \equiv x(t - a_i)$ . Другими словами, метки движутся одинаково со сдвигом по времени, могут ускоряться, замедляться или моментально изменять свое местоположение, но после первого покидания области чтения, которое обязательно должно произойти, более никогда не возвращаются в неё.

Так как законы движения меток  $x_i(t)$  известны и детерминированы, число меток в области чтения в момент  $t$  можно определить как функцию  $N(t) = |\{i : a_i \leq t \leq a_i + T_L\}|$ . Для корректного определения случайных процессов потребуем, чтобы эта функция была ограничена сверху, то есть  $\exists \bar{N} = \max_{t \geq 0} N(t)$ . Для этого достаточно потребовать, чтобы существовал интервал  $\Delta t$ , такой, что  $\forall i : a_{i+1} - a_i \geq \Delta t$ .

Благодаря допущению о постоянстве BER при приеме ответов от меток, в дальнейшем будет гораздо проще построить модель, так как единственной существенной величиной, характеризующей множество меток, будет функция  $N(t)$ , а не отдельные значения координат  $x_i(t)$  или скоростей  $x'_i(t)$  меток. Хотя это допущение является очень сильным, оно может быть более или менее существенно в зависимости от взаимного положения меток и считывателя. Например, при движении меток по горизонтали на конвейерной ленте и горизонтальном размещении антенны считывателя над лентой уровень сигнала будет более равномерным, чем при размещении меток на машинах, а считывателя — под углом к дороге.

Последнее допущение относительно области чтения заключается в том, что метки покидают область чтения только на границе раундов. Другими словами, если в раунде приняло участие  $n$  меток, то любая метка остается в области чтения до конца и может ответить считывателю в любом из слотов.

Пусть  $r \in \mathbb{N}$  — произвольный номер раунда инвентаризации. Поведение считывателя характеризуется значением опрашиваемого флага  $X_r \in \{A, B\}$  и признаком сброса питания после раунда  $e_r \in \{0, 1\}$ .

**Определение 3.1.** *Спецификацией раунда* будем называть пару значений опрашиваемого флага и признака сброса питания  $(X, e)$ , которую будем сокращенно обозначать символом  $\alpha \stackrel{\text{def}}{=} X^e$ .

Например,  $\alpha_5 = B^1$  означает, что в пятом раунде считыватель ведет опрос по флагу инвентаризации  $B$  и после раунда сбрасывает питание, а  $\alpha_2 = A^0$  говорит о том, что во втором раунде опрос идет по флагу  $A$  и питание в конце не сбрасывается.

**Определение 3.2.** *Сценарием работы считывателя*  $\alpha = \alpha_1 \alpha_2 \dots \alpha_R$  будем называть последовательность спецификаций раундов конечной длины  $R$ .

Будем считать, что после  $R$ -го раунда считыватель начинает «проигрывать» сценарий с самого начала, с первого раунда. Например, переключение флага каждые два раунда со сбросом питания на каждом четвертом можно описать как любой их сценариев  $A^0 A^0 B^0 B^1$ ,  $A^0 A^0 B^0 B^1 A^0 A^0 B^0 B^1$  и так далее.

Под *идентификацией метки* мы будем понимать, как и в ранее во второй главе, либо успешную передачу EPCID метки, либо передачу TID.

**Задача 3.1.** Пусть известны законы движения меток  $x_i(t) \equiv x(t - t_i)$ , вероятность битовой ошибки в передаче ответов  $\beta$  и сценарий работы считывателя  $\alpha = \alpha_1 \alpha_2 \dots \alpha_R$ , а также размеры и длительности команд считывателя и ответов меток. Пусть для идентификации меток требуется только EPCID (или комбинация EPCID и TID). Требуется найти вероятность, с которой каждая метка будет успешно идентифицирована.

Для решения этой задачи опишем два случайных процесса: *фоновый процесс*, моделирующий число меток, участвующих в каждом раунде, и *основной процесс*, моделирующий проезд области чтения отдельной меткой и позволяющий оценить вероятность её идентификации. Для описания переходов основного процесса необходимо иметь оценку длительностей раундов, которую можно вычислить, исходя из оценки числа участвующих в каждом раунде меток. Последнюю оценку получим с помощью итерационного расчета стационарного распределения фонового процесса.

### 3.3 Моделирование раундов инвентаризации

Прежде, чем перейти к построению пространства состояний и операций над ним, рассмотрим подробнее процесс идентификации меток и найдем оценки следующих величин:

- вероятность  $P_n(m)$  передачи EPCID  $m$  метками из  $n$ ;
- вероятность  $p_{\text{id}}$  успешной идентификации отдельной метки;
- средняя ожидаемая длительность  $\tau = \tau(n)$  раунда с  $n$  метками;
- максимальная длительность раунда  $\tau_{\text{max}}$ .

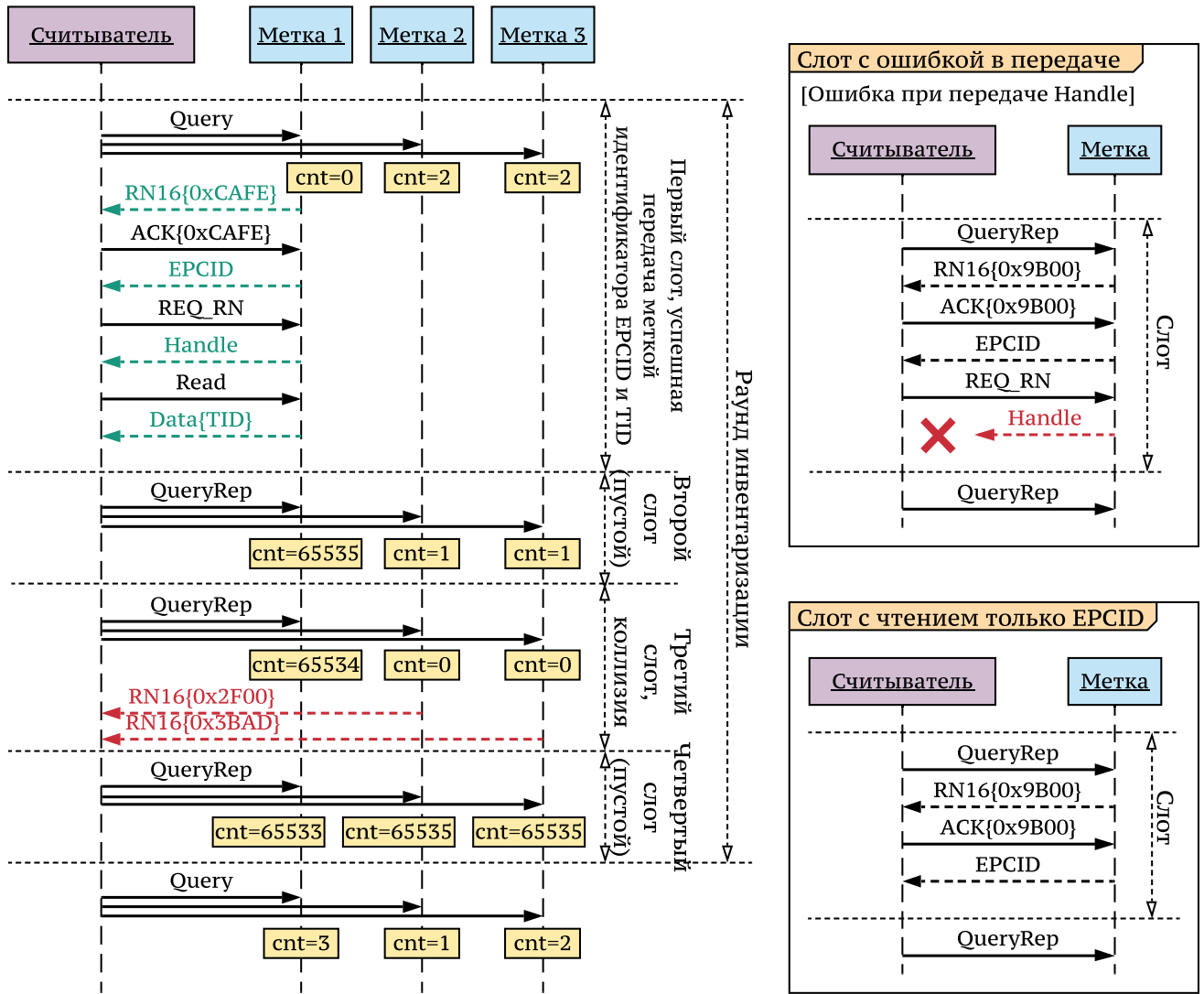


Рисунок 3.1 — Пример раунда инвентаризации с  $Q = 2$  и числом слотов  $N_s = 2^Q = 4$  в случае идентификации метки по EPCID и TID

Пример раунда инвентаризации показан на рис. 3.1. В этом примере предполагается, что параметр  $Q = 2$ , то есть число слотов  $N_s = 2^Q = 4$ , и в раунде участвуют три метки. Первая метка выбрала для передачи первый слот, а вторая и третья – третий. Идентификация метки происходит по комбинации EPCID и TID, поэтому в первом слоте считыватель после получения EPCID запрашивает случайное слово RN16 командой Req\_RN и использует его для контроля доставки в команде Read. В ответ на последнюю метка передает содержимое своего банка памяти TID. Если бы идентификация происходила только по EPCID, то первый слот был бы проще, обмен сообщениями выглядел бы как на правом нижнем фрагменте. Справа вверху показан пример слота, в котором произошла ошибка в передаче одного из ответов (Handle). Так как модельный считыватель не передает команды повторно, слот на этом завершается, метка остается неидентифицированной. Справа внизу показан пример слота, в котором считыватель не запрашивает TID.

Ошибка может произойти при передаче любого из ответов. Учитывая то, что BER постоянен и равен  $\beta$ , вероятность успешной передачи ответа msg можно найти как:

$$P_{\text{rx}}\{\text{msg}\} = (1 - \beta)^{|\text{msg}|}, \quad (3.1)$$

где  $|\text{msg}|$  – длина ответа  $\text{msg}$  в битах.

В дальнейшем потребуется оценивать число меток, участвующих в очередном раунде. Введём следующее определение.

**Определение 3.3.** Назовем метку *активной*, если она находится в области чтения, и ее значение флага сессии совпадает с передаваемым считывателем в поле Target команды Query. Значение этого поля будем называть *флагом опроса*.

*Замечание 3.1.* Следует подчеркнуть, что все модельные метки являются пассивными в техническом смысле слова, то есть они работают только в поле действия считывателя и не содержат автономного источника питания. Данное определение активности — исключительно логическое. Оно было выбрано, так как наиболее точно и лаконично описывает состояние метки.

Пусть в начале раунда в системе присутствует  $n$  активных меток. Обозначим  $\mu_0(n)$  — случайное число пустых слотов в этом раунде,  $\mu_1(n)$  — число слотов с ответом единственной метки без коллизий и  $\mu_2(n)$  — число слотов с коллизиями. Для дальнейших расчетов нам потребуются распределения вероятностей случайных величин  $\mu_0(n)$  и  $\mu_1(n)$ . Распределение  $\mu_0(n)$  определяется следующим образом:

$$\mathbb{P}\{\mu_0(n) = z\} = \frac{1}{N_s^n} C_{N_s}^z \left\{ \begin{matrix} n \\ N_s - z \end{matrix} \right\} (N_s - z)!, \quad (3.2)$$

где  $\left\{ \begin{matrix} n \\ N_s - z \end{matrix} \right\}$  — число Стирлинга 2-го рода, то есть число способов разбиения  $n$  меток по  $N_s - z$  непустым подмножествам.

Вероятность события  $\{\mu_1(n) = m\}$ , то есть того, что ровно  $m$  из  $n$  меток выбрали уникальные слоты и ответят считывателю без коллизий, будем обозначать как  $\bar{P}_n(m) = \mathbb{P}\{\mu_1(n) = m\}$ , и будем вычислять с помощью формулы, приведенной в работе [67]:

$$\bar{P}_n(m) = \mathbb{P}\{\mu_1(n) = m\} = \frac{N_s! n!}{m! N_s^n} \sum_{z=0}^{n-m} \frac{(-1)^z (N_s - m - z)^{(n-m-z)}}{(n - m - z)! (N_s - m - z)!}. \quad (3.3)$$

Сделаем два замечания относительно формул (3.2) и (3.3). Во-первых, в (3.3) предполагается, что число активных меток  $n$  не превосходит числа слотов  $N_s$ . Во-вторых, на практике обе формулы неудобно применять из-за очень больших чисел в числителях и знаменателях уже при  $N_s > 8$ , то есть  $Q = \log_2 N_s > 3$ . Частично эту проблему удастся решить группировкой множителей, но не во всех случаях, поэтому при численных расчетах для вычисления обеих вероятностей будем пользоваться методом Монте-Карло.

**Утверждение 3.1.** Если в раунде инвентаризации участвует  $n > 0$  меток,  $P_{\text{rx}}\{\text{RN16}\}$  – вероятность успешной передачи ответа RN16, а  $\bar{P}_n(k)$  – вероятность того, что ровно  $k \leq n$  меток выбрали такие слоты, которые не выбрали другие метки, то вероятность

того, что ровно  $m \leq n$  меток изменят после раунда хранимое значение флага определяется следующим выражением:

$$P_n(m) = \sum_{i=m}^n \bar{P}_n(i) C_i^m x (P_{rx}\{RN16\})^m (1 - P_{rx}\{RN16\})^{i-m}, \quad m \leq n. \quad (3.4)$$

*Доказательство.* Метка передает идентификатор, если ей пришла команда АСК со случайным словом, совпадающим с переданным ранее меткой в ответе RN16. Согласно сделанным предположениям, это происходит, когда метка передает сообщение RN16 единственной в слоте, и сообщение доставляется без ошибок<sup>3</sup>. Тогда событие  $X$ :

$X =$  «ровно  $m$  из  $n$  меток передали идентификаторы»

можно представить в виде суммы непересекающихся событий  $X = \sum_{i=m}^n Y_i$ , где:

$Y_i =$  «ровно  $i$  из  $n$  меток передали RN16 без коллизий,  
и ровно  $m$  из  $i$  из них было принято без ошибок».

По формуле полной вероятности, учитывая независимость выбора слотов метками и возникновения ошибок в передаче их ответов, а также независимость возникновения ошибок в каждом переданном метками бите, получаем цепочку равенств, доказывающих утверждение:

$$\begin{aligned} P_n(m) &= \mathbb{P}\{X\} = \sum_{i=m}^n \mathbb{P}\{Y_i\} = \\ &= \sum_{i=m}^n (\mathbb{P}\{\text{ровно } i \text{ из } n \text{ меток выбрали свободные слоты}\} \times \\ &\quad \mathbb{P}\{\text{ровно } m \text{ из } i \text{ меток успешно передали RN16}\}) = \\ &= \sum_{i=m}^n \bar{P}_n(i) \times (C_i^m (P_{rx}\{RN16\})^m (1 - P_{rx}\{RN16\})^{i-m}) \end{aligned}$$

■

Для того, чтобы удостовериться, что  $P_n(m)$  корректно описывает вероятность, докажем, что  $\sum_{m=0}^n P_n(m) = \sum_{i=0}^n \bar{P}_n(i) = 1$  (справедливость последнего равенства утверждается в

<sup>3</sup>Если бы использовалась более точная модель интерференции, получение ответа было бы возможно также, например, если сигнал от другой метки был в том же слоте, но очень ослабленный.

работе [67]). Действительно,

$$\begin{aligned}
\sum_{m=0}^n P_n(m) &= \sum_{m=0}^n \sum_{i=m}^n \bar{P}_n(i) C_i^m P_{rx}\{\text{RN16}\}^m (1 - P_{rx}\{\text{RN16}\})^{i-m} = \\
&= \sum_{i=0}^n \sum_{m=0}^i \bar{P}_n(i) C_i^m P_{rx}\{\text{RN16}\}^m (1 - P_{rx}\{\text{RN16}\})^{i-m} = \\
&= \sum_{i=0}^n \bar{P}_n(i) \sum_{m=0}^i C_i^m P_{rx}\{\text{RN16}\}^m (1 - P_{rx}\{\text{RN16}\})^{i-m} = \\
&= \sum_{i=0}^n \bar{P}_n(i) (P_{rx}\{\text{RN16}\} + (1 - P_{rx}\{\text{RN16}\}))^i = \\
&= \sum_{i=0}^n \bar{P}_n(i).
\end{aligned}$$

Вероятность успешной идентификации зависит от того, что используется в качестве идентификатора: EPCID или комбинация TID и EPCID. Условную вероятность успешной идентификации метки при условии отсутствия коллизии и ошибки в передаче RN16 будем обозначать как  $p_{id}$ :

$$p_{id} = \begin{cases} P_{rx}\{\text{EPCID}\}, & \text{только EPCID} \\ P_{rx}\{\text{EPCID}\}P_{rx}\{\text{Handle}\}P_{rx}\{\text{TID}\}, & \text{EPCID и TID} \end{cases} \quad (3.5)$$

Для вычисления оценки средней длительности раунда, воспользуемся приближенной схемой расчёта в допущении о независимости определения типов слотов. Будем считать, что каждый слот может принадлежать одному из трех типов: пустой слот (без ответа метки), слот с коллизией (ошибка при передаче RN16) и слот с ответом без коллизии. Обозначим через  $t_0$  длительность пустого слота,  $t_1^{(epc)}$  и  $t_1^{(tid)}$  — средние длительности слотов с ответом без коллизий при идентификации по EPCID или по TID соответственно, а через  $t_2$  — длительность слота с коллизией. Эти длительности могут быть вычислены следующим образом:

$$\begin{aligned}
t_0 &= T_{QRep} + T_1 + T_3 \\
t_1^{(epc)} &= T_{QRep} + (T_1 + T_2) + T_{RN16} + P_{rx}\{\text{RN16}\} \times \\
&\quad (T_{ACK} + (T_1 + T_2) + T_{EPCID}) \\
t_1^{(tid)} &= T_{QRep} + (T_1 + T_2) + T_{RN16} + P_{rx}\{\text{RN16}\} \times \\
&\quad (T_{ACK} + (T_1 + T_2) + T_{EPCID} + P_{rx}\{\text{EPCID}\} \times \\
&\quad (T_{Req\_RN} + (T_1 + T_2) + T_{Handle} + P_{rx}\{\text{Handle}\} \times \\
&\quad (T_{Read} + (T_1 + T_2) + T_{TID}))) \\
t_1 &= \begin{cases} t_1^{(epc)}, & \text{только EPCID} \\ t_1^{(tid)}, & \text{EPCID и TID} \end{cases} \\
t_2 &= T_{QRep} + (T_1 + T_2) + T_{RN16}
\end{aligned} \quad (3.6)$$

Согласно допущению, вероятности слотов определяются независимо. Обозначим  $p_0(n)$ ,  $p_1(n)$  и  $p_2(n)$  вероятности того, что произвольный слот пуст, содержит ответ ровно одной метки или коллизии, соответственно. Эти вероятности можно вычислить, используя (3.2) и (3.3):

$$\begin{aligned} p_0(n) &= \frac{1}{N_s} \mathbb{M}\mu_0 = \frac{1}{N_s} \sum_{i=0}^{N_s} i \mathbb{P}\{\mu_0 = i\} \\ p_1(n) &= \frac{1}{N_s} \mathbb{M}\mu_1 = \frac{1}{N_s} \sum_{i=0}^{N_s} i \mathbb{P}\{\mu_1 = i\} \\ p_2(n) &= 1 - p_0(n) - p_1(n) \end{aligned} \quad (3.7)$$

С помощью соотношений (3.6) и (3.7) длительность раунда  $\tau(n)$  можно определить как сумму математического ожидания длительности слота и, если после раунда сбрасывается питание, длительности этого сброса:

$$\tau_r(n) = N_s \sum_{i=0}^2 p_i(n) t_i + (T_{\text{Query}} - T_{\text{QRep}}) + e_r T_{\downarrow}, \quad (3.8)$$

где  $n$  — число меток, участвующих в раунде,  $e_r$  — индикатор сброса питания после опроса, а  $T_{\downarrow}$  — длительность сброса. Слагаемое  $(T_{\text{Query}} - T_{\text{QRep}})$  необходимо, так как при расчете длин слотов  $t_i$  предполагалось, что каждый слот начинается с команды QRep, хотя первый слот начинается с более длинной команды Query.

Хотя в действительности распределения величин  $\mu_0$ ,  $\mu_1$  и  $\mu_2$  не являются независимыми, приведенный способ расчета очень простой и позволяет получить достаточно точную оценку длительности раундов. На рис. 3.2 показан расчет с помощью описанной аналитической модели и метода Монте-Карло при различных значениях BER и числе участвующих в опросе меток от 0 до 20 (остальные параметры были фиксированы, число слотов  $N_s = 2^Q = 16$ ). Можно видеть, что допущение о независимости определения типа слота оказывает минимальное влияние на точность оценки.

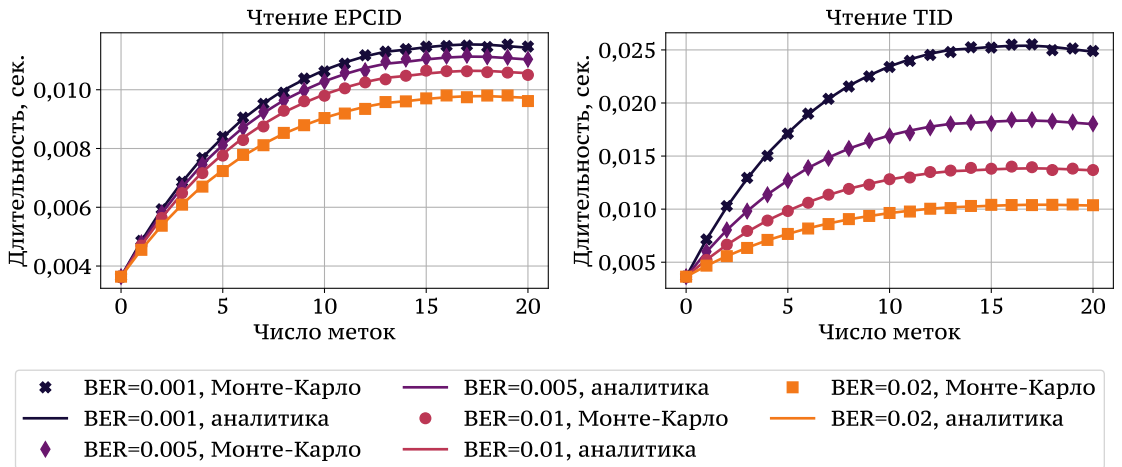


Рисунок 3.2 — Сравнение результатов расчета средней длительности раундов с помощью аналитической модели и метода Монте-Карло



Если известно распределение вероятностей числа активных меток  $\boldsymbol{\pi} \in \mathbb{R}^{\bar{N}+1}$ , где  $\pi_n$  — вероятность того, что в системе ровно  $n$  активных меток,  $n \in [0, \bar{N}]$ , то среднюю длину раунда можно опередлить как математическое ожидание  $\tau_r = \mathbb{M}\tau_r(n)$ :

$$\tau_r = \sum_{n=0}^{\bar{N}} \pi_n \tau_r(n). \quad (3.9)$$

Пользуясь выражениями для длительностей слотов  $t_1, t_2$  и  $t_3$ , можно найти оценку максимальной длительности раунда  $\tau_{max}$ :

$$\tau_{max} = \begin{cases} \bar{N}t_1 + (N_s - \bar{N})t_0, & \bar{N} \leq N_s \\ (N_s - 1)t_1 + t_2, & \bar{N} > N_s. \end{cases} \quad (3.10)$$

Справедливость этого выражения следует из того, что  $t_0 < t_2 < t_1$  (значение  $T_3$  заведомо не превышает  $T_{RN16}$ ): если слотов больше, чем меток, то максимум достигается, когда все метки успешно передают свои идентификаторы. Если же слотов меньше, то в самом длинном раунде все слоты кроме одного будут содержать успешные передачи, а в одном слоте произойдет коллизия оставшихся  $\bar{N} - (N_s - 1)$  меток.

### 3.4 Вычисление оценки длительностей раундов

В этом разделе опишем случайный процесс, моделирующий изменение числа активных меток. С помощью него найдем среднее число меток, участвующих в каждом раунде, и вычислим оценку средних длительностей раундов. Для этого сначала введем понятие размеченных сценариев и определим, как их строить и транслировать в последовательность операций, которые считыватель может совершать над метками. Затем рассмотрим, как этим операциям поставить в соответствие матрицы переходов случайного процесса, найдем его стационарное распределение и вычислим среднее число меток в каждом раунде.

#### 3.4.1 Размеченные сценарии и элементарные операции

Пусть в некоторый момент времени в системе находится  $N$  меток,  $0 \leq N \leq \bar{N}$ , и пусть  $n$  из них активны, то есть готовы принять участие в раунде, в котором опрос просходит по флагу  $X \in \{A, B\}$ ,  $0 \leq n \leq N$ .

Число  $n$  может измениться по трем причинам (см. рис. 3.3). Во-первых, если флаг опроса  $X$  после очередного раунда инвентаризации не изменяется, и  $n' \leq n$  меток в этом раунде передали свои EPCID, то эти метки инвертировали свои флаги и в следующем раунде не

будут принимать участие, то есть активными будут  $n - n'$  меток. Если же после раунда считыватель инвертирует флаг опроса  $X$ , то число активных меток будет  $N - (n - n')$ . Во-вторых, само число  $N$  может измениться, если какая-либо метка покинула систему, или, наоборот, вошла в нее. Это изменение может затронуть число активных меток  $n$ , если систему покинула активная метка (в этом случае  $n$  уменьшается на единицу), или если метка поступила в систему, и текущий флаг опроса равен  $A$  (значение  $n$  увеличивается на единицу). Наконец, в-третьих, считыватель может сбросить питание, после чего все метки будут хранить значение флага, равное  $A$ . Если считыватель продолжит опрос с флагом  $X = A$ , то активными будут все  $n = N$  меток, а если с флагом  $B$ , то активных меток точно не будет,  $n = 0$ .

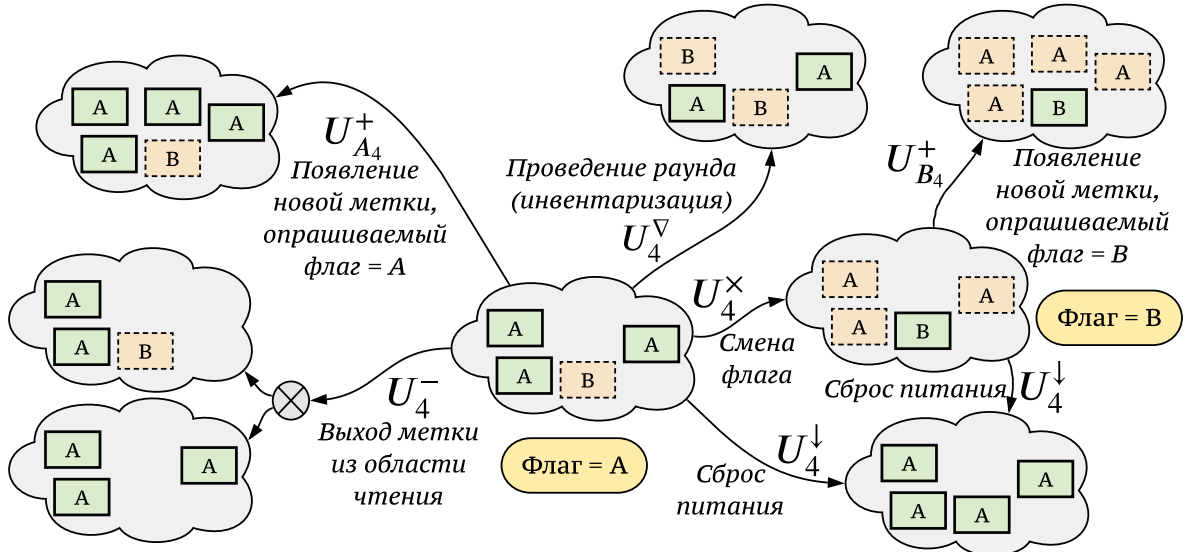


Рисунок 3.3 — Операции над системой

Таким образом, изменение числа активных меток происходит в результате пяти действий, которые в дальнейшем будем называть *элементарными операциями*: проведения раунда инвентаризации ( $U_N^{\nabla}$ ), изменения флага опроса ( $U_N^{\times}$ ), сброса питания ( $U_N^{\downarrow}$ ), добавления метки в систему ( $U_N^+$ ) или выхода метки ( $U_N^-$ ). Как будет показано далее, с помощью композиции элементарных операций можно описать, как изменяется распределение числа активных меток после каждого раунда. Однако первый вопрос — как по известному сценарию работы считывателя определить, какие элементарные операции нужно выполнять.

В начале главы было введено понятие сценария (см. определение 3.2) как последовательности  $\alpha = \alpha_1 \alpha_2 \dots \alpha_R$ , где  $\alpha_r = (X_r, e_r) \equiv X_r^{e_r}$  — спецификация раунда. По заданному сценарию можно описать операцию, которую считыватель должен провести над метками. Например, если  $\alpha = \alpha_1 \alpha_2 \alpha_3 \alpha_4 = A^0 B^1 A^0 A^1$ , то в первом раунде нужно провести опрос ( $U_N^{\nabla}$ ) и инвертировать флаг опроса с  $A$  на  $B$  ( $U_N^{\times}$ ), во втором раунде — провести опрос и сбросить питание ( $U_N^{\downarrow}$ ), в третьем раунде просто провести опрос, а в четвертом — провести опрос и сбросить питание.

В сценарии содержится только информация о работе считывателя, однако нет данных о том, когда метки появляются и выходят из системы. Обозначим число меток в области чтения в  $r$ -м раунде как  $N_r$ , а число меток, покидающих и поступающих в область чтения за время  $r$ -го раунда как, соответственно,  $\Delta_r^-$  и  $\Delta_r^+$ . Предположим, что удалось установить,

сколько меток было в области чтения в начале первого раунда  $N_1$ , а также все значения  $\{\Delta_r^-\}_{r=1}^R$  и  $\{\Delta_r^+\}_{r=1}^R$ . Отметим, что для раунда  $r > 1$  число меток можно вычислить как  $N_r = N_{r-1} - \Delta_{r-1}^- + \Delta_{r-1}^+$ . Введем следующее определение, объединяющее сценарий  $\alpha$  и оценки  $\{N_r\}$ ,  $\{\Delta_r^-\}$  и  $\{\Delta_r^+\}$ .

**Определение 3.4.** Будем называть *размеченным сценарием* последовательность символов  $\tilde{\alpha} = \tilde{\alpha}_1 \tilde{\alpha}_2 \dots \tilde{\alpha}_R$ , каждый из которых имеет вид пятерки  $\alpha_r = (X_r, e_r; N_r, \Delta_r^-, \Delta_r^+)$ . Символы  $\alpha_r$  будем называть *размеченными спецификациями раундов*. В дальнейшем будем использовать обозначение  $[\begin{smallmatrix} N_r & X & e_r \\ \Delta_r^- & & \Delta_r^+ \end{smallmatrix}]$ , причем для сокращения записи иногда не будем указывать  $\Delta_r^-$ ,  $\Delta_r^+$  и  $e_r$ , если они равны нулю.

Возвращаясь к предыдущему примеру, если сценарий  $\alpha = A^0 B^1 A^0 A^1$ ,  $N_1 = 5$ ,  $\{\Delta_r^+\} = \{0, 1, 0, 1\}$  и  $\{\Delta_r^-\} = \{1, 0, 0, 1\}$ , то  $\{N_r\} = \{5, 4, 5, 5\}$  и

$$\begin{aligned} \tilde{\alpha} &= ((A, 0; 5, 1, 0), (B, 1; 4, 0, 1), (A, 0; 5, 0, 0), (A, 1; 5, 1, 1)) = \\ &= [\begin{smallmatrix} 5 & A_0^0 \\ 0 & B_1^1 \end{smallmatrix}] [\begin{smallmatrix} 5 & A_0^0 \\ 0 & A_1^1 \end{smallmatrix}] = [\begin{smallmatrix} 5 & A \\ 1 & B_1^1 \end{smallmatrix}] [\begin{smallmatrix} 5 & A \\ 1 & A_1^1 \end{smallmatrix}]. \end{aligned}$$

Для такого размеченного сценария уже можно полностью описать все действия над системой: в первом раунде после опроса ( $U_5^\nabla$ ) нужно инвертировать флаг ( $U_5^\times$ ) и удалить метку ( $U_{B,5}^+$ ), во втором раунде — провести опрос ( $U_4^\nabla$ ), сбросить питание ( $U_4^\downarrow$ ) и добавить одну метку ( $U_{A,4}^+$ ), в третьем — только провести опрос ( $U_5^\nabla$ ), а в четвертом — провести опрос ( $U_5^\nabla$ ), сбросить питание ( $U_3^\downarrow$ ), удалить одну метку ( $U_5^-$ ) и затем добавить еще одну метку ( $U_{A,4}^+$ ). Отметим, что пример последнего раунда показывает, почему недостаточно знать число меток в каждом раунде, а важно также знать, сколько меток поступает и покидает область чтения.

Для построения размеченного сценария нужно найти число меток перед первым раундом, а также определить, сколько меток поступает и покидает систему в каждом раунде. Пусть модельное время начинается в момент  $T_0$ . Обозначим последовательность моментов начала раундов опросов как  $\{t_r\}_{r=1}^\infty$ . Тогда  $\Delta_r^+ = |\{a_i : t_r \leq a_i < t_{r+1}\}|$  и  $\Delta_r^- = |\{a_i : t_r \leq a_i + T_L < t_{r+1}\}|$ , а  $N_1 = N(T_0)$ . Таким образом, для нахождения  $\Delta_r^-$  и  $\Delta_r^+$  нужно знать, когда начинается каждый раунд, а для этого нужно вычислить оценки длительностей каждого раунда  $\{\tau_r\}$ . Длительности раундов опроса  $\{\tau_r\}$  вычисляются с помощью выражений (3.8) и (3.9), но для их использования нужно знать распределение вероятностей числа меток  $n$ , принимающих участие в каждом раунде.

Прежде, чем перейти к решению вопроса о нахождении оценок длительностей раундов, отметим, что значение  $N_r$  можно вычислить, как  $N_r = N(T_0 + \tau_1 + \tau_2 + \dots + \tau_{r-1})$ . Величина  $T_0$  может быть произвольной, но желательно устанавливать ее достаточно большой, чтобы избежать погрешности из-за краевых эффектов.

### 3.4.2 Матрицы элементарных операций

Как отмечалось ранее, число активных меток является случайной величиной, так как в общем случае вероятность того, что в течение раунда инвентаризации ровно  $n - n'$  меток передадут свои идентификаторы, меньше единицы. Рассмотрим случайный процесс  $\{\eta_r \in [0, \bar{N}]\}_{r=1}^\infty$ , моделирующий число активных меток в системе. Время в этом процессе дискретно и увеличивается на единицу после выполнения операции. В дальнейшем, в качестве такой операции будет выступать выполнение раунда и всех действий, определенных спецификацией. Однако, в этом разделе, прежде чем ввести определение операции, моделирующей весь раунд, для удобства будем считать, что  $r$  меняется после каждой *элементарной* операции.

Пусть  $U$  — одна из элементарных операций, число активных меток перед ее выполнения равно  $\eta_r = n$ , а после —  $\eta_{r+1} = n'$ ,  $n, n' \in [0, \bar{N}]$ . В силу сделанных ранее допущений, в частности — о постоянстве BER во всей области чтения, каждую элементарную операцию будем задавать распределением вероятностей  $\mathbb{P}\{\eta_{r+1} = n' | \eta_r = n\}$ . Переходные вероятности операции  $U$  можно задать в виде матрицы порядка  $\bar{N} + 1$ . Для удобства будем считать, что нумерация строк и столбцов в матрицах  $U$  и элементов вектора  $\boldsymbol{\pi}$  начинается с нуля, чтобы номера строк и столбцов в точности соответствовали числу активных меток; тогда  $\{U\}_{ij} = \mathbb{P}\{\eta_{r+1} = j | \eta_r = i\}$ . Рассмотрим переходные матрицы каждой из элементарных операций (см. рис. 3.4).

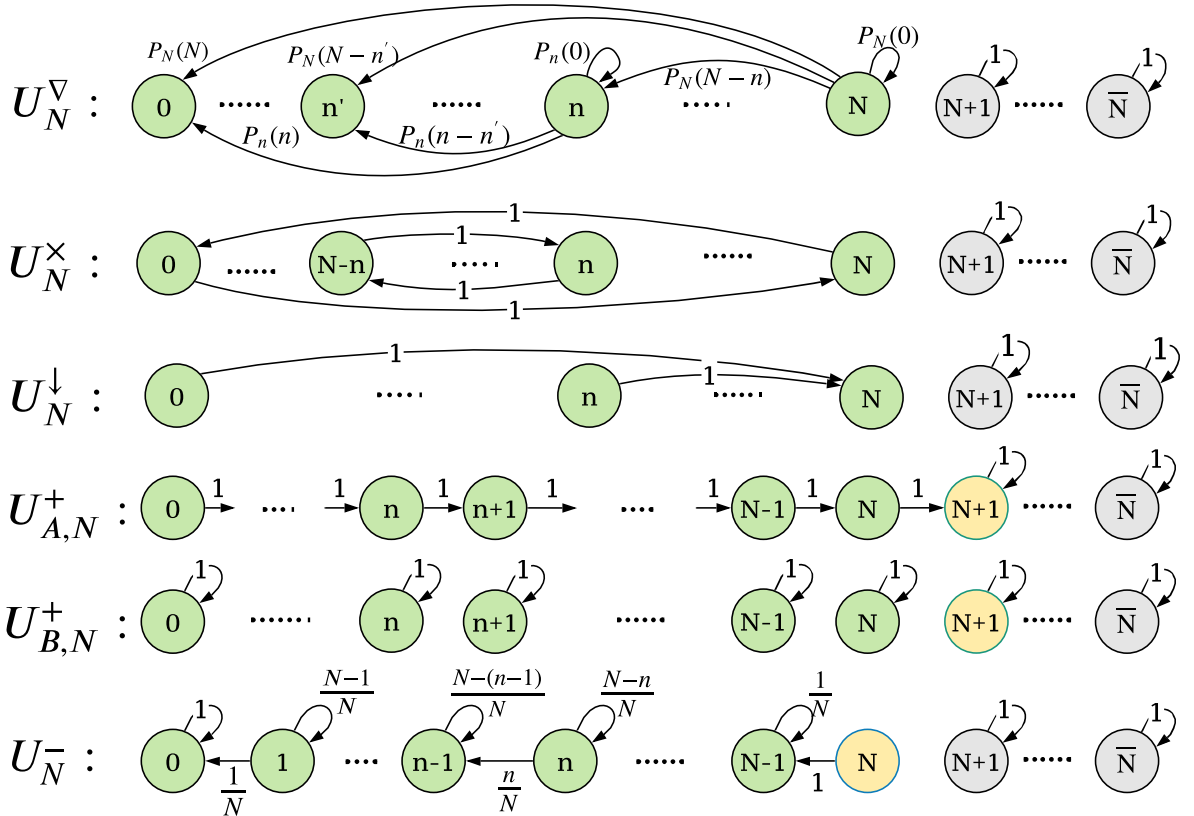


Рисунок 3.4 — Изменение числа активных меток при выполнении элементарных операций

Применение операции опроса меток приводит к тому, что часть меток передают (возможно, успешно) свои EPCID и инвертируют флаг. Непосредственно после этой операции флаг не меняется, питание не отключается, а метки не покидают и не поступают в область чтения. Вероятность того, что после опроса ровно  $n'$  из  $n$  меток будут активны, равна  $P_n(n - n')$  (см. выражение (3.4)), а матрица  $U_N^\nabla$  задается так:

$$\{U_N^\nabla\}_{ij} = \begin{cases} P_i(i - j), & 0 \leq j \leq i \leq N \\ 1, & N < i = j \leq \bar{N} \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.11)$$

В результате инвертирования флага опроса неактивные метки становятся активными и наоборот, и число активных меток становится равным  $N - n$ :

$$\{U_N^\times\}_{ij} = \begin{cases} 1, & 0 \leq i \leq N, j = N - i \\ 1, & N < i = j \leq \bar{N} \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.12)$$

Сброс питания приводит к тому, что у всех меток в области чтения хранимое значение флага становится равным  $A$ . Так как после этого опрашивать метки по флагу  $B$  не имеет практического смысла, будем считать, что флаг опроса также становится равным  $A$ , то есть все метки — активны:

$$\{U_N^\downarrow\}_{ij} = \begin{cases} 1, & 0 \leq i \leq N, j = N \\ 1, & N < i = j \leq \bar{N} \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.13)$$

Выполнение операции добавления метки ведет к тому, что в системе становится на одну метку больше (то есть увеличивается  $N$ ). Изменение числа активных меток зависит от текущего значения флага опроса: если флаг опроса  $X = A$ , то число активных меток увеличивается, а если  $X = B$ , не изменяется:

$$\{U_{A,N}^+\}_{ij} = \begin{cases} 1, & 0 \leq i < \bar{N}, j = i + 1 \\ 1, & N + 1 \leq i = j \leq \bar{N} \\ 0, & \text{в остальных случаях} \end{cases} \quad (3.14)$$

$$U_{B,N}^+ = I_{\bar{N}+1},$$

где  $I_{\bar{N}+1}$  — единичная матрица порядка  $\bar{N} + 1$ .

Наконец, при выходе метки из области чтения число меток  $N$  уменьшается. Изменение числа активных меток зависит от того, была ли покинувшая систему метка активной. Здесь необходимо сделать одно замечание. В идеале, нужно учитывать, какая метка находится ближе к области выхода, и смотреть, активна ли она. Однако, это привело бы к серьезному увеличению пространства состояний: вместо одного числа активных меток нужно было бы

хранить в состоянии системы положения меток и значения их флагов. Поэтому при определении этой элементарной операции сделаем допущение о том, что покинуть систему может равновероятно любая метка. В этом случае вероятность изменения числа активных меток пропорциональна их числу в системе:

$$\{U_N^-\}_{ij} = \begin{cases} \frac{i}{N}, & 0 < i \leq N, j = i - 1 \\ \frac{N-i}{N}, & 0 \leq i < N, i = j \\ 1, & N \leq i = j \leq \bar{N} \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.15)$$

Прежде, чем перейти к моделированию раундов инвентаризации, сделаем замечание относительно элементов матриц. Можно видеть, что во всех матрицах справа внизу содержится единичный блок. Вообще говоря, строки при  $i > N$  не имеют физического смысла: любая операция вида  $U_N^\bullet$  применяется к системе, в которой находится  $N$  меток. Было бы логично сделать так, чтобы все матрицы имели порядок  $N + 1$  ( $N + 2$  для  $U_N^+$ ), то есть чтобы каждый элемент соответствовал возможному изменению числа активных меток, учитывая, что в системе перед выполнением операции находится ровно  $N$  меток. Однако, далее потребуется рассматривать композиции элементарных операций, для чего их матрицы будут перемножаться. Можно видеть, однако, что вероятность попадания в состояние  $n > N$  в результате выполнения любой из операций (за исключением  $U_N^+$ ) равна нулю, и любое из состояний  $n > N$  является поглощающим.

### 3.4.3 Построение операций по размеченному сценарию

Обозначим распределение вероятностей случайной величины  $\eta_r$  как  $\boldsymbol{\pi}^{(r)} \in \mathbb{R}^{\bar{N}+1}$ . Здесь  $\pi_n^{(r)} = \mathbb{P}\{\eta_r = n\}$  при  $n \in [0, N_r]$ , и  $\pi_n^{(r)} \equiv 0$  при  $n > N_r$ . Если перед выполнением элементарной операции  $U_i$  распределение вероятностей  $\boldsymbol{\eta}$  было  $\boldsymbol{\pi}$ , то распределение после ее выполнения будет  $\boldsymbol{\pi}U_i$ , после выполнения  $U_{i+1}$  распределение будет  $\boldsymbol{\pi}U_iU_{i+1}$  и так далее. Действительно, учитывая, что вероятность изменения  $\eta_r$  вследствие выполнения операции определяется только текущим значением  $\eta_r$  и типом элементарной операции, случайный процесс  $\{\eta_r\}$  будет неоднородным марковским процессом с дискретным временем, матрицы переходов которого есть матрицы операций.

Представим весь процесс изменения состояния системы в виде последовательности операций  $D_r$ , по одной операции на раунд опроса. Каждую операцию  $D_r$  можно представить в виде композиции элементарных операций (см. пример 3.5). Отсюда и далее будем полагать, что время в случайном процессе  $\eta_r$  (число активных меток) меняется на единицу в начале каждого раунда.

Рассмотрим расширенный размеченный сценарий  $\tilde{\boldsymbol{\alpha}} = \tilde{\alpha}_1\tilde{\alpha}_2\dots\tilde{\alpha}_R$ . Пусть  $\tilde{\alpha}_r = (X, N, e, \Delta^-, \Delta^+)$  и  $\tilde{\alpha}_{r+1} = (Y, \bullet, \bullet, \bullet, \bullet)$  (здесь  $\bullet$  — произвольное значение). Все, что проис-

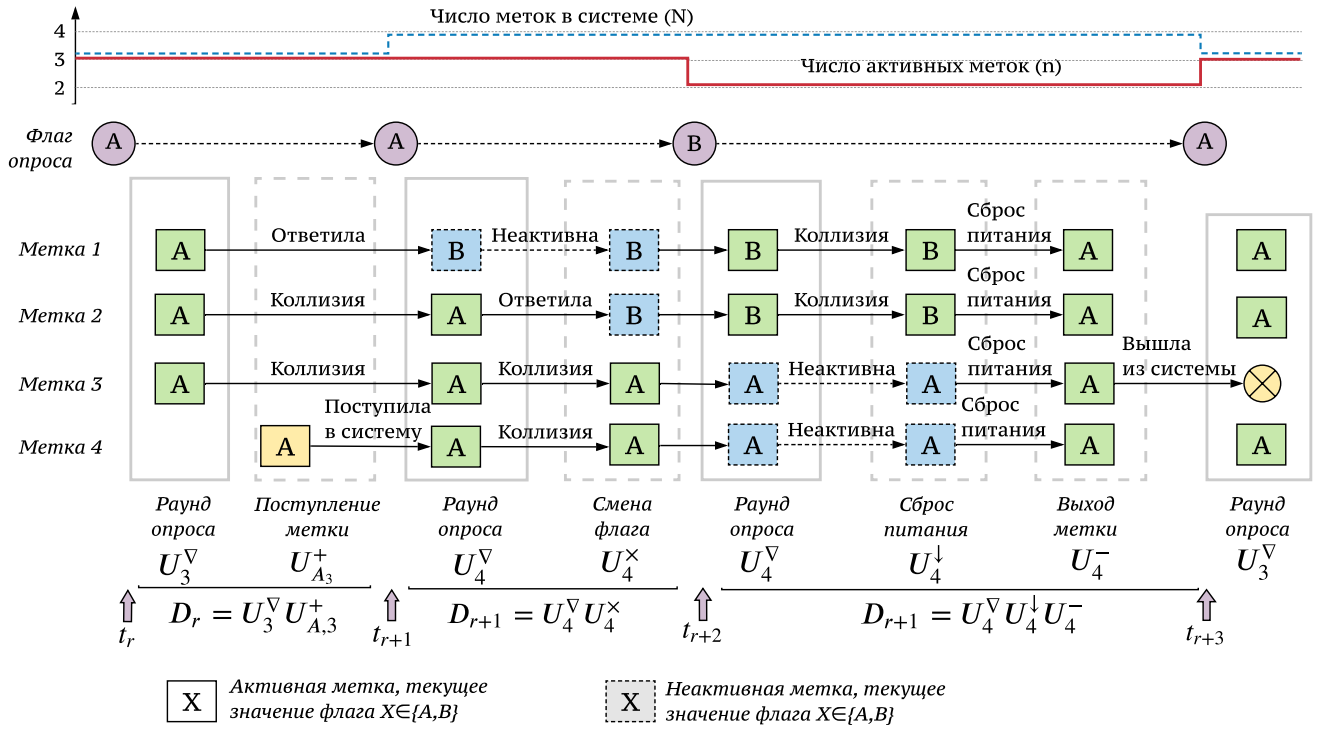
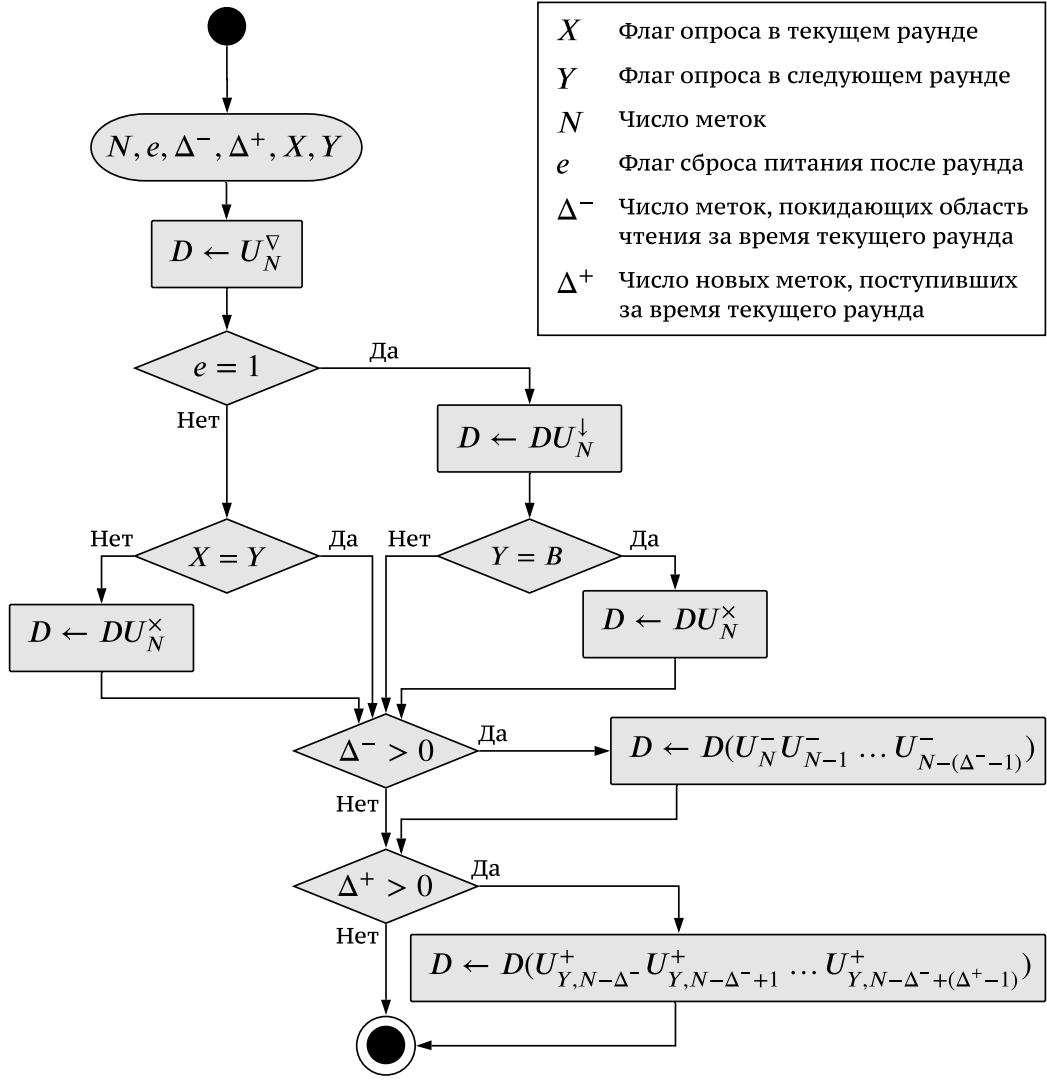


Рисунок 3.5 — Пример представления раундов в виде операций, и декомпозиция этих операций до элементарных операций

ходит с системой в  $r$ -м раунде, полностью описывается значениями текущего флага опроса  $X$ , числом меток  $N$ , количеством покидающих и поступающих в систему меток  $\Delta^-$  и  $\Delta^+$ , признаком сброса питания после опроса  $e$  и значением флага в следующем раунде  $Y$ .

Алгоритм построения матрицы  $D = D_r$  приведен на рис. 3.6. Матрица  $D$  представляется в виде произведения матриц элементарных операций  $U_i$ , причем первым множителем в этом произведении всегда является  $U_N^\nabla$ , то есть матрица операции инвентаризации меток. Далее, если считыватель выключается, то нужно умножить  $D$  на  $U_N^\downarrow$  и, если в следующем раунде считыватель опрашивает метки по флагу  $Y = B$ , умножить на матрицу операции смены флага  $U_N^\times$ . Если считыватель не выключается, то  $D$  нужно умножить на матрицу  $U_N^\times$  в том случае, если флаг опроса в следующем раунде инвертируется, то есть  $X \neq Y$ . Если за время опроса и отключения (если  $e = 1$ ) считывателя одна или более меток покинули область чтения, то есть  $\Delta^- > 0$ , то умножаем  $D$  на матрицы  $U_N^-, U_{N-1}^-, \dots, U_{N-(\Delta^- - 1)}^-$  (так как после каждой операции вида  $U_N^-$  число меток в области чтения становится меньше на единицу, уменьшается и нижний индекс). Если за время опроса в систему поступили метки, то есть  $\Delta^+ > 0$ , то аналогично умножаем матрицу  $D$  на матрицы операций добавления меток  $U_{Y, N'}^+, U_{Y, N'+1}^+, \dots, U_{Y, N+(\Delta^+ - 1)}^+$ , где  $N' = N - \Delta^-$ .

Отметим, что порядок умножения матриц элементарных операций важен. К примеру, пусть в раунде  $r + 1$  считыватель ведет опрос по флагу  $A$ , и с ненулевой вероятностью все метки в конце раунда  $r$  хранят флаг  $B$ , то есть число активных меток равно нулю ( $\eta_r = 0$ ). Пусть также после раунда  $r$  одна метка поступает в систему, а другая — покидает ее. Новая метка всегда хранит флаг сессии со значением  $A$ , то есть в данном случае будет активной в раунде  $r + 1$ . Очевидно, что после этого в начале  $(r + 1)$ -го раунда в системе обязатель-

Рисунок 3.6 — Алгоритм построения матрицы перехода  $D$ 

но будет хотя бы одна активная метка. Но если бы операция добавления учитывалась до операции удаления метки, то с вероятностью  $1/(N+1)$  число активных меток уменьшилось бы до нуля. По тем же причинам, смена флага должна учитываться до добавления меток, но после сброса питания.

Вернемся к примеру расширенного сценария  $\tilde{\alpha} = \begin{bmatrix} 5A_0^0 \\ 1A_0^0 \end{bmatrix} \begin{bmatrix} 4B_1^1 \\ 0B_1^1 \end{bmatrix} \begin{bmatrix} 5A_0^0 \\ 0A_0^0 \end{bmatrix} \begin{bmatrix} 5A_1^1 \\ 1A_1^1 \end{bmatrix}$ . Для него представление в виде операций, моделирующих раунды, и декомпозиция до уровня элементарных операций будет выглядеть так:

$$\tilde{\alpha} = \begin{bmatrix} 5A_0^0 \\ 1A_0^0 \end{bmatrix} \begin{bmatrix} 4B_1^1 \\ 0B_1^1 \end{bmatrix} \begin{bmatrix} 5A_0^0 \\ 0A_0^0 \end{bmatrix} \begin{bmatrix} 5A_1^1 \\ 1A_1^1 \end{bmatrix} \implies D_1, D_2, D_3, D_4, \text{ где}$$

$$D_1 = U_5^\nabla U_5^\times U_5^-,$$

$$D_2 = U_4^\nabla U_4^\downarrow U_{A,4}^+,$$

$$D_3 = U_5^\nabla,$$

$$D_4 = U_5^\nabla U_5^\downarrow U_5^- U_{A,4}^+.$$



### 3.4.4 Расчет распределения числа активных меток

Имея последовательность операций для размеченного сценария, можно рассчитать распределение вероятностей числа активных меток перед началом каждого раунда. Как отмечалось выше, случайный процесс  $\eta_r$  — неоднородный марковский. Время в этом процессе увеличивается на единицу на каждом раунде, а матрицами переходов являются  $D_1, D_2, \dots, D_R$ . Так как после выполнения сценария система моделируется опять с начала размеченного сценария, то после операции  $D_R$  будут опять применяться операции  $D_1, D_2, \dots, D_R$ . В общем случае можно записать это так:

$$D_r \equiv D_{(r-1) \pmod{R} + 1}.$$

Если процесс  $\eta_r$  на  $r$ -м шаге имеет распределение  $\boldsymbol{\pi}^{(r)}$ , то после  $r$ -го раунда распределение будет  $\boldsymbol{\pi}^{(r)} D_r$ , после  $r+1$  шага —  $\boldsymbol{\pi}^{(r)} D_r D_{r+1}$  и так далее. Так как после  $R$ -й переходной матрицы опять используется  $D_1$ , то распределение процесса на  $(r+R)$ -м шаге будет

$$\boldsymbol{\pi}^{(r+R)} = \boldsymbol{\pi}^{(r)} D_r D_{r+1} \cdots D_R D_1 \cdots D_{r-1}.$$

Обозначим произведение матриц, стоящих в правой части, как  $\tilde{D}_r$ , то есть положим

$$\tilde{D}_r = D_r D_{r+1} \cdots D_R D_1 \cdots D_{r-1}.$$

Рассмотрим последовательность значений процесса  $\{\eta_r\}$  на шагах  $r, r+R, r+2R, \dots$  и обозначим эту последовательность как  $\{\eta_i^{[r]}\}_{i=0}^{\infty}$ ,  $\eta_i^{[r]} = \eta_{r+Ri}$ . Так как процесс  $\{\eta_i^{[r]}\}$  является вложенным процессом в марковский процесс  $\{\eta_r\}$ , то он и сам является марковским, причем в отличие от  $\{\eta_r\}$  однородным: переходные матрицы между шагами задаются матрицей  $\tilde{D}_r$ . Найти его стационарное распределение  $\boldsymbol{\pi}^{(r)}$  можно из системы линейных уравнений:

$$\begin{cases} \boldsymbol{\pi}^{(r)} \tilde{D}_r &= \boldsymbol{\pi}^{(r)}, \\ \sum_{n=0}^{\bar{N}} \pi_n^{(r)} &= 1. \end{cases}$$

Из найденного вектора  $\boldsymbol{\pi}^{(r)}$  можно найти  $\boldsymbol{\pi}^{(r+1)} = \boldsymbol{\pi}^{(r)} D_r$ ,  $\boldsymbol{\pi}^{(r+2)} = \boldsymbol{\pi}^{(r+1)} D_{r+1}$  и так далее. Таким образом, рассчитать распределение вероятностей перед каждым раундом  $r = 1, 2, \dots, R$  можно, решив следующую систему уравнений:

$$\begin{cases} \boldsymbol{\pi}^{(1)} \tilde{D}_1 &= \boldsymbol{\pi}^{(1)}, \\ \boldsymbol{\pi}^{(1)} D_1 &= \boldsymbol{\pi}^{(2)}, \\ \boldsymbol{\pi}^{(2)} D_2 &= \boldsymbol{\pi}^{(3)}, \\ \dots & \\ \boldsymbol{\pi}^{(R-1)} D_{R-1} &= \boldsymbol{\pi}^{(R)}, \\ \sum_{n=0}^{\bar{N}} \pi_n^{(1)} &= 1. \end{cases} \quad (3.16)$$

Используя найденные распределения числа активных меток, с помощью формул (3.8) и (3.9) можно вычислить новые оценки длительностей раундов  $\tau_r = \mathbb{M}\tau_r(n)$ .

### 3.4.5 Итерационный расчет оценок длительностей раундов

В результате расчета  $\tau_r$  по найденным распределениям  $\{\pi^{(r)}\}_{r=1}^R$  оценки длительностей раундов могут измениться. Если изначально раунды предполагались слишком короткими или длинными, то после того, как будет найдено распределение числа активных меток, их длительности могут соответственно увеличиться или уменьшиться. Так как моменты поступления меток  $\{a_i\}$  и закон изменения числа меток  $N(t)$  не зависят от длительностей раундов, то может измениться и размеченный сценарий.

Рассмотрим пример. Пусть задан сценарий  $\alpha = A^0 A^0 A^0 \dots$ , в начальный момент в системе две метки и новая метка поступает на 120-й миллисекунде ( $a_1 = 120$ ). Предположим изначально, что первые два раунда имеют длительность 100 миллисекунд (то есть  $\tau_1 = \tau_2 = 100$ ). Тогда размеченный сценарий будет иметь вид  $\tilde{\alpha} = [{}^2A][{}^2A_1][{}^3A] \dots$ . Если после пересчета выяснится, что раунды имеют длительности  $\tau_1 = 150$  и  $\tau_2 = 95$ , то новый размеченный сценарий будет иметь вид  $\tilde{\alpha}' = [{}^2A_1][{}^3A_1] \dots$ . Соответственно, изменятся и матрицы переходных вероятностей операций  $D_r$ , поэтому в итоге могут измениться и оценки  $\{\pi^{(r)}\}$ , и  $\{\tau_r\}$ . Таким образом, может потребоваться многократный пересчет оценок  $\tau_r$ .

Итерационный алгоритм, позволяющий найти оценки длительностей раундов  $\{\tau_r\}$ , принимает на вход сценарий  $\alpha = \alpha_1 \alpha_2 \dots \alpha_R$ , закон изменения числа меток  $N(t)$ , максимально допустимую ошибку  $\varepsilon > 0$  и максимальное число итераций  $\bar{K}$ . Алгоритм выполняет следующие действия (см. рис. 3.7).

*Шаг 1.* Произвольно выбираются начальные оценки длительностей раундов  $\tau_1, \tau_2, \dots, \tau_R$ , а счетчик числа итераций  $k$  выставляется равным единице.

*Шаг 2.* Вычисляется число меток в системе в начале первого раунда  $N_1$ , время начала каждого раунда  $t_r = T_0 + \tau_1 + \tau_2 + \dots + \tau_{r-1}$ , число поступивших и покинувших систему за  $r$ -й раунд меток  $\Delta_r^+ = |\{i : t_r \leq a_i < t_{r+1}\}|$  и  $\Delta_r^- = |\{i : t_r \leq a_i + T_L < t_{r+1}\}|$ . Затем вычисляется число меток во втором и последующих раундах  $N_r = N_{r-1} + \Delta_{r-1}^+ - \Delta_{r-1}^-$  (это же число можно получить как  $N_r = N(t_r)$ ).

*Шаг 3.* По найденным значениям  $\{N_r\}$ ,  $\{\Delta_r^+\}$ ,  $\{\Delta_r^-\}$  и заданному сценарию  $\alpha = \alpha_1 \alpha_2 \dots \alpha_R$  строится размеченный сценарий  $\tilde{\alpha} = \tilde{\alpha}_1 \tilde{\alpha}_2 \dots \tilde{\alpha}_R$ ,  $\tilde{\alpha}_r = [{}_{\Delta_r^-}^{N_r} X_{\Delta_r^+}^{e_r}]$ .

*Шаг 4.* С помощью найденного размеченного сценария строятся матрицы переходных вероятностей  $D_1, D_2, \dots, D_R$  процесса  $\{\eta_r\}_{r=1}^\infty$ . Для нахождения матриц  $D_r$  (см. блок-схему на рис. 3.6) используются матрицы элементарных операций, которые задаются выражениями (3.11), (3.12), (3.13), (3.14) и (3.15).

*Шаг 5.* Вычисляется стационарное распределение числа активных меток перед каждым раундом  $\pi^{(1)}, \pi^{(2)}, \dots, \pi^{(R)}$  как решение системы линейных алгебраических уравнений (3.16).

*Шаг 6.* С помощью найденных распределений вероятностей  $\{\pi^{(r)}\}_{r=1}^R$  вычисляются новые оценки длительностей раундов  $\tau'_1, \tau'_2, \dots, \tau'_R$ , где  $\tau'_r = \sum_{n=0}^{\bar{N}} \pi_n^{(r)} \tau(n)$ , а длительность раунда для заданного числа активных меток  $\tau(n)$  вычисляется с помощью (3.8).

*Шаг 7.* Вычисляется среднеквадратичное отклонение между предыдущей и новой оценкой длительностей раундов  $\sigma = \sqrt{\sum_{r=1}^R (\tau'_r - \tau_r)^2 / R}$ . Если  $\sigma \leq \varepsilon$  или число итераций  $k$



Рисунок 3.7 — Итерационный алгоритм расчета длительностей раундов

достигло максимального  $\bar{K}$ , то последние вычисленные оценки  $\{\tau'_r\}_{r=1}^R$  возвращаются в качестве результата выполнения алгоритма, и он завершается. В противном случае, если  $\sigma > \epsilon$  и  $k < \bar{K}$ , счетчик числа итераций  $k$  увеличивается на единицу, найденные оценки  $\{\tau'_r\}$  используются вместо исходных  $\{\tau_r\}$ , и алгоритм переходит на шаг 2.

Следует сделать несколько замечаний относительно работы алгоритма. Во-первых, в общем случае алгоритм может не сойтись к определенному набору значений  $\{\tau_r\}_{r=1}^R$  со сколь угодно малой точностью. Это связано с возможным заикливанием: при одной оценке события изменения числа меток происходят в одних раундах, при пересчете оценок — в других; но после еще одной или нескольких переоценок длительностей изменения числа меток опять начинают происходить в тех же раундах, что и ранее. Для того, чтобы алгоритм наверняка завершился, введено ограничение на число итераций. Отметим, что в ходе интенсивных чис-

ленных экспериментов алгоритм завершался всегда, причем в подавляющем большинстве случаев ему хватало всего двух итераций.

Во-вторых, в качестве входа стоит использовать не сам сценарий  $\alpha = \alpha_1 \alpha_2 \dots \alpha_R$ , а его многократно увеличенную версию вида

$$\alpha^{[L]} = \underbrace{\alpha \dots \alpha}_{L \text{ повторов}} = \underbrace{\alpha_1 \dots \alpha_R \alpha_1 \dots \alpha_R \dots \alpha_1 \dots \alpha_R}_{LR \text{ символов}}.$$

Это приходится делать для того, чтобы в моделируемый участок времени попало достаточное число событий поступления и выхода меток из системы, так как обычно  $a_{i+1} - a_i \gg \tau_{max}$ . Чем больше  $L$ , тем точнее будут полученные результаты. Нужно, однако, учесть, что увеличение длины сценария ведет к увеличению времени расчета.

В-третьих, желательно начинать моделирование с того момента  $T_0$ , когда в системе уже находится некоторое количество меток. Так как переходная матрица  $D_R$  строится по символам расширенного сценария  $\tilde{\alpha}_R = \begin{bmatrix} N_R & X_{\Delta_R}^{eR} \\ \Delta_R^- & \Delta_R^+ \end{bmatrix}$  и  $\tilde{\alpha}_1 = \begin{bmatrix} N_1 & Y_{\Delta_1}^{e1} \\ \Delta_1^- & \Delta_1^+ \end{bmatrix}$ , то если начинать моделирования с момента  $T_0 = 0$ , для которого  $N(T_0) = 0$ , а в среднем в системе находится  $N_a = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t=0}^T N(t) dt > 1$  меток, то в результате перехода вся информация о том, в каких состояниях находились метки, потеряется, и распределение вероятностей будет  $\pi^{(R+1)} \equiv \pi^{(0)} \equiv (1, 0, \dots, 0)$ ; все метки после  $R$ -го раунда как будто внезапно исчезнут из системы. Чтобы этого избежать, желательно начинать моделирование с некоторого момента  $T_0$ , в который  $N(T_0) \approx N_a$ . Например, если метки поступают через равные промежутки времени  $\Delta t$ , можно выбрать такой момент, когда в области чтения находится  $\lfloor T_L / \Delta t \rfloor$  меток.

После того, как найдены оценки длительностей раундов  $\{\tau_r\}$ , числа меток в системе в начале каждого раунда  $\{N_r\}$  и числа покидающих и поступающих меток  $\{\Delta_r^-\}$  и  $\{\Delta_r^+\}$ , можно рассчитать вероятность идентификации отдельной метки.

### 3.5 Расчет вероятности идентификации

После выполнения итерационного алгоритма, известны оценки длительностей раундов  $\{\tau_r\}_{r=1}^R$ , размеченный сценарий  $\tilde{\alpha} = \tilde{\alpha}_1 \tilde{\alpha}_2 \dots \tilde{\alpha}_R$ ,  $\tilde{\alpha}_r = \begin{bmatrix} N_r & X_{\Delta_r}^{e_r} \\ \Delta_r^- & \Delta_r^+ \end{bmatrix}$  и распределения числа активных меток в каждом раунде  $\{\pi^{(r)}\}_{r=1}^R$ . Для расчета вероятности идентификации рассмотрим проезд области чтения одной выделенной меткой. Система при этом ведет себя как обычно: считыватель работает, а метки поступают и покидают систему по сценарию  $\tilde{\alpha}$ . Выберем такой раунд  $r_0$ , что  $\Delta_{r_0-1}^+ > 0$ , то есть перед раундом  $r_0$  в систему поступила новая метка. Время нахождения метки в области известно и равно  $T_L$  по условию задачи. Определим число  $Q_{r_0}$  как количество раундов, в течение которых метка, поступившая в раунде  $r_0$  находится в системе, то есть:

$$\tau_{r_0} + \tau_{r_0+1} + \dots + \tau_{r_0+Q_{r_0}-2} < T_L \leq \tau_{r_0} + \tau_{r_0+1} + \dots + \tau_{r_0+Q_{r_0}-1},$$

где  $\tau_r \equiv \tau_{(r-1) \pmod R + 1}$ , то есть  $\tau_{R+1} \equiv \tau_1$ ,  $\tau_{R+2} \equiv \tau_2$  и так далее.

Рассмотрим конечный двумерный случайный процесс  $\{(\eta_r^{[r_0]}, \varphi_r^{[r_0]})\}_{r=1}^{Q_{r_0}}$ , в котором  $\eta_r^{[r_0]} \equiv \eta_{r+r_0-1}$  — число активных меток в  $(r + r_0 - 1)$ -м раунде, а  $\varphi_r$  — состояние выделенной метки: если метка еще ни разу не передала успешно свой идентификатор (EPCID или TID), то  $\varphi_r^{[r_0]} = 0$ , если метка неактивна, и  $\varphi_r^{[r_0]} = 1$ , если активна. Если же выделенная метка хотя бы раз успешно передала свой идентификатор, то  $\varphi_r^{[r_0]} = 2$  независимо от того, активна ли она, то есть  $\varphi_r^{[r_0]} = 2$  — поглощающее состояние процесса  $\{\varphi_r\}$ . Вероятность идентификации наблюдаемой метки при условии, что она поступила в раунде  $r_0$ , можно выразить как  $\mathbb{P}\{\varphi_{Q_{r_0}}^{[r_0]} = 2\}$ .

Номер раунда  $r_0$ , в котором поступила наблюдаемая метка, определяется не однозначно. Разумно предположить, что вероятность выбора такого раунда пропорциональна его длительности. Положим  $\Delta_0^+ = \Delta_R^+$  и обозначим множество всевозможных номеров раундов, в которых происходило увеличение числа меток в системе, как  $\mathfrak{R}$ :

$$\mathfrak{R} = \{r \mid r \in [1, R], \Delta_{r-1}^+ > 0\},$$

а вероятность поступления метки в раунде  $r_0 \in \mathfrak{R}$  оценим как:

$$p_{r_0}^{[a]} = \frac{\tau_{r_0}}{\sum_{r \in \mathfrak{R}} \tau_r}. \quad (3.17)$$

Используя введенные обозначения, вероятность идентификации метки можно выразить с помощью полной вероятности:

$$P_X = \sum_{r \in \mathfrak{R}} p_r^{[a]} \mathbb{P}\{\varphi_{Q_r}^{[r]} = 2\}. \quad (3.18)$$

Таким образом, для нахождения вероятности идентификации метки нужно найти вероятности идентификации метки  $\mathbb{P}\{\varphi_{Q_{r_0}}^{[r_0]} = 2\}$  при условии поступления ее в раунде  $r_0 \in \mathfrak{R}$ . Стоит отметить, что процесс  $\{\varphi_r^{[r_0]}\}$  не является марковским, так как вероятности его переходов между состояниями 0 и 1 зависят не только от его текущего состояния, но и от того, сколько в системе активных меток. Например, если  $\varphi_r^{[r_0]} = 0$  (наблюдаемая метка неактивна) и других активных меток в системе нет, то вероятность перехода  $\varphi_r^{[r_0]}$  в состояние 1 или 2 определяется только вероятностью успешной передачи ответа RN16 меткой в очередном раунде; если же в системе есть активные метки, то нужно также учитывать вероятность коллизий.

### 3.5.1 Определение основного процесса

Рассмотрим двумерный процесс  $\{(\eta_r^{[r_0]}, \varphi_r^{[r_0]})\}_{r=1}^{Q_{r_0}}$ . Во всех дальнейших расчетах вплоть до вычисления вероятности  $P_X$  будем использовать процесс, начинающийся в раунде  $r_0$ , поэтому, для сокращения записи, будем опускать верхний индекс  $[r_0]$ , но подразумевать его. Таким образом, вместо  $\varphi_r^{[r_0]}$  будем писать  $\varphi_r$ , вместо  $Q_{r_0}$  — просто  $Q$  и так далее.

Изменения компонентов процесса  $\{(\eta_r, \varphi_r)\}$  не являются независимыми. Например, если  $\eta_r = 0$ , вероятность того, что  $\varphi_r = 1$  равна нулю: наблюдаемая метка не может

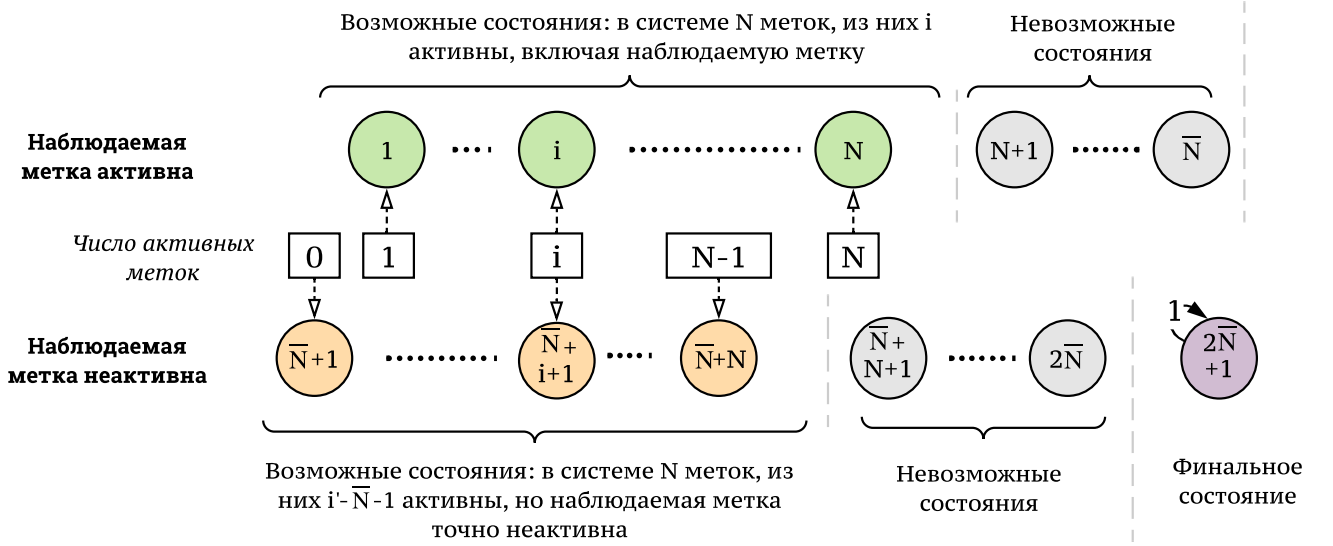


Рисунок 3.8 — Структура пространства состояний основного процесса  $\{\gamma_r\}$ , когда в системе  $N$  меток

быть активной, если активных меток в системе нет. Чтобы избавиться от недостижимых состояний вместо двумерного процесса будем рассматривать одномерный процесс  $\{\gamma_r\}_{r=1}^Q$ ,  $\gamma_r \in [1, 2\bar{N} + 1]$  (см. рис. 3.8), в котором:

$$\begin{aligned}
 1 \leq \gamma_r \leq \bar{N} &\Leftrightarrow \varphi_r = 1 \text{ и } \eta_r = \gamma_r; \\
 \bar{N} + 1 \leq \gamma_r \leq 2\bar{N} &\Leftrightarrow \varphi_r = 0 \text{ и } \eta_r = \gamma_r - \bar{N} - 1; \\
 \gamma_r = 2\bar{N} + 1 &\Leftrightarrow \varphi_r = 2.
 \end{aligned} \tag{3.19}$$

Попадание в состояние  $\gamma_r = 2\bar{N} + 1$  означает, что наблюдаемая метка успешно передала свой идентификатор, это состояние является поглощающим. При  $\gamma_r \leq \bar{N}$  число активных меток  $\eta_r$  равно  $\gamma_r$ , причем наблюдаемая метка активна. Если же процесс  $\gamma_r$  находится в состоянии из интервала  $[\bar{N} + 1, 2\bar{N}]$ , то наблюдаемая метка неактивна, и число активных меток  $\eta_r$  определяется как  $\gamma_r - (\bar{N} + 1)$ . Легко видеть, что для состояний  $\eta_r = 0$ ,  $\varphi_r = 1$  (активных меток в системе нет, но наблюдаемая метка активна), а также  $\eta_r = \bar{N}$ ,  $\varphi_r = 0$  (все метки активны, но наблюдаемая метка неактивна), нет соответствующих состояний  $\gamma_r$ ; кроме того, значения  $\eta_r$  неразличимы при  $\gamma_r = 2\bar{N} + 1$ .

Переходные вероятности для процесса  $\{\gamma_r\}$  найдем аналогично тому, как ранее определялись переходные вероятности для случайного процесса  $\{\eta_r\}_{r=1}^\infty$ : сначала покажем, что вероятности переходов  $\{\gamma_r\}$  при выполнении элементарных операций зависят только от текущего состояния и выпишем их, а затем покажем, как построить матрицы переходных вероятностей между раундами.

### 3.5.2 Переходные вероятности основного процесса для элементарных операций

Для обозначений переходных вероятностей процесса  $\{\gamma_r\}$  при выполнении элементарных операций будем использовать нотацию, аналогичную обозначениям для процесса  $\{\eta_r\}$  ( $U_N^\nabla, U_N^\times, \dots$ ), но вместо буквы  $U$  использовать  $V$ . Как и в разделе 3.4.2, здесь для удобства будем предполагать, что состояние процесса  $\{\gamma_r\}$  изменяется при выполнении одной из элементарных операций: инвентаризации ( $V_N^\nabla$ ), смены флага ( $V_N^\times$ ), сброса питания ( $V_N^\downarrow$ ), добавления или удаления метки ( $V_{X,N}^+$  и  $V_N^-$  соответственно). В дальнейшем же шаг процесса будет, как обычно, соответствовать раунду.

Пусть  $V$  — одна из элементарных операций, состояние процесса  $\gamma_r$  перед ее выполнением есть  $\gamma_r = i$ , а после —  $\gamma_{r+1} = j$ ,  $i, j \in [1, 2\bar{N} + 1]$ . Как и ранее для процесса  $\{\eta_r\}$ , в силу сделанных допущений, в частности — о постоянстве BER во всей области чтения, каждую элементарную операцию можно задать распределением вероятностей  $\mathbb{P}\{\gamma_{r+1} = j | \gamma_r = i\}$ . Эти вероятности определяются аналогично тому, как это было сделано для переходных вероятностей  $\{\eta_r\}$ , однако здесь приходится отдельно рассматривать переходы при  $\phi_r = 0$  и  $\phi_r = 1$ . Рассмотрим переходные матрицы  $V \in \mathbb{R}^{(2\bar{N}+1) \times (2\bar{N}+1)}$  каждой из элементарных операций (см. рис. 3.4). Как и в разделе 3.4.2, некорректные состояния  $j \in [N + 1, \bar{N}] \cup [\bar{N} + N + 1, 2\bar{N}]$ , которые соответствуют  $\eta_r > N$  (то есть в которых активных меток больше, чем всего меток в системе), не будут достижимы из корректных состояний  $i \in [1, N] \cup [\bar{N} + 1, \bar{N} + N]$ , а у них самих будут только переходы-петли с вероятностью 1. Единственное исключение из этого правила — операция добавления метки  $V_{A,N}^+$ . Переходные вероятности при выполнении элементарных операций показаны на рис. 3.9 и рис. 3.10.

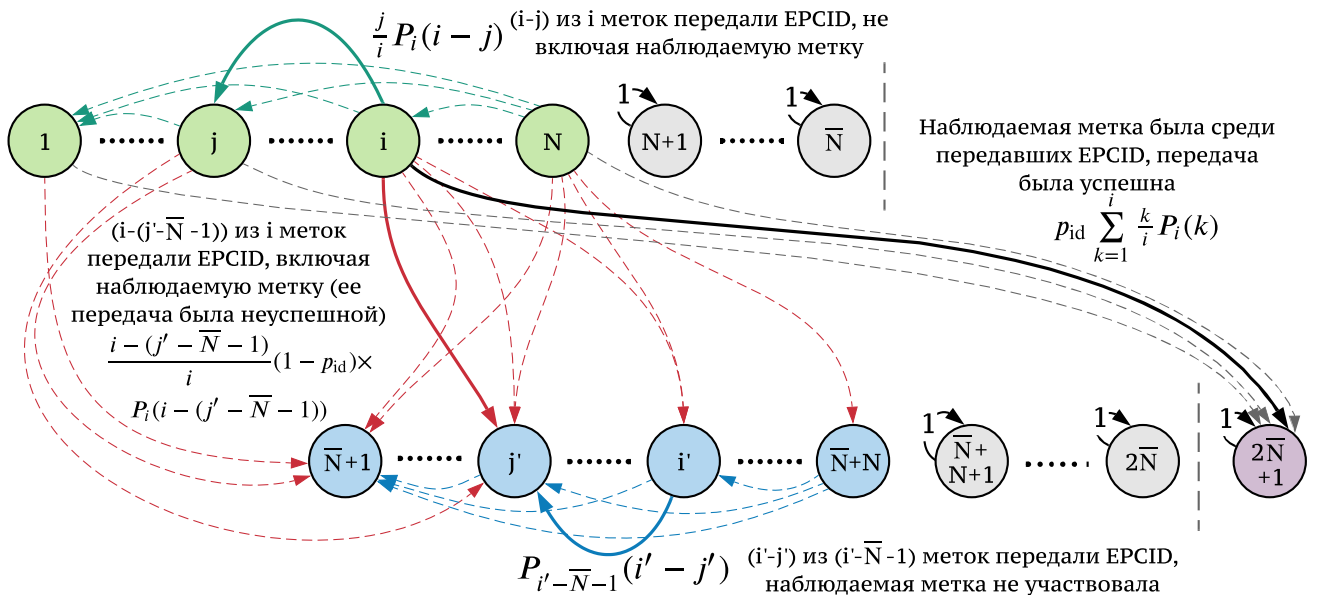


Рисунок 3.9 — Переходные вероятности основного процесса  $\{\gamma_r\}$  при выполнении операции инвентаризации  $V_N^\nabla$

Применение операции опроса меток  $V_N^\nabla$  приводит к тому, что часть активных меток успешно передают RN16 и затем пытаются передать EPCID. Если одной из таких меток

оказалась наблюдаемая метка (то есть  $n \leq \bar{N}$ ), то процесс может перейти в поглощающее состояние, если наблюдаемая метка сможет полностью передать свой идентификатор. Если же ей это не удастся, то она может либо остаться активной (случай неуспешной передачи RN16), либо инвертировать флаг, но ошибиться уже после передачи RN16. При построении вероятностей удобно рассматривать три независимых события:  $(i - j)$  из  $i$  активных меток успешно передали свои RN16 (вероятность  $P_i(i - j)$ ); наблюдаемая метка попала в число меток, успешно передавших RN16 (вероятность  $(i - j)/i$ ); наблюдаемая метка успешно передала свой идентификатор (вероятность  $p_{id}$ ). С учетом этого замечания и определения состояний процесса  $\{\gamma_r\}$  (см. (3.19)), вероятности определяются следующим образом (см. рис. 3.9):

$$\{V_N^\nabla\}_{ij} = \begin{cases} \frac{i}{j} P_i(i - j), & 1 \leq j \leq i \leq N \\ P_{i-\bar{N}-1}(i - j), & \bar{N} + 1 \leq j \leq i \leq \bar{N} + N \\ \frac{i-(j-\bar{N}-1)}{i} (1 - p_{id}) P_i(i - (j - \bar{N} - 1)), & 1 \leq i \leq N, \bar{N} + 1 \leq j \leq \bar{N} + N \\ p_{id} \sum_{k=1}^i \frac{k}{i} P_i(k), & 1 \leq i \leq N, j = 2\bar{N} + 1 \\ 1, & N < i = j \leq \bar{N} \\ 1, & \bar{N} + N < i = j \leq 2\bar{N} + 1 \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.20)$$

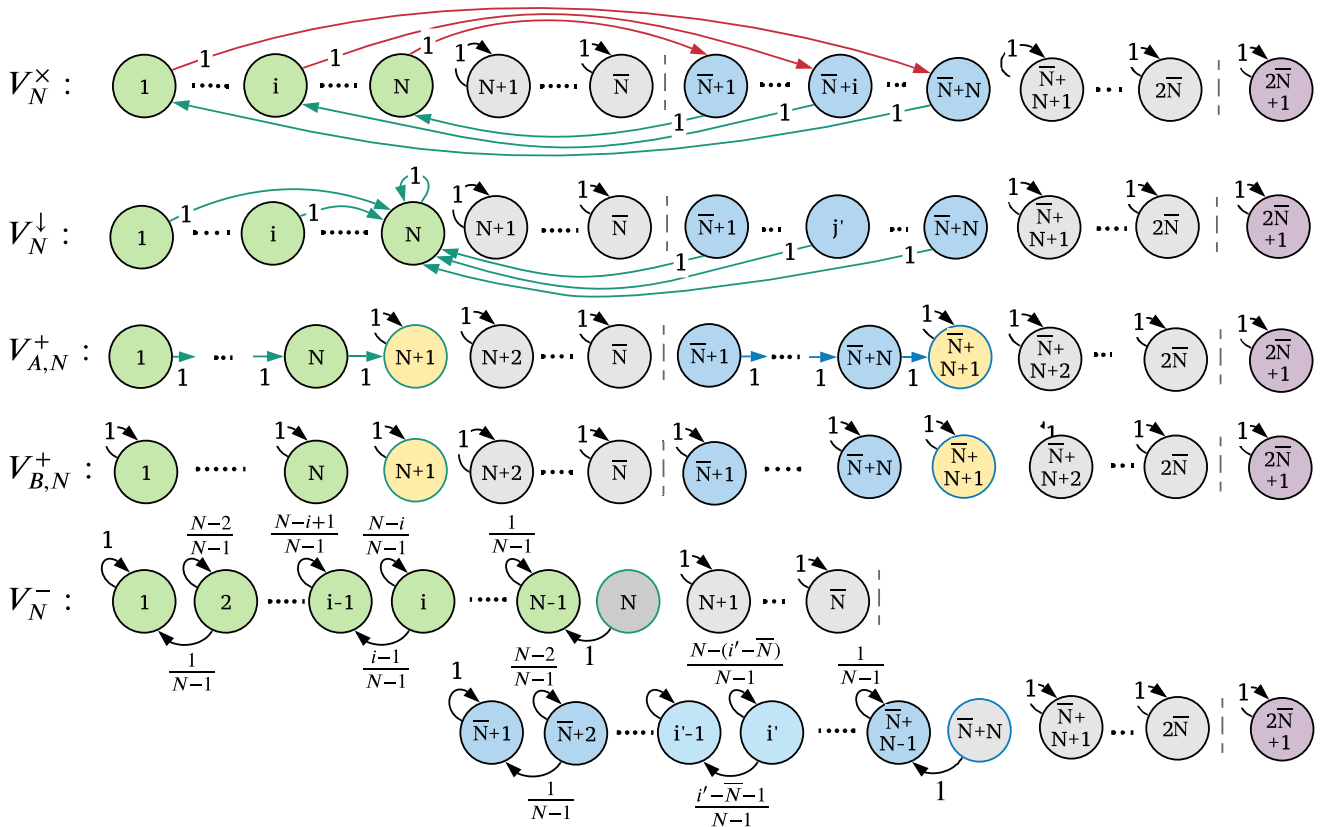


Рисунок 3.10 — Переходные вероятности основного процесса  $\{\gamma_r\}$  при выполнении элементарных операций



При смене флага  $V_N^\times$  активные и неактивные метки меняются местами. Это справедливо и для наблюдаемой метки, поэтому значение  $\varphi_r$  меняется с 1 на 0 и обратно. Учитывая определение (3.19), получаем следующие значения переходных вероятностей (см. рис. 3.10):

$$\{V_N^\times\}_{ij} = \begin{cases} 1, & 1 \leq i \leq N, j = \bar{N} + N + 1 - i \\ 1, & \bar{N} + 1 \leq i \leq \bar{N} + N, j = N - (i - \bar{N} - 1) \\ 1, & N < i = j \leq \bar{N} \\ 1, & \bar{N} + N < i = j \leq 2\bar{N} + 1 \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.21)$$

В результате выключения питания  $V_N^\downarrow$  все метки сбрасывают свои флаги в  $A$ , на него же переключается считыватель. Поэтому из любого корректного непоглощающего состояния  $i \in [1, N] \cup [\bar{N} + 1, \bar{N} + N]$  операция переводит процесс в состояние  $\gamma_{r+1} = N$ :

$$\{V_N^\downarrow\}_{ij} = \begin{cases} 1, & 1 \leq i \leq N, j = N \\ 1, & \bar{N} + 1 \leq i \leq \bar{N} + N, j = N \\ 1, & N < i = j \leq \bar{N} \\ 1, & \bar{N} + N < i = j \leq 2\bar{N} + 1 \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.22)$$

Операции добавления метки  $V_{A,N}^+$  и  $V_{B,N}^+$  никак не влияют на состояние наблюдаемой метки  $\varphi_r$  и изменяют число активных меток только в том случае, если текущий флаг опроса равен  $A$ :

$$\{V_{A,N}^+\}_{ij} = \begin{cases} 1, & 1 \leq i \leq N, j = i + 1 \\ 1, & \bar{N} + 1 \leq i \leq \bar{N} + N, j = i + 1 \\ 1, & N < i = j \leq \bar{N} \\ 1, & \bar{N} + N < i = j \leq 2\bar{N} + 1 \\ 0, & \text{в остальных случаях} \end{cases} \quad (3.23)$$

$$V_{B,N}^+ = I_{2\bar{N}+1}$$

Здесь  $I_{2\bar{N}+1}$  — единичная матрица порядка  $2\bar{N} + 1$ .

При моделировании выхода метки из области чтения  $V_N^-$  будем предполагать, что выйти может любая из меток, кроме наблюдаемой<sup>4</sup>. Про наблюдаемую же метку заведомо известно, что она остается в системе, поэтому при расчете вероятностей ее учитывать не нужно. В остальном, вероятности определяются так же, как и в (3.15), с поправкой на структуру состояний  $\{\gamma_r\}$  (см. рис. 3.10).

<sup>4</sup>Отметим, что из-за этого допущения возможна некоторая неточность, так как в действительности метки выходят в соответствии с их законом движения, и вероятность выхода метки с определенным флагом  $X$  может не быть равна отношению числа меток с флагом  $X$  к общему числу меток; но это допущение существенно упрощает модель и позволяет не учитывать положения меток.

$$\{V_N^-\}_{ij} = \begin{cases} \frac{i-1}{N-1}, & 1 < i \leq N, j = i - 1 \\ \frac{N-i}{N-1}, & 1 \leq i = j \leq N \\ \frac{i-\bar{N}-1}{N-1}, & \bar{N} + 1 < i \leq \bar{N} + N, j = i - 1 \\ \frac{N-(i-\bar{N})}{N-1}, & \bar{N} + 1 \leq i = j \leq \bar{N} + N \\ 1, & N < i = j \leq \bar{N} \\ 1, & \bar{N} + N < i = j \leq 2\bar{N} + 1 \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.24)$$

### 3.5.3 Матрицы переходных вероятностей между раундами для основного процесса

Здесь и далее будем считать, что состояние процесса  $\{\gamma_r\}$  меняется между раундами, а не между элементарными операциями. Рассмотрим размеченный сценарий  $\tilde{\alpha} = \tilde{\alpha}_1 \tilde{\alpha}_2 \dots \tilde{\alpha}_R$ ,  $\tilde{\alpha}_r = [\frac{N_r}{\Delta_r^-} X \frac{e_r}{\Delta_r^+}]$ . Будем считать, что наблюдаемая метка поступила в раунде  $r_0$ , для которого  $\Delta_{r_0}^+ > 0$ , и находилась в системе в течение  $Q_{r_0}$  раундов. Тогда для наблюдаемой метки работа считывателя будет задаваться размеченным сценарием  $\tilde{\alpha}^{[r_0]} = \tilde{\alpha}_1^{[r_0]} \tilde{\alpha}_2^{[r_0]} \dots \tilde{\alpha}_{Q_{r_0}}^{[r_0]}$ , где  $\tilde{\alpha}_i^{[r_0]} \equiv \tilde{\alpha}_{r_0+i-1}$ . Обозначим матрицы переходных вероятностей процесса  $\{\gamma_r^{[r_0]}\}$  между раундами  $r$  и  $r+1$ , как  $C_r^{[r_0]}$ .

Пусть  $\tilde{\alpha}_r^{[r_0]} = (X, N, e, \Delta^-, \Delta^+)$  и  $\tilde{\alpha}_{r+1}^{[r_0]} = (Y, \bullet, \bullet, \bullet, \bullet)$  (здесь  $\bullet$  — произвольное значение). Как и переходная матрица  $D_r$  фонового процесса  $\{\eta_r\}$ , матрица переходных вероятностей основного процесса  $C_r^{[r_0]}$  определяется значениям  $X, N, e, \Delta^+, \Delta^-$  и  $Y$ , и ее можно получить как произведение матриц элементарных операций  $V_N^\nabla, V_N^\times, V_N^\downarrow, V_{X,N}^+, V_N^-$ . Для этого можно использовать алгоритм, описанный в разделе 3.4.3 и показанный на рис. 3.6, заменяя матрицы  $U_N^\nabla, U_N^\times, U_N^\downarrow, U_{X,N}^+, U_N^-$  на матрицы элементарных операций для основного процесса  $V_N^\nabla, V_N^\times, V_N^\downarrow, V_{X,N}^+, V_N^-$ .

### 3.5.4 Расчет вероятности поглощения процесса

Зная переходные вероятности процесса  $\{\gamma_r^{[r_0]}\}$ , можно вычислить вероятность того, что метка, поступившая в систему в раунде  $r_0 \in \mathfrak{R}$  успешно передаст свой идентификатор. Обозначим  $\theta^{(r_0,r)} \in \mathbb{R}^{2\bar{N}+1}$  распределение вероятностей процесса  $\gamma_r^{[r_0]}$  на  $r$ -м шаге, а  $\theta^{(r_0,1)}$  — его начальное распределение. Так как вероятности переходов между  $r$ -м и  $(r+1)$ -м раундами определяются матрицей  $C_r^{[r_0]}$ , то

$$\theta^{(r_0,r+1)} = \theta^{(r_0,r)} C_r^{[r_0]} = \theta^{(r_0,1)} C_1^{[r_0]} C_2^{[r_0]} \dots C_r^{[r_0]}.$$

В частности, распределение вероятностей после  $Q_{r_0}$  раундов можно вычислить как:

$$\boldsymbol{\theta}^{(r_0, Q_{r_0}+1)} = \boldsymbol{\theta}^{(r_0, 1)} C_1^{[r_0]} C_2^{[r_0]} \dots C_{Q_{r_0}}^{[r_0]}.$$

В силу определения состояний процесса  $\{\gamma_r^{[r_0]}\}$  (3.19), вероятность попадания процесса  $\{\varphi_r^{[r_0]}\}$  в поглощающее состояние  $\varphi_r^{[r_0]} = 2$  тождественно равна вероятности попадания процесса  $\{\gamma_r^{[r_0]}\}$  в его поглощающее состояние  $\gamma_{Q_{r_0}}^{[r_0]} = 2\bar{N} + 1$ , то есть

$$\mathbb{P}\{\varphi_{Q_{r_0}}^{[r_0]} = 2\} \equiv \mathbb{P}\{\gamma_{Q_{r_0}}^{[r_0]} = 2\bar{N} + 1\} = \boldsymbol{\theta}_{2\bar{N}+1}^{(r_0, Q_{r_0}+1)}.$$

Таким образом, для вычисления вероятности поглощения процесса  $\{\varphi_r^{[r_0]}\}$ , то есть успешной идентификации наблюдаемой метки, поступившей в систему в раунде  $r_0$ , нужно вычислить распределение вероятностей после выполнения  $Q_{r_0}$  раундов. Так как размеченный сценарий  $\tilde{\boldsymbol{\alpha}}$ , по которому строятся матрицы переходных вероятностей  $C_1^{[r_0]}, C_2^{[r_0]}, \dots, C_{Q_{r_0}}^{[r_0]}$ , был построен одновременно с вычислением оценок длительностей раундов  $\tau_1, \tau_2, \dots, \tau_R$  в результате выполнения итерационного алгоритма (см. раздел 3.4.5), то неизвестным остается только начальное распределение  $\boldsymbol{\theta}^{(r_0, 1)}$ .

Для нахождения начального распределения  $\boldsymbol{\theta}^{(r_0, 1)}$  нужно учесть, что об этом распределении уже кое-что известно. Во-первых, известно, будет ли наблюдаемая метка активна сразу после поступления: она будет активной, если и только если в раунде  $r_0$  считыватель ведет опрос по флагу  $A$ , поскольку новая метка всегда хранит флаг со значением  $A$ . Значит, если  $\boldsymbol{\alpha}_{r_0} = A^e$ , то значение  $\varphi_1^{[r_0]} = 0$ , и, соответственно,  $\gamma_1^{[r_0]} \in [1, N]$ ; если же  $\boldsymbol{\alpha}_{r_0} = B^e$ , то  $\varphi_1^{[r_0]} = 1$  и  $\gamma_1^{[r_0]} \in [\bar{N} + 1, \bar{N} + N]$ . Во-вторых, известно число меток в области чтения  $N$  известно, так как известен размеченный сценарий  $\tilde{\boldsymbol{\alpha}}$ . Наконец, в-третьих, известно распределение числа активных меток  $\eta_{r_0}$  в начале раунда  $\boldsymbol{\pi}^{(r_0)}$  — оно было найдено в результате выполнения итерационного алгоритма. Таким образом:

$$\boldsymbol{\theta}_n^{(r_0, 1)} = \begin{cases} \pi_n^{(r_0)}, & \tilde{\boldsymbol{\alpha}}_{r_0} = A_N^e, 1 \leq n \leq N \\ \pi_{n-(\bar{N}+1)}^{(r_0)}, & \tilde{\boldsymbol{\alpha}}_{r_0} = B_N^e, \bar{N} + 1 \leq n \leq \bar{N} + N \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3.25)$$

Подставляя найденные выражения для вероятности попадания  $\{\gamma_r^{[r_0]}\}$  в поглощающее состояние в выражение (3.18), получаем, что вероятность идентификации наблюдаемой метки  $P_X$  вычисляется так:

$$P_X = \sum_{r_0 \in \mathfrak{R}} p_{r_0}^{[a]} \{\boldsymbol{\theta}^{(r_0, 1)} C_1^{[r_0]} C_2^{[r_0]} \dots C_{Q_{r_0}}^{[r_0]}\}_{2\bar{N}+1},$$

где вероятности  $p_{r_0}^{[a]}$  вычисляются согласно (3.17).

### 3.6 Результаты моделирования

Для численного исследования был выбран простой случай поступления меток через равные интервалы. С помощью аналитической модели исследовались три характеристики:

- распределение числа активных меток по раундам;
- оценка длительностей раундов;
- вероятность идентификации отдельной метки.

Первые две характеристики получаются аналитически с помощью фонового процесса, а третья — с помощью основного процесса. Для валидации результатов аналитической модели была разработана простая имитационная модель, моделирующая процесс проезда метками области чтения и обмен командами и ответами при постоянном BER. Отметим, что эта модель гораздо проще и быстрее той имитационной модели, которая была описана в предыдущей главе.

Исходный код аналитической и имитационной модели на языке Python 3 с использованием Cython, результаты экспериментов и более подробное описание доступны на GitHub: <https://github.com/larioandr/thesis-rfidam>.

Перед тем, как перейти к описанию параметров эксперимента и его результатов, сделаем одно замечание. В коде и выводе неудобно использовать обозначения спецификаций раундов вида  $X^e$ , которые использовались при построении математической модели. Чтобы избежать верхних индексов и оставить сценарии читаемыми, была использована следующее кодирование в виде строк: если после раунда нет сброса питания, то раунд обозначается только заглавной буквой А или В, а если после раунда считыватель выключается, то после А или В добавляется строчная буква х. Таким образом, сценарий  $A^0B^0A^1B^0$  в реализации модели кодируется строкой “АВАхВ”.

#### 3.6.1 Параметры модели

Все параметры, которые используются в эксперименте, можно разделить на три группы: *гиперпараметры*, *метапараметры* и *свободные параметры*.

*Гиперпараметры* влияют на точность результатов и скорость расчетов. В модели используется три гиперпараметра:

- число моделируемых меток;
- кратность расширенных сценариев;
- множитель раундов при свертке.

Кратность расширенных сценариев определяет, сколько раз копируется сценарий при построении расширенного сценария (см. раздел 3.4.5). Чем выше кратность, тем больше моментов поступления и выхода меток попадет в моделируемый (аналитической моделью) интервал,

Таблица 4 — Параметры аналитической модели RFID

Параметр	Значение
<i>Гиперпараметры</i>	
Число моделируемых меток	3000
Кратность расширенных сценариев ( $K$ )	2000
Множитель раундов при свертке ( $K_c$ )	3
<i>Метапараметры</i>	
Интервал $T_{ar}$	12,5 мкс
Интервал $RT_{cal}$	37,5 мкс
Интервал $TR_{cal}$	56,25 мкс
Показатель числа слотов ( $Q$ )	2
Длительность отключения ( $T_{\downarrow}$ )	100 мс
Число символов на бит ( $M$ )	2
Использовать расширенную преамбулу ( $TR_{ext}$ )	0 (нет)
Коэффициент DR (64/3 или 8)	64/3
Длина EPCID	96 бит
Длина TID	64 бит
Время нахождения в области чтения ( $T_L$ )	2,5 сек
<i>Свободные параметры</i>	
Интервалы между поступлениями	200 мс, 400 мс, 1 сек
BER	от 0 до 0,05
Сценарии	См. в конце раздела <a href="#">3.6.1</a>

и тем точнее будет результат. Однако при этом растет сложность расчета. Число моделируемых меток определяет, сколько событий поступлений новых меток будет моделироваться в имитационной модели. Множитель при свертке сценариев используется при исследовании характеристик раундов и будет описан подробнее в следующем разделе.

*Метапараметры* - это те величины, которые сами по себе не оказывают существенного влияния на результаты, и для их различных значений можно подобрать диапазоны других (мета- и обычных) параметров, на которых будут получены близкие результаты. Например, так как в модели явно задается BER, то параметр  $M$  (число символов на бит в ответах меток) влияет только на длительность сообщений, поэтому, если для  $M=8$  взять большую область чтения и увеличить интервал между метками пропорционально, то можно получить

тот же результат, что и для  $M=1$ , маленькой области чтения и коротких интервалов между поступлениями. К метапараметрам относятся:

- параметры передачи команд считывателя  $T_{ari}$ ,  $RT_{cal}$ ,  $TR_{cal}$ ;
- параметры передачи ответов меток  $M$ ,  $DR$ ,  $TR_{ext}$ ;
- длительность отключения считывателя  $T_{\downarrow}$ ;
- время нахождения метки в области чтения  $T_L$ ;
- число читаемых слов из TID и размер EPCID в байтах;
- показатель числа слотов  $Q$ .

Отметим, что если бы BER не был свободным параметром (например, свободным параметром был бы SNR), то параметры  $T_{ari}$ ,  $M$ ,  $DR$ ,  $TR_{ext}$  оказывали бы более существенное влияние и их можно было бы считать свободными, так как от них бы зависел BER. Также отметим, что величина  $Q$  влияет на вероятность коллизий, как и число меток в области чтения, а последнее зависит от интервала между поступлениями, который является свободным параметром.

*Свободные параметры* - это те параметры, которые наиболее существенно влияют на результат. К ним относятся:

- интервал между поступлениями меток;
- сценарий работы считывателя;
- вероятность битовой ошибки (BER).

При моделировании исследовались четыре группы сценариев:

- $A^0$  — базовый сценарий без сбросов питания и переключения флагов («всегда  $A$ »);
- $A^0 B^0$ ,  $A^0 A^0 A^0 A^0 B^0 B^0 B^0$  — сценарии без сброса питания, но с инвертированием флагов опроса;
- $A^1$ ,  $A^0 A^1$ ,  $A^0 A^0 A^0 A^1$  — сценарии без смены флагов, но со сбросом питания;
- $A^0 B^0 A^0 B^0$ ,  $A^0 A^0 A^0 A^1 A^0 B^0 A^0 B^1$  — смешанные сценарии.

Значения остальных свободных параметров, а также гиперпараметров и метапараметров приведены в табл. 4.

### 3.6.2 Анализ свойств раундов

В первую очередь с помощью фонового процесса были получены оценки длительностей раундов и среднее число активных меток в каждом раунде. Также с помощью основного процесса для каждого раунда были найдены оценки вероятности идентификации для меток, поступающих в этом раунде. Для валидации эти значения сравнивались с данными, полученными с помощью имитационного моделирования.

Из-за использования расширенных сценариев в аналитической модели исследуется очень большое число раундов (порядка десяти тысяч). В отличие от аналитики, в имитационной модели один и тот же сценарий (не расширенный) моделируется до тех пор, пока не будет промоделировано достаточное число меток. Так как имитационная модель учитывает положение и состояние конкретных меток, при каждом начале моделирования сценария

система будет находиться в новом состоянии. Число раундов в имитационной модели будет много больше, порядка нескольких миллионов. Из-за этого сравнивать оценки длительностей, среднего числа активных меток и вероятности идентификации для каждого отдельного раунда невозможно и нецелесообразно.

Чтобы решить эту проблему и получить наглядное сравнение оценок по раундам, вместо сравнения в каждом отдельном раунде будем сравнивать значения, полученные усреднением по подмножеству раундов, равноотстоящих от начала сценария, то есть “свернем” длинный вектор метрик в более короткий. Так как исходный сценарий  $\alpha$  может быть очень коротким, а расширенный сценарий  $\tilde{\alpha}$ , используемый в расчетах — слишком длинный, будем использовать сценарий  $\hat{\alpha}$ , полученный, как и расширенный сценарий, конкатенацией сценария  $\alpha$ , но меньшее число раз  $K_c$ . Это число  $K_c$  будем называть *множителем раундов при свертке*. Если кратность расширенных сценариев  $K \sim 1000$ , то множитель при свертке  $K_c \sim 10$  (например, в эксперименте использовалось  $K = 2000$  и  $K_c = 3$ ).

Пусть длина сценария равна  $R$ , в модели исследовалось  $\bar{R}$  раундов, и были получены оценки некоторой метрики  $x_1, x_2, \dots, x_{\bar{R}}$ . Для аналитики это означает, что  $\bar{R} = KR$ , а для имитационной модели — что для моделирования нужного числа меток потребовалось “проиграть”  $\bar{R}$  раундов. Тогда в качестве значения метрики  $X$  в раунде  $r$  примем число, равное  $(x_r + x_{r+RK_c} + \dots + x_{r+n_rRK_c})/n_r$ , где  $n_r$  таково, что  $r + n_rRK_c \leq \bar{R}$  и  $r + (n_r + 1)RK_c > \bar{R}$ .

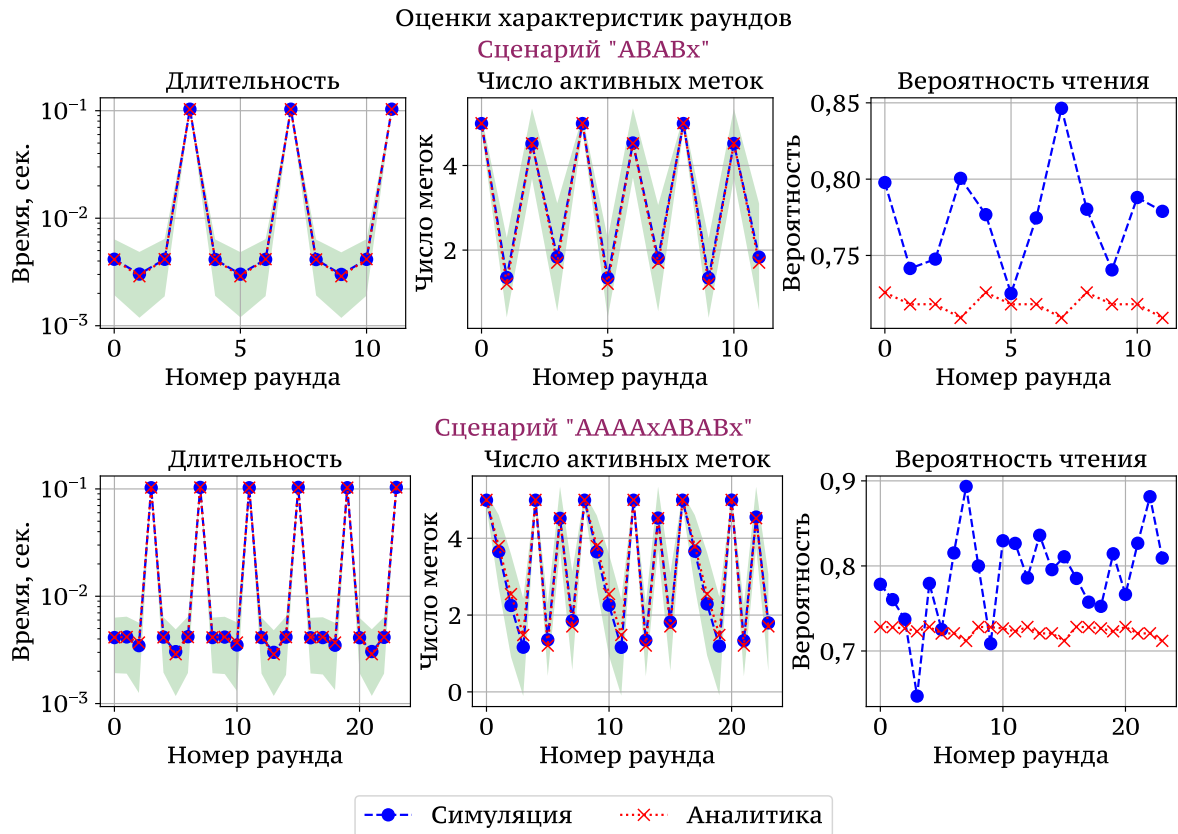


Рисунок 3.11 — Оценки характеристик раундов

Результаты сравнения оценок длительностей раундов, среднего числа активных меток и вероятностей идентификации (при поступлении в заданном раунде), полученных с помощью аналитической и имитационной модели для смешанных сценариев, приведены на рис. 3.11.

При построении результатов использовался множитель раундов при свертке  $K_c = 3$ . Можно видеть, что оценки длительности и среднего числа активных меток практически полностью совпадают (серая область — разброс оценок из имитационной модели). В то же время, оценки вероятности идентификации отличаются, хотя и незначительно.

Из полученных результатов можно сделать вывод, что фоновый процесс позволяет очень точно оценить длительность раундов и среднее число активных меток. Оценки вероятности идентификации, полученные с помощью аналитики, оказываются немного ниже, чем оценки, полученные из имитационной модели.

### 3.6.3 Расчет вероятности идентификации

Для каждого сценария были рассчитаны вероятности идентификации с помощью аналитической и имитационной моделей. Результаты сравнения приведены на рис. 3.12. Рассматривались три различных интервала между поступлениями меток: очень короткий (0,2 сек), средний (0,4 сек) и длинный (1 сек). Так как время нахождения каждой метки в области чтения  $T_L = 2,5$  сек фиксировано, то этим значениям соответствуют различные максимальные числа меток в области чтения  $\bar{N}$ , равные 13, 7 и 3 соответственно. За счет этого можно эффективно учесть влияние коллизий на вероятность идентификации (параметр  $Q$  был выбран равным 2, то есть в каждом раунде 4 слота).

Результаты, приведенные на рис. 3.12, показывают, что оценки оказываются во всех случаях близки, характеры зависимостей от BER совпадают, но оценки аналитической модели всегда несколько ниже. Отметим, что на практике при  $BER > 0,02$  вероятность успешно передать данные слишком мала, поэтому основной интерес представляют левые половины графиков, а на них отклонение оказывается совсем малым. Таким образом, при небольших значениях BER аналитическая модель позволяет достаточно точно оценить вероятность идентификации.

Чтобы более наглядно оценить влияние выбора сценария на вероятность идентификации, на рис. 3.13 показаны оценки, полученные с помощью аналитической модели для различных сценариев и интервалов. Самыми неэффективными оказываются сценарий  $A^0$  (без сбросов питания и смены флагов) и  $A^1$  (сброс питания после каждого раунда), причем первый из них дает практически идентичные результаты независимо от интервала, а второй оказывается тем менее эффективен, чем короче интервал. Низкую эффективность сценария  $A^0$  и слабую зависимость от длины интервалов между поступлениями можно объяснить тем, что после первой ошибки в передаче EPCID метка инвертирует флаг и больше не участвует в опросе. Так как новая метка успевает ошибиться (или успешно передать идентификатор) в течение первых раундов после поступления, к моменту появления следующей метки она уже неактивна, поэтому не влияет на коллизии. В сценарии  $A^1$ , напротив, в каждом раунде участвуют все метки, находящиеся в области чтения, поэтому при большом числе меток вероятность идентификации снижается из-за коллизий. Кроме того, на частые сбросы питания



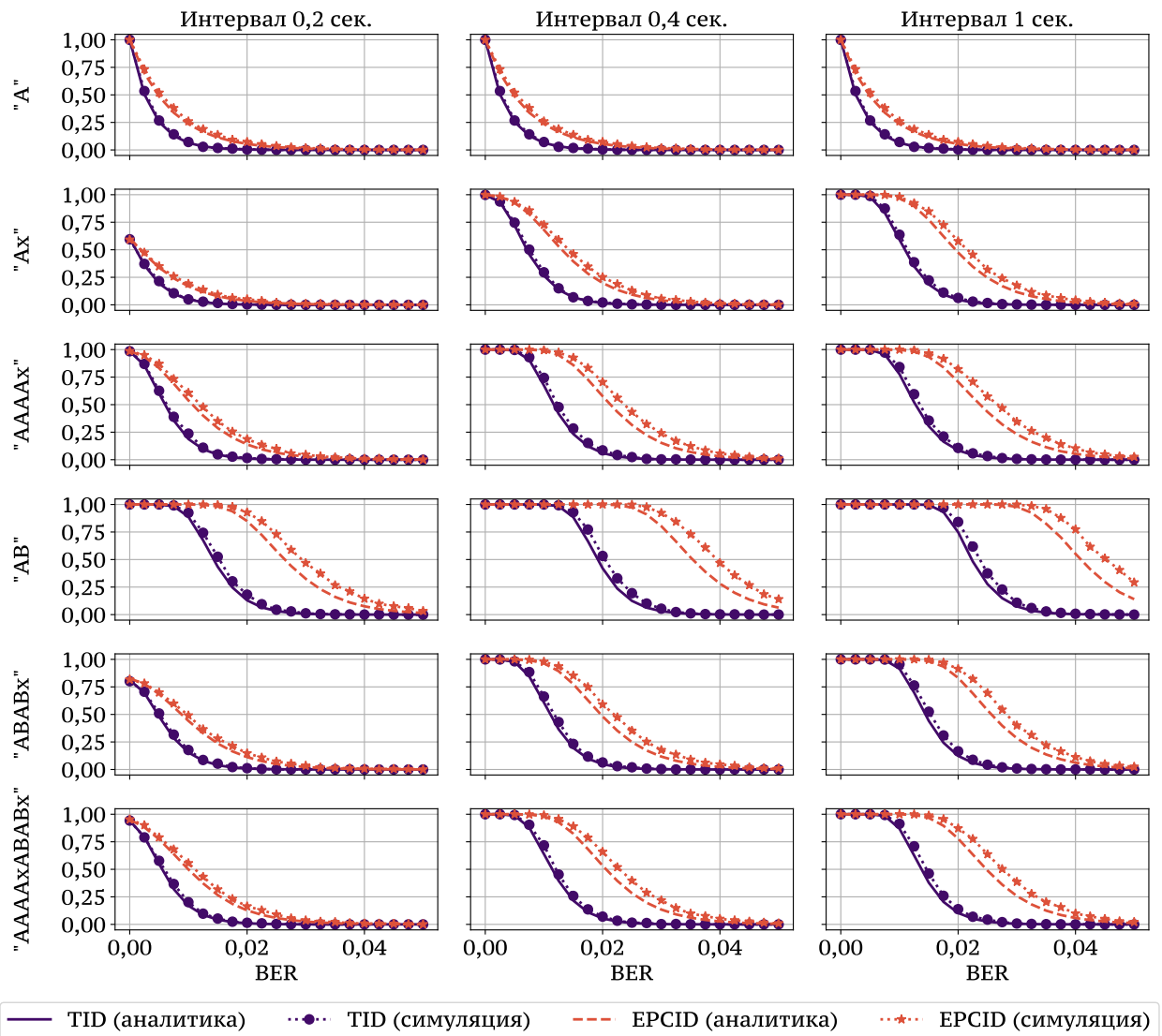


Рисунок 3.12 — Сравнение оценок вероятности идентификации, полученных с помощью аналитической и имитационной модели

расходуется очень много времени, из-за чего вероятность снижается и при небольшом числе меток. Примечательно, что при самых коротких интервалах вероятность идентификации в сценарии  $A^1$  оказывается даже ниже, чем в сценарии  $A^0$ .

Самые лучшие результаты показывают сценарии, в которых не используется сброс питания, а только периодические смены флагов. При этом, чем больше меток в области чтения, тем реже стоит делать переключения, и наоборот: при идентификации по TID и поступлении меток каждый 0,2 сек вероятность идентификации в сценарии  $A^0 A^0 A^0 A^0 B^0 B^0 B^0 B^0$  оказывается немного выше, чем в сценарии  $A^0 B^0$ , а при интервале 1 сек — наоборот. Этот эффект можно объяснить тем, что более редкие переключения позволяют более эффективно противостоять росту вероятности коллизии при увеличении числа меток в области чтения. Отметим, что при значениях BER до 0,01 различий практически нет.

Сценарии со сбросами питания дают вероятность идентификации существенно ниже, чем сценарии со сменой флага и без сброса питания. Примечательно, что при коротких интервалах выгоднее не менять флаг (сценарий  $A^0 A^0 A^0 A^1$ ), чем изменять его в дополнение к

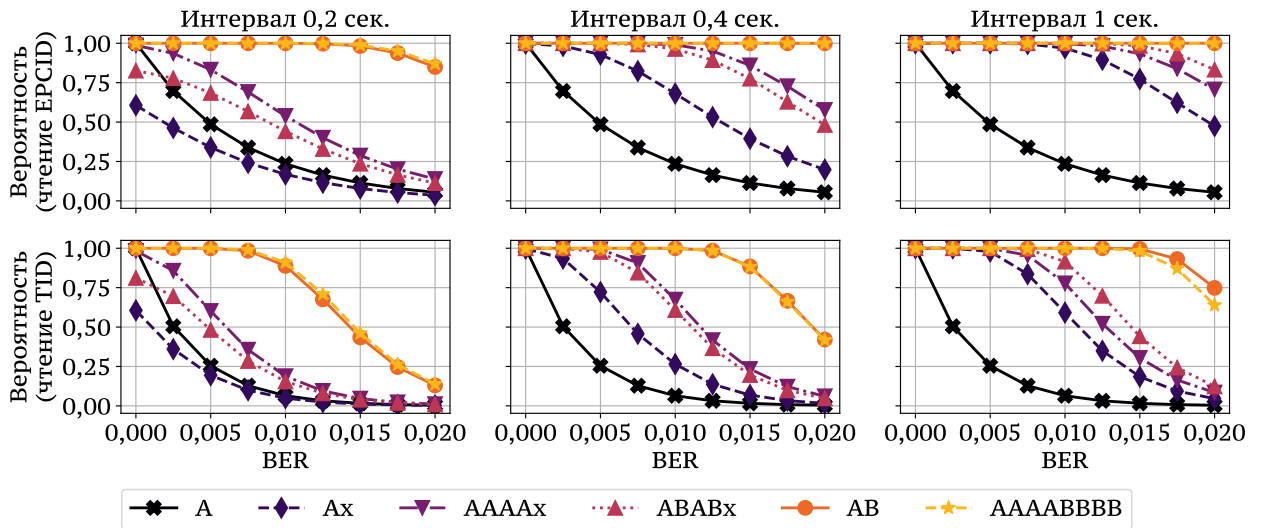


Рисунок 3.13 — Вероятности идентификации в различных сценариях

сбросу питания (сценарий  $A^0B^0A^0B^1$ ). Как и для сценариев со сменой флага без сброса питания, это можно объяснить тем, что смены флагов ведут к увеличению вероятности коллизии.

В целом, можно сделать вывод о том, что сбросов питания лучше избегать, если есть возможность периодически изменять флаги опроса. Если по какой-то причине возможности менять флаги нет, то желательно периодически сбрасывать питание во всех случаях, кроме присутствия большого числа меток. Отметим, что часто сбросы питания возникают независимо, из-за смены антенн или необходимости периодических отключений по требованиям регулирующих органов.

### 3.6.4 Исследование ошибок в оценке вероятности идентификации

Полагая результаты имитационного моделирования точными, вычислим относительную и абсолютную ошибки в определении вероятности идентификации. На рис. 3.14 в левой части показаны относительные ошибки для сценариев, в которых происходит только отключение питания, только смена флагов или и то, и другое (либо ни то, ни другое). Ошибки показаны с помощью свечного графика (график типа ящик с усами), на котором тело свечи ограничено первым и третьим квартилем, горизонтальная линия в свече показывает медиану, а границы усов взяты по 1-му и 99-му процентилем.

Чтобы лучше понять причины этого роста, на правой части графика изображена диаграмма рассеяния, точки которой соответствуют ошибкам, полученным для тех входных данных, на которых была вычислена вероятность идентификации, отмеченная на абсциссе. Из этих диаграмм можно видеть, что относительные ошибки увеличиваются при уменьшении вероятности идентификации, что может говорить о небольшой абсолютной ошибке, которая не уменьшается до нуля при уменьшении вероятности идентификации. Чтобы подтвердить это предположение, на рис. 3.15 показаны абсолютные ошибки. Можно видеть, что для боль-

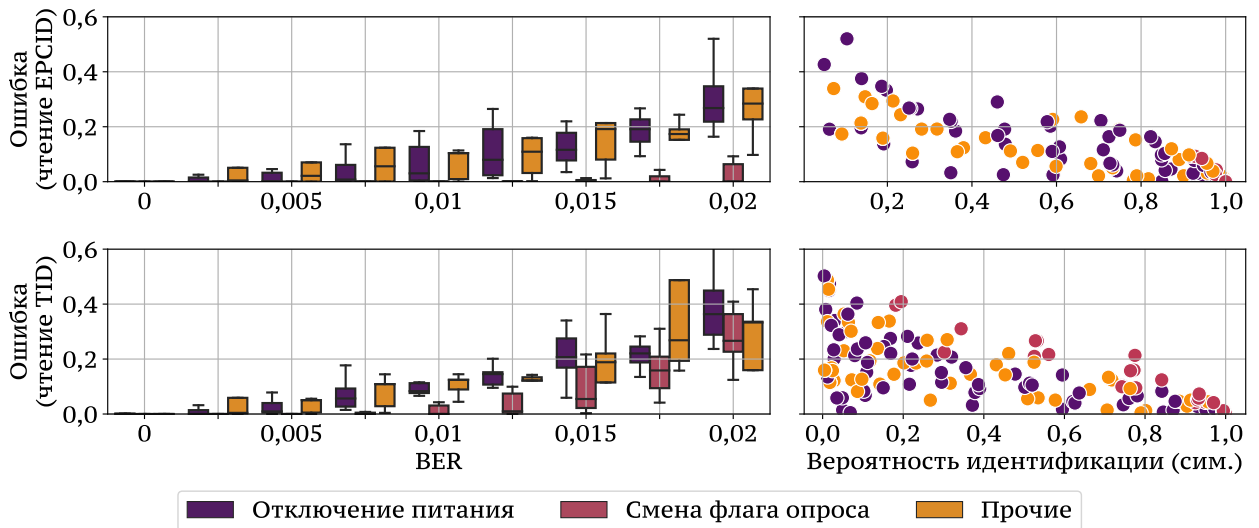


Рисунок 3.14 — Относительные ошибки оценки вероятности идентификации

шей части сценариев медианы ошибок не превосходят 0,05, и только при достаточно большом BER в некоторых сценариях приближаются к 0,1. Так как при росте BER вероятность идентификации уменьшается, то, соответственно, растет и относительная ошибка.

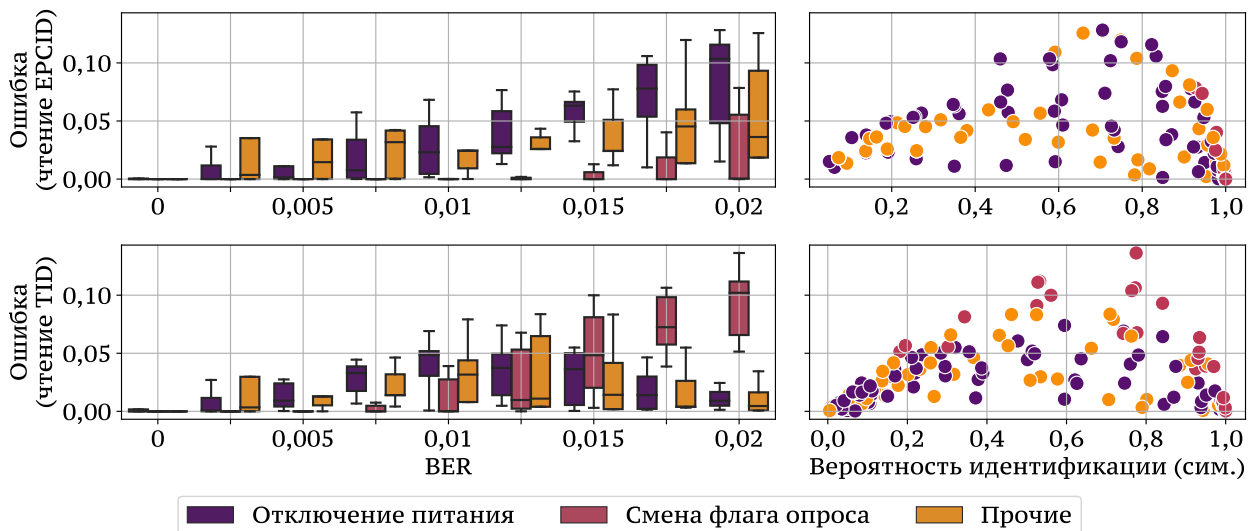


Рисунок 3.15 — Абсолютные ошибки оценки вероятности идентификации

Полученные результаты говорят о том, что точность аналитической модели оказывается достаточно высокой, для низких значениях BER  $< 0,01$  относительная ошибка мала: для идентификации по EPCID медианы не превосходят 5%, а для идентификации по TID — 10%. Для более высоких значений BER относительная ошибка растет до 0,4, но абсолютная ошибка остается достаточно малой, медианные значения не превышает 0,11.

### 3.7 Заключение

В главе были представлены следующие результаты.

1. Предложена новая аналитическая модель системы радиочастотной идентификации мобильных меток, позволяющая учитывать сбросы питания считывателем и смены флагов сессий. Модель позволяет находить оценки длительностей раундов и вероятности идентификации меток. Модель включает в себя два неоднородных марковских процесса, описывающих число участвующих в каждом раунде меток, и состояние отдельной метки, для которой вычисляется вероятность идентификации.
2. Приведены численные результаты, показывающие, что периодические смены флагов опросов значительно повышают вероятность идентификации движущейся метки. Сбросы питания также повышают вероятность, но менее эффективны.
3. Приведены результаты сравнения аналитической и имитационной моделей, подтверждающие высокую точность аналитической модели.

## Глава 4. Анализ производительности опорной беспроводной сети

Беспроводные сети часто используются для создания опорных сетей на больших расстояниях, особенно когда проводная сеть недоступна по тем или иным причинам. Для построения опорных сетей часто используются радиорелейное оборудование или радиомаршрутизаторы стандарта IEEE 802.11 (WiFi) или IEEE 802.16 (WiMax).

Одна из отличительных черт беспроводных сетей с множественным доступом — сложные методы доступа к каналу. В беспроводной сети с множественным доступом к каналу одновременно может быть подключено несколько станций, причем некоторые станции могут не слышать друг друга (проблема скрытых станций). Из-за этого возникают ситуации, когда две или более станций ведут одновременные передачи, которые искажаются на приемнике и возникают коллизии. Кроме того, передачи сигналов в беспроводных сетях значительно больше подвержены искажениям из-за многолучевого распространения сигнала, движения станций, изменений условий окружающей среды и других факторов. По этим причинам в беспроводных сетях с множественным доступом (в том числе, в сетях IEEE 802.11 и IEEE 802.16) используются гораздо более сложные схемы доступа к каналу, по сравнению с проводными или радиорелейными сетями.

В распределенной системе радиочастотной идентификации транспорта RFID-считыватели подключаются к центрам обработки данных, поэтому для оценки общей эффективности системы нужно иметь оценки задержек и потерь в сети. Имея эти оценки, можно определить, например, число считывателей, которые могут быть одновременно подключены, чтобы система работала без перегрузок, или сколько маршрутизаторов может быть в сети, чтобы задержка оставалась в заданных пределах.

В диссертационном исследовании для оценки характеристик многошаговых беспроводных сетей будут использоваться модели тандемных сетей массового обслуживания с узлами MAP/PN/1/N. PN-распределения времени обслуживания будут строиться методом моментов по значениям, полученным из имитационной модели беспроводного канала. Для оценки ошибок, в настоящей главе приведены результаты численного эксперимента, в котором сравнивались значения межконцевых задержек, полученные с помощью имитационного моделирования беспроводной сети и модели тандемной сети массового обслуживания.

Рассчитать характеристики тандемной сети массового обслуживания MAP/PN/1/N аналитически оказывается не всегда возможно из-за экспоненциальной зависимости размера задачи от числа станций. Для поиска численных характеристик предлагается итерационно заменять MAP-потoki обслуженных пакетов потоками меньшей размерности. Приведенные в настоящей главе численные результаты показывают, что предложенный метод позволяет получить результаты, близкие к методу Монте-Карло, и при этом требуют меньшего времени для вычислений.

Тема и результаты, представленные в главе, были опубликованы в журналах [194–199], а также в работах, индексируемых Scopus/WoS [200–205].

#### 4.1 Моделирование многошаговой беспроводной сети с помощью тандемной сети массового обслуживания

Рассмотрим многошаговую беспроводную сеть, которая передаст информацию от RFID-считывателя или видеокамеры в центр обработки данных. При исследовании производительности этой сети методами теории массового обслуживания для моделирования каналов связи используются случайные задержки (обслуживающие приборы), для маршрутизаторов — очереди ограниченной или бесконечной емкости, а для источников данных — случайные входящие потоки (см. рис. 4.1). Для определения открытой сети массового обслуживания нужно задать распределение интервалов  $A_i$  между пакетами, поступающими в сеть, распределения длительностей обслуживания пакетов на каждой  $k$ -й станции  $B_{k,i}$ , а также дисциплину обслуживания и емкость очередей, если они полагаются конечными.

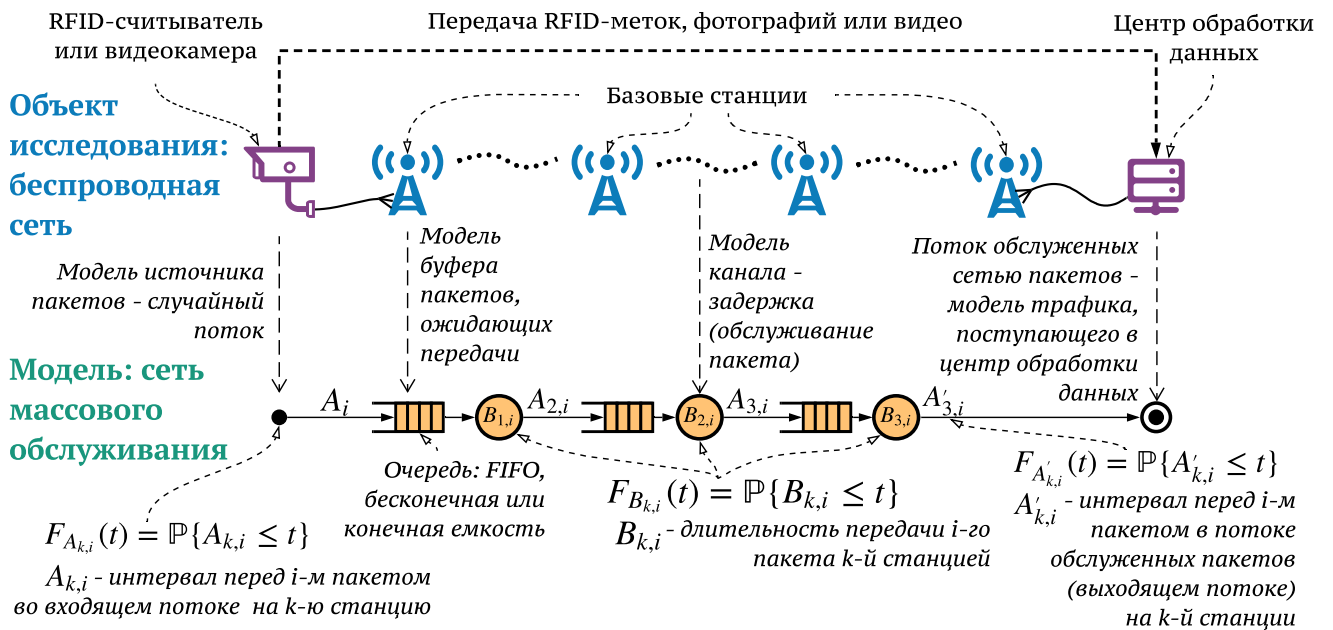


Рисунок 4.1 — Беспроводная сеть с линейной топологией и ее аналитическая модель

Для моделирования времени обслуживания будут использоваться PH-распределения  $B_i \sim PH(S, \bar{\tau})$ , а для моделирования интервалов между поступлениями в сеть пакетов — MAP-поток  $A \sim MAP(D_0, D_1)$ . Определения PH-распределений и MAP-поточков были даны в главе 1, а свойства, необходимые для вычисления характеристик тандемной сети массового обслуживания, будут приведены в следующем разделе.

Поставим формально задачу вычисления средней межконцевой задержки и вероятности потери пакетов из-за переполнения очередей в сети массового обслуживания. Пусть заданы длина сети (число узлов)  $N$ , емкость очередей  $M$ , входящий MAP-поток  $X \sim MAP(D_0, D_1)$  и PH-распределения времени обслуживания  $Y_k \sim PH(S_k, \bar{\tau}_k)$ ,  $k = 1, 2, \dots, N$ . Будем обозначать момент поступления  $i$ -го пакета на вход  $k$ -го узла как  $t_{i,k}^{(a)}$ , а момент завершения его обслуживания — как  $t_{i,k}^{(d)}$ . Очевидно, что для любого  $k < N$  выполняется  $t_{i,k}^{(d)} = t_{i,k+1}^{(a)}$ , если пакет не застает очередь  $(k + 1)$ -го узла заполненной (в этом случае пакет теряется).

Рассмотрим произвольный  $i$ -й пакет. Результатом обработки этого пакета может быть один из двух исходов: либо на некотором  $k$ -м узле пакет застанет очередь заполненной и будет отброшен, либо он будет обработан последовательно каждым из узлов, и покинет систему тогда, когда завершится его обработка последним узлом в момент  $t_{i,N}^{(d)}$ . Вероятность первого исхода (то есть потери пакета) будем обозначать  $P_l$ , она представляет собой долю пакетов, заставших какой-либо узел сети полностью заполненным. Для пакетов, которые не были потеряны, можно вычислить межконцевую задержку  $\Delta t_i = t_{i,N}^{(d)} - t_{i,1}^{(a)}$ .

Задача исследования состоит в поиске методов численной оценки средней межконцевой задержки  $\overline{\Delta t}$  и вероятности потери пакета  $P_l$ .

Из-за экспоненциального роста размера пространства состояний при увеличении числа станций в сети, вычисление точных значений задержек и вероятностей потери пакетов возможно только для небольших сетей. Для получения оценок параметров в общем случае предлагается использовать метод редукции выходящих потоков, то есть заменять потоки обслуженных пакетов на выходе из очередного узла сети МАР-поток небольшого размера. Этот метод будет подробно описан и численно исследован в этой главе диссертационной работы.

## 4.2 Открытая тандемная сеть массового обслуживания с узлами МАР/РН/1/М

В этом разделе рассмотрим основные свойства РН-распределений, МАР-потоков и систем массового обслуживания МАР/РН/1/М. На базе этих свойств, сформулируем итерационный алгоритм, позволяющий рассчитывать характеристики сетей массового обслуживания типа МАР/РН/1/М  $\rightarrow \bullet/\text{РН}/1/\text{М} \rightarrow \dots \rightarrow \bullet/\text{РН}/1/\text{М}$ , и приведем анализ его сложности.

### 4.2.1 Свойства РН-распределений и МАР-потоков

Пусть  $Y \sim PH(S, \bar{\tau})$ , матрица  $S$  и вектор  $\bar{\tau}$  удовлетворяют ограничениям (1.1). Тогда функция распределения  $F(y)$  и моменты случайной величины  $Y$  определяются как [112]:

$$\begin{aligned} F(y) &= \mathbb{P}\{Y < y\} = 1 - \bar{\tau} e^{Sy} \mathbf{1} \\ MY^k &= k! \bar{\tau} (-S)^{-k} \mathbf{1} \end{aligned} \quad (4.1)$$

Пусть  $X \sim MAP(D_0, D_1)$ , где матрицы  $D_0$  и  $D_1$  удовлетворяют ограничениям (1.2). Матрица  $D_1$  описывает переходы управляющей марковской цепи, сопровождающиеся генерацией пакетов (наблюдаемые переходы), а матрица  $D_0$  — переходы, при которых генерации

нового пакета не происходит (невидимые переходы). Если после генерации очередного пакета МАР-поток находится в состоянии  $i$ , время до появления следующего пакета имеет РН-распределение  $X^{(i)} = PH(D_0, \bar{e}_i)$ , где  $\bar{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$  — единичный вектор, в котором на  $i$ -й позиции стоит единица. Число состояний  $W$  в управляющей цепи потока будем называть его размером или порядком и обозначать как  $|X|$ .

Функция распределения интервалов между событиями, значения моментов МАР-потока  $X$  и коэффициенты корреляции с лагом  $k$  определяются выражениям:

$$\begin{aligned} \mathbb{P}\{X < t\} &= 1 - \bar{\alpha} e^{D_0 t} \bar{\mathbf{1}} \\ \mathbb{M}X^k &= k! \bar{\alpha} (-D_0)^{-k} \bar{\mathbf{1}} \\ \rho_k &= \frac{\lambda \bar{\pi} P^k (-D_0)^{-1} \bar{\mathbf{1}} - 1}{2\bar{\pi} (-D_0)^{-1} \bar{\mathbf{1}} - 1}. \end{aligned} \quad (4.2)$$

Здесь  $P = (-D_0)^{-1} D_1$  — матрица вложенной цепи. Эта матрица  $P$  и вектор стационарных вероятностей  $\bar{\alpha}$  рассчитываются с помощью системы линейных алгебраических уравнений:

$$\bar{\alpha} : \begin{cases} \bar{\alpha} P = \bar{\alpha} \\ \sum_{i=1}^W \alpha_i = 1 \end{cases} \quad (4.3)$$

Стационарное распределение  $\bar{\pi} \in \mathbb{R}^W$  вероятностей потока  $MAP(D_0, D_1)$  является решением системы линейных уравнений:

$$\begin{cases} \bar{\pi} (D_0 + D_1) = \bar{\mathbf{0}} \\ \bar{\pi} \bar{\mathbf{1}} = 1 \end{cases}, \quad (4.4)$$

где  $\bar{\mathbf{0}}$  и  $\bar{\mathbf{1}}$  — вектор-строки, состоящие из всех нулей и единиц соответственно.

Интенсивность МАР-потока можно найти как  $\lambda = (\mathbb{M}X)^{-1}$ . Ее можно вычислить как математическое ожидание случайной величины, равной суммарной интенсивности наблюдаемых переходов:

$$\lambda = \sum_{j=1}^W \pi_j \sum_{k=1}^W \{D_1\}_{jk} = \bar{\pi} D_1 \bar{\mathbf{1}} \quad (4.5)$$

Формула (4.5) удобна, когда не требуется вычисление моментов старших порядков или коэффициентов корреляции, т.к. в этом случае также не требуется обращать матрицу  $D_0$ .

МАР-поток является обобщением над РН-распределением. Существенной чертой МАР-потока является то, что корреляция между интервалами может быть отличной от нуля (см. формулу (4.2)). В то же время, последовательность интервалов, каждый из которых имеет РН-распределение с одинаковыми параметрами  $PH(S, \bar{\tau})$ , также можно считать МАР-поток, корреляция в котором будет нулевой. Для РН-распределения  $PH(S, \bar{\tau})$  соответствующий МАР-поток будет иметь матрицы следующего вида:



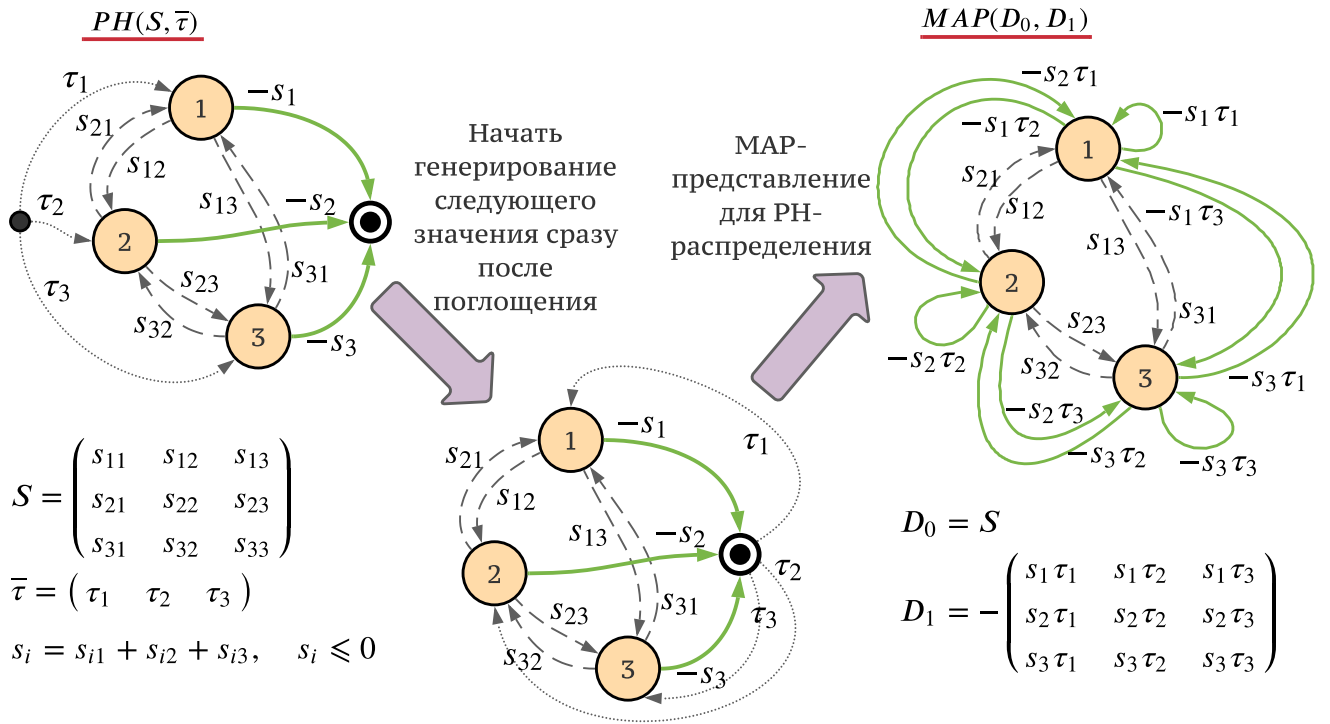


Рисунок 4.2 — Пример представления PH-распределения в виде МАР-потока

$$D_0 = S, \quad D_1 = (-S\bar{\tau})\bar{\tau} = - \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_V \end{pmatrix} (\tau_1 \quad \tau_2 \quad \dots), \quad (4.6)$$

где  $s_i = \sum_{j=1}^V s_{ij}$  Пример построения МАР-потока по PH-распределению приведен на рис. 4.2.

#### 4.2.2 Свойства системы МАР/PH/1/М

Ключевое свойство систем массового обслуживания МАР/PH/1/М, благодаря которому их удобно использовать для моделирования многошаговых сетей — замкнутость на множестве МАР-потоков: согласно следующим теоремам (см. [109]), результат просеивания МАР-потока — МАР-поток, суперпозиция МАР-потоков — МАР-поток, и поток обслуженных заявок, выходящих из системы МАР/PH/1/М, также является также МАР-потоком. В дальнейшем будем обозначать выходной поток  $Z$  из системы массового обслуживания с входным потоком  $X$ , временем обслуживания  $Y$  и емкостью очереди  $M$  как  $\mathcal{D}(X, Y, M)$ .

**Теорема 4.1.** [109] *Результат просеивания МАР-потока  $X \sim \text{МАР}(D_0, D_1)$  с вероятностью  $p$  — МАР-поток  $X_p \sim \text{МАР}(D_0 + (1 - p)D_1, pD_1)$  (в дальнейшем обозначается как  $pX$ )*

**Теорема 4.2.** [109] *Суперпозиция МАР-потоков  $X_1$  и  $X_2$ ,  $X_i \sim \text{МАР}(D_0^{(i)}, D_1^{(i)})$ ,  $i = 1, 2$  – МАР-поток  $X$ :*

$$X = X_1 \oplus X_2 \sim \text{МАР}(D_0^{(1)} \oplus D_0^{(2)}, D_1^{(1)} \oplus D_1^{(2)}),$$

где  $\oplus$  – сумма Кронекера.

*Замечание 4.1.* Если потоки  $X_1$  и  $X_2$  имеют размерности  $W_1$  и  $W_2$ , то размерность суммарного потока  $X$  равняется  $W = W_1 W_2$ .

**Теорема 4.3.** [109] *Пусть в системе МАР/РН/1/М  $X \sim \text{МАР}(D_0, D_1)$ ,  $D_i \in \mathbb{R}^{W \times W}$ ,  $i = 0, 1$  – входящий поток,  $Y \sim \text{РН}(S, \bar{\tau})$ ,  $\mathbb{R}^{V \times V}$  – время обслуживания, обслуживание ведется согласно дисциплине FIFO, а емкость очереди равна  $M$ . Тогда поток выходящих (обслуженных) пакетов есть МАР-поток  $Z \sim \text{МАР}(D'_0, D'_1)$ , матрицы которого определяются как:*

$$D'_0 = \begin{bmatrix} D_0 \otimes I_V & D_1 \otimes (\bar{\tau} \otimes \bar{I}_V) & 0 & \cdots & 0 & 0 \\ 0 & D_0 \otimes S & D_1 \otimes I_V & \cdots & 0 & 0 \\ 0 & 0 & D_0 \otimes S & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & D_0 \otimes S & D_1 \otimes I_V \\ 0 & 0 & 0 & \cdots & 0 & (D_0 + D_1) \otimes S \end{bmatrix}, \quad (4.7)$$

$$D'_1 = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ I_W \otimes C_t & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & I_W \otimes C_t & 0 \end{bmatrix}, \quad (4.8)$$

где  $C_t = (-S\bar{I}_V) \otimes \bar{\tau}$ , а  $I_V, I_W$  – единичные матрицы порядка  $V$  и  $W$  соответственно.

Из структуры матриц потока  $Z$  можно сделать следующие выводы. Во-первых, поток  $Z$  имеет размерность  $|Z| = (M+2)|X||Y| = (M+2)VW$ . Во-вторых, каждому состоянию потока  $Z$  соответствует некоторое число пакетов в системе и состояния входящего МАР-потока  $X$  и обслуживающего прибора  $Y$ . Пусть управляющая цепь потока  $Z$  находится в состоянии  $n$ . Тогда в системе  $\lfloor \frac{n}{VW} \rfloor$  пакетов, входящий МАР-поток находится в состоянии  $\lfloor \frac{n(\text{mod } VW)}{V} \rfloor + 1$ , а обслуживающий прибор – в состоянии  $n(\text{mod } V) + 1$ .

Предыдущее замечание можно обобщить. По сути, при построении матриц выходящего МАР-потока производится построение марковской цепи, управляющей работой системы МАР/РН/1/М. Действительно, в такой системе переходы происходят из-за возникновения одного из двух событий: переход в цепи входящего МАР-потока или переход в цепи РН-распределения. При этом, если переход в цепи входящего потока сопровождается формированием сообщения, и система не находится в состоянии полностью заполненной очереди, то размер системы увеличивается. Если же переход происходит в цепи РН-распределения в поглощающее состояние, то размер системы уменьшается на единицу. Элементы матрицы  $D'_0$  соответствуют всем переходам, кроме поглощения в цепи РН-распределения, последним же соответствуют элементы матрицы  $D'_1$ . По этой причине, когда будет требоваться найти

распределение вероятностей управляющей цепи системы МАР/РН/1/М, будем использовать инфинитезимальный генератор управляющей цепи выходящего потока  $Z$ .

Обозначим  $\bar{\pi}$  стационарное распределение  $X$ , а  $\bar{\theta}$  стационарное распределение  $Z$ . Вектор  $\bar{\theta}$  можно переписать в виде  $\bar{\theta} = (\bar{\theta}_0, \bar{\theta}_1, \dots, \bar{\theta}_{M+1})$ , где  $\bar{\theta}_k$  — часть вектора  $\bar{\theta}$ , соответствующая тем состояниям системы, в которых число пакетов равно  $k$ . Пусть  $\mu(t)$  — случайный процесс, значение которого в каждый момент времени равняется числу пакетов в системе. Обозначим  $p_k = \lim_{t \rightarrow \infty} \mathbb{P}\{\mu(t) = k\}$  — стационарная вероятность того, что в системе  $k = \overline{0, M+1}$  пакетов. Тогда, зная стационарное распределение  $\bar{\theta}$  управляющей цепи МАР-потока  $Z$ , можно вычислить распределение  $\bar{p}$ :

$$p_k = \sum_{j=1}^{VW} \{\bar{\theta}_k\}_j. \quad (4.9)$$

Отсюда можно рассчитать среднее число пакетов в системе как  $m_1 = \mathbb{M}\mu$ :

$$m_1 = \sum_{k=0}^{M+1} k \mathbb{P}\{\mu = k\} = \sum_{k=0}^{M+1} k p_k = \sum_{k=0}^{M+1} k \sum_{j=1}^{VW} \{\bar{\theta}_k\}_j = \sum_{k=0}^{M+1} \sum_{j=1}^{VW} k \theta_{kVW+j}. \quad (4.10)$$

Для получения вероятности потери пакета из-за переполнения очереди нужно более подробно рассмотреть часть вектора  $\bar{\theta}$ , соответствующую случаю заполненной очереди (т.е.  $\mu = M+1$ ). Состояния внутри блоков матриц (4.7) и (4.8) сгруппированы по состояниям входящего МАР-потока, т.е. состояния  $\{\bar{\theta}_k\}_{jV+1}, \dots, \{\bar{\theta}_k\}_{(j+1)V}$  соответствуют всевозможным состояниям цепи РН-распределения и состоянию  $j$  входящего МАР-потока, когда в системе  $k$  пакетов. Определим:

$$\bar{\Psi}_k = \left( \sum_{j=1}^V \{\bar{\theta}_k\}_j, \dots, \sum_{j=1}^V \{\bar{\theta}_k\}_{(W-1)V+j} \right) \quad (4.11)$$

— распределение вероятностей входящего МАР-потока при  $k$  пакетах в системе. Тогда вероятность  $P_L$  потери пакета можно выразить так:

$$P_L = \bar{\Psi}_{M+1} \frac{D_1 \bar{\mathbf{1}}}{\lambda_X} \quad (4.12)$$

Наконец, зная интенсивность входящего потока  $\lambda_X$ , среднее число пакетов в системе  $m_1$  и вероятность потери пакетов  $P_L$ , используя формулу Литтла, можно рассчитать среднюю задержку:

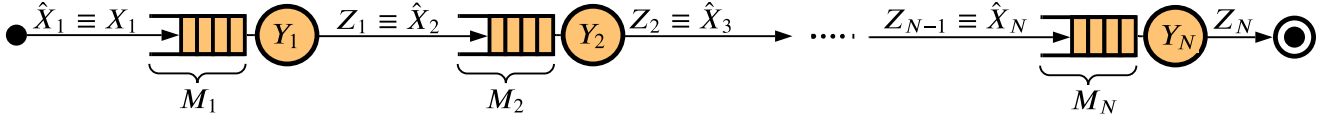
$$T = \frac{m_1}{(1 - P_L)\lambda_X}. \quad (4.13)$$

Множитель  $(1 - P_L)$  в знаменателе возникает в следствие того, что из-за переполнения очереди лишь часть пакетов входящего потока попадает в систему, т.е. фактическая интенсивность потока пакетов, поступающих в систему, составляет  $(1 - P_L)\lambda_X$ .

### 4.2.3 Точный расчет характеристик сети массового обслуживания

Используя формулы (4.10), (4.12) и (4.13) несложно вычислить среднюю межконцевую задержку пакетов в сети и вероятность потери пакета на каком либо узле. Будем рассматривать два варианта сетей: сети с кросс-трафиком, в которых потоки данных поступают на каждый узел, и сети без кросс-трафика, в которых внешний трафик поступает только на первую станцию, и все последующие станции передают только его.

Сеть без кросс-трафика:



Сеть с кросс-трафиком:

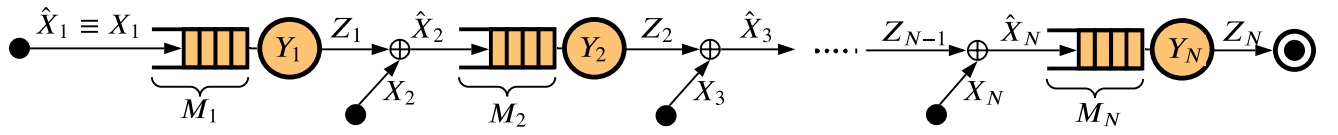


Рисунок 4.3 — Сети массового обслуживания с кросс-трафиком и без него

Пусть в сети (см. рис. 4.3)  $N$  станций, емкость очереди на  $i$ -й станции ( $i = 1, 2, \dots, N$ ) равна  $M_i \in \mathbb{N}$ , а длительность обслуживания  $Y_i$  имеет PH-распределение  $Y_i \sim PH(S_i, \bar{\tau})$ ,  $S_i \in \mathbb{R}^{V_i \times V_i}$ ,  $V_i \in \mathbb{N}$ . Если в сети есть кросс-трафик, то на каждую станцию поступает пользовательский поток  $X_i \sim MAP(D_{0,i}, D_{1,i})$ ; если кросс-трафика нет, то поток  $X_1 \equiv X = MAP(D_0, D_1)$  поступает только на первую станцию. Обозначим порядок входящего MAP-потока на  $i$ -ю станцию как  $W_i$ , то есть  $D_{0,i}, D_{1,i} \in \mathbb{R}^{W_i \times W_i}$ ,  $W_i \in \mathbb{N}$ .

Обозначим  $Z_i$  - выходящий MAP-поток с  $i$ -й станции, а  $\hat{X}_i$  - общий входящий поток на  $i$ -ю станцию. Тогда, согласно теоремам 4.2 и 4.3, потоки  $\hat{X}_i$  и  $Z_i$  — MAP-потоки. Используя ранее введенные обозначения, их можно определить формально следующим образом:

$$\begin{aligned} \hat{X}_1 &\equiv X_1 \\ \hat{X}_i &= \begin{cases} X_i \oplus Z_{i-1}, & \text{в сети есть кросс-трафик и } i = 2, 3, \dots, N \\ Z_{i-1}, & \text{в сети нет кросс-трафика} \end{cases} \\ Z_i &= \mathcal{D}(\hat{X}_i, Y_i, M) \end{aligned} \quad (4.14)$$

Обозначим порядок выходящих MAP-потоков  $Z_i$  как  $U_i$ ,  $i = 1, 2, \dots, N$ , то есть  $Z_i \sim MAP(\tilde{D}_{i,0}, \tilde{D}_{i,1})$  и  $\tilde{D}_{i,0}, \tilde{D}_{i,1} \in \mathbb{R}^{U_i \times U_i}$ . Тогда величина  $U_i$  определяется с помощью следующего утверждения.

**Утверждение 4.1.** Порядок  $U_i$  MAP-потока обслуженных пакетов  $Z_i$  на выходе из  $i$ -го узла,  $i = 1, 2, \dots, N$ , определяется как:

$$U_i = \begin{cases} \prod_{j=1}^i (M_j + 2) V_j W_j, & \text{если в сети есть кросс-трафик} \\ W_1 \prod_{j=1}^i (M_j + 2) V_j, & \text{если кросс-трафика нет.} \end{cases}$$

*Доказательство.* Утверждение доказывается по индукции. При  $i = 1$   $Z_1 = \mathcal{D}(X_1, Y_1, M_1)$  и  $U_1 = (M_1 + 2)V_1W_1$  согласно замечанию к теореме 4.3.

Пусть утверждение верно при  $i = n - 1$ . Если в сети нет кросс-трафика, то  $Z_i = \mathcal{D}(\hat{X}_i, Y_i, M_i) = \mathcal{D}(Z_{i-1}, Y_i, M_i)$  и величина  $U_i$  определяется как:

$$\begin{aligned} U_i &= (M_i + 2)V_iU_{i-1} = (M_i + 2)V_i \times (W_1 \prod_{j=1}^{i-1} (M_j + 2)V_j) \\ &= W_1 \prod_{j=1}^i (M_j + 2)V_j. \end{aligned}$$

Если же в сети есть кросс-трафик, то

$$Z_i = \mathcal{D}(\hat{X}_i, Y_i, M_i) = \mathcal{D}(Z_{i-1} \oplus X_i, Y_i, M_i),$$

и порядок потока  $Z_i$  определяется следующей цепочкой равенств:

$$\begin{aligned} U_i &= (M_i + 2)V_i(U_{i-1}W_i) = (M_i + 2)V_iW_i \times \prod_{j=1}^{i-1} (M_j + 2)V_jW_j \\ &= \prod_{j=1}^i (M_j + 2)V_jW_j. \end{aligned}$$

■

Схема расчета характеристик сети выглядит следующим образом.

*Шаг 1.* Положим  $i := 1$ .

*Шаг 2.* Если  $i = 1$ , то положим  $\hat{X}_i = X_1$ . Если же  $i > 1$ , то вычисляем  $\hat{X}_i$  согласно (4.14):  $\hat{X}_i = Z_{i-1}$ , если в сети нет кросс-трафика, и  $\hat{X}_i = Z_{i-1} \oplus X_i$  иначе. Обозначим матрицы потока  $\hat{X}_i$  как  $\hat{D}_{i,0}$  и  $\hat{D}_{i,1}$ , то есть  $\hat{X}_i = \text{MAP}(\hat{D}_{i,0}, \hat{D}_{i,1})$ .

*Шаг 3.* С помощью теоремы 4.3 вычисляем матрицы  $D'_{i,0}$ ,  $D'_{i,1}$  MAP-потока  $Z_i = \mathcal{D}(\hat{X}_i, Y_i, M_i)$ .

*Шаг 4.* Для выходящего MAP-потока  $Z_i$  вычисляем его стационарное распределение  $\bar{\theta}^{(i)}$  с помощью системы линейных алгебраических уравнений:

$$\begin{cases} \bar{\theta}^{(i)}(D'_{i,0} + D'_{i,1}) &= 0 \\ \bar{\theta}^{(i)}\mathbf{1} &= 1 \end{cases}$$

*Шаг 5.* Рассчитываем среднее число пакетов в очереди  $i$ -й станции согласно (4.10):

$$m_1^{(i)} = \sum_{k=0}^{M_i+1} k \sum_{j=1}^{V_i \hat{W}_i} \theta_{kV_i \hat{W}_i + j}^{(i)},$$

где  $V_i = |Y_i|$  — порядок РН-распределения  $Y_i$ , а  $\hat{W}_i = |\hat{X}_i|$  — порядок входящего MAP-потока  $\hat{X}_i$ .

*Шаг 6.* Определяем стационарное распределение вероятностей  $\bar{\pi}^{(i)}$  входящего потока  $\hat{X}_i$ . Если в сети нет кросс-трафика и  $i > 1$ , то полагаем  $\bar{\pi}^{(i)} \equiv \bar{\theta}^{(i-1)}$ . В противном случае находим  $\bar{\pi}^{(i)}$  как решение системы линейных алгебраических уравнений:

$$\begin{cases} \bar{\pi}^{(i)}(\hat{D}_{i,0} + \hat{D}_{i,1}) &= 0 \\ \bar{\pi}^{(i)}\mathbf{1} &= 1 \end{cases}$$

*Шаг 7.* С помощью найденного на предыдущем шаге стационарного распределения  $\bar{\pi}^{(i)}$  входящего потока  $\hat{X}_i$  и формулы (4.5) вычисляем интенсивность поступления пакетов на  $i$ -ю станцию:

$$\lambda_i = \bar{\pi}^{(i)}\hat{D}_{i,1}\bar{\mathbf{1}}.$$

*Шаг 8.* Рассчитываем распределение состояний входящего МАР-потока при наличии в системе  $M_i + 1$  пакета (то есть при заполненной системе):

$$\bar{\Psi}^{(i)} = \left( \sum_{j=1}^{V_i} \{\bar{\theta}_{M_i+1}^{(i)}\}_j, \dots, \sum_{j=1}^{V_i} \{\bar{\theta}_{M_i+1}^{(i)}\}_{(W_{i-1})V_i+j} \right).$$

Здесь вектор  $\bar{\theta}_{M_i+1}^{(i)}$  — часть вектора  $\bar{\theta}^{(i)}$ , относящаяся к состояниям системы, когда в ней находится  $M_i + 1$  пакет.

*Шаг 9.* Вычисляем вероятность потери пакета из-за переполнения  $i$ -й очереди с помощью (4.12):

$$P_L^{(i)} = \bar{\Psi}^{(i)} \frac{\hat{D}_{i,0}}{\lambda_i} \bar{\mathbf{1}}.$$

*Шаг 10.* Вычисляем среднюю задержку пакетов на  $i$ -й станции с помощью (4.13):

$$T_i = \frac{m_1^{(i)}}{(1 - P_L^{(i)})\lambda_i}$$

*Шаг 11.* Если  $i < N$ , то увеличиваем  $i := i + 1$  и переходим на шаг 2. В противном случае переходим далее, на шаг 12.

*Шаг 12.* Вычисляем вероятность потери пакета  $P_L = \prod_{i=1}^N (1 - P_L^{(i)})$ .

*Шаг 13.* Вычисляем общую задержку  $T = \sum_{i=1}^N T_i$ .

Предложенная схема проста в вычислении. По сути, на каждом шаге с помощью нескольких операций произведения Кронекера строятся блочные матрицы для выходящего МАР-потока, а также, если в сети есть кросс-трафик, с помощью суммы Кронекера строятся матрицы входящего потока. Далее решаются две (если в сети есть кросс-трафик) или одна (в противном случае) системы линейных алгебраических уравнений для определения стационарных вероятностей входящего и исходящего потока. Наконец, с помощью нескольких операций умножений найденных распределений на матрицы потоков, вычисляются искомые характеристики — вероятность потери пакета, средний размер системы и задержка.

Главный недостаток этой схемы расчета — чрезвычайно высокая вычислительная сложность.

**Утверждение 4.2.** Пусть входящие МАР-потоки имеют порядок  $W$ , РН-распределения — порядок  $V$ , емкость очередей равна  $M$ , и сеть содержит  $N$  станций. Тогда итерационная схема расчета характеристик тандемной сети имеет сложность:

- $O((MVW)^{3N})$ , если в сети есть кросс-трафик;
- $O(W^3(MV)^{3N})$ , если кросс-трафика в сети нет.

*Доказательство.* Рассмотрим  $i$ -ю итерацию алгоритма,  $i \leq N$ , то есть расчет характеристик  $i$ -й станции сети. Отметим сперва, что при  $i > 1$  порядок выходящего потока с предыдущей  $i-1$ -й станции есть  $(M+2)V\hat{W}_i$ , где  $\hat{W}_i$  — порядок входящего потока на  $i$ -ю станцию. Согласно 4.1, при наличии в сети кросс трафика  $U_i = ((M+2)VW)^i$ , а если кросс-трафика нет, то  $U_i = W((M+2)V)^i$ .

Сложность итерации определяется шагами 4 и 6, в которых необходимо решать системы линейных алгебраических уравнений, причем порядок матрицы системы на шаге 4 (генератор выходящего потока) заведомо выше, чем системы на шаге 6 (генератор входящего потока). Полагая, что для решения системы используется алгоритм наподобии метода Гаусса, на шаге 4 потребуется  $O(U_i^3)$  операций. Остальные шаги имеют более низкую сложность: шаги 1, 10 и 11 —  $O(1)$ , шаг 2 —  $O(U_{i-1}^2 W^2)$ , шаг 3 —  $O(U_i^2)$ , шаг 5 —  $O(VW + M)$ , шаги 7 и 9 —  $O(U_i^2)$ , шаг 8 —  $O(VM)$ . Сложность шагов 12 и 13 есть  $O(N)$ .

Таким образом, если в сети есть кросс-трафик, сложность алгоритма составит

$$\begin{aligned} O((VWM)^3) + O((VWM)^6) + \dots + O((VWM)^{3N}) + O(N) \\ = O(VWM)^{3N}, \end{aligned}$$

а если кросс-трафика в сети нет, то

$$\begin{aligned} O(W^3(VM)^3) + O(W^3(VM)^6) + \dots + O(W^3(VM)^{3N}) + O(N) \\ = O(W^3(VM)^{3N}). \end{aligned}$$

■

**Замечание 1.** Учитывая, что матрицы выходящего потока  $Z_i$  имеют блочный трехдиагональный вид, они будут сильно разрежены. Благодаря этому можно попытаться применить более эффективные способы решения системы алгебраических уравнений. Однако, показатель степени в оценке сложности все равно не окажется ниже, чем  $(2 + \varepsilon)N$ .

На практике утверждения 4.1 и 4.2 означают, что искать решения с помощью представленной схемы вычислений становится невозможным даже при относительно небольших  $N$ ,  $V$  и  $W$ . В таблице 5 приведены примеры роста порядков для различных порядков входящих процессов и времен обслуживания.

Таким образом, для практического применения открытых сетей с узлами МАР/РН/1/М необходимы более эффективные методы расчета.

			Номер станции				
$W$	$V$	$M$	1	2	3	4	5
Сети без кросс-трафика							
1	1	1	3	9	27	81	243
1	1	3	5	25	125	625	3'125
2	2	2	16	128	1'024	8'192	65'536
3	1	3	15	75	375	1'875	9'375
1	3	3	15	225	3'375	50'625	759'375
3	3	3	45	675	10'125	151'875	2'278'125
Сети с кросс-трафиком							
1	1	1	3	9	27	81	243
1	1	3	5	25	125	625	3'125
2	2	2	16	256	4'096	65'536	1'048'576
3	1	3	15	225	3'375	50'625	759'375
1	3	3	15	225	3'375	50'625	759'375
3	3	3	45	2'025	91'125	4'100'625	184'528'125

Таблица 5 — Порядки выходящих МАР-поток в зависимости от порядков РН-распределения ( $V$ ), входящих МАР-поток ( $W$ ) и емкости очереди ( $M$ )

### 4.3 Расчет характеристик сети массового обслуживания методом Монте-Карло

Расчет методом Монте-Карло заключается в многократном проигрывании прохождения пакетов по сети массового обслуживания и замерах длительностей их пребывания в узлах, длин очередей и потерь. Для реализации этого метода лучше всего подходят системы дискретно-событийного моделирования, в которых обрабатываются возникающие события, а время изменяется только в начале обработки очередного события.

Методы и алгоритмы имитационного моделирования открытых сетей массового обслуживания хорошо известны, поэтому далее приведем лишь краткое схематичное описание работы модели и способа расчета характеристик сети.

Для моделирования сети с узлами МАР/РН/1/М нужно описать обработку двух типов событий: появление нового пакета во входящем МАР-потоке и завершение обработки пакета на узле.

Пусть, как и ранее,  $N$  — число станций в сети. Обозначим  $K$  — глобальный счетчик пакетов,  $T_k^a$  — время появления  $k$ -го пакета,  $\Delta$  — множество вычисленных задержек,  $a_i$  — число сгенерированных пакетов из  $i$ -го потока,  $l_i$  — число потерянных пакетов на  $i$ -й станции,  $t$  — модельное время,  $n_i$  — число пакетов на  $i$ -й станции.

Если какая-либо из перечисленных величин  $\chi$  будет нас интересовать в определенный момент модельного времени  $t$ , то будем писать  $\chi(t)$ . Кроме того, при описании операций будем использовать запись  $\chi' := f(\chi)$ , подразумевая, что после завершения выполнения те-



кущего действия величина  $\chi$  примет значение  $\chi'$ , а до окончания операции она сохраняет значение  $\chi$ .

### 4.3.1 Обработка событий

При **инициализации** модели для каждого потока  $X_i$  выбирается интервал до поступления первого пакета  $\tau_i$ , и на время  $\tau_i$  назначается обработка события поступления пакета на  $i$ -ю станцию.

Рассмотрим действия, происходящие при возникновении событий поступления нового пакета и завершении обслуживания.

**Обработка поступления нового пакета.** Если в момент времени  $t = t_0$  во входящем потоке, поступающем в узел с номером  $i$ , происходит появление пакета, выполняются следующие действия.

*Шаг 1.* Пакет обрабатывается в зависимости от того, занят ли прибор и есть ли место в очереди:

- Если  $n_i = 0$ , то есть обслуживающий прибор узла  $i$  свободен, то вычисляется случайная длительность обслуживания  $\tau_s$ ,  $\mathbb{P}\{\tau_s \leq \tau\} = F_{Y_i}(\tau)$ , на момент времени  $t_0 + \tau_s$  назначается событие окончания обслуживания на  $i$ -м приборе, а число пакетов в системе увеличивается на единицу:  $n'_i := n_i + 1$ .
- Если  $1 \leq n_i \leq M_i$ , то есть обслуживающий прибор занят, но в очереди есть место, то пакет помещается в очередь, и счетчик числа пакетов в системе увеличивается на единицу:  $n'_i := n_i + 1$ .
- Если  $n_i = M_i + 1$ , то есть в очереди нет места, пакет теряется, а счетчик потерянных пакетов  $l_i$  увеличивается на единицу:  $l'_i := l_i + 1$ .

*Шаг 2.* Для пакета сохраняется время его появления  $T_K^a := t_0$ . *Шаг 3.* Счетчики числа сгенерированных пакетов  $K$  и  $a_i$  увеличиваются на единицу:  $K' := K + 1$ ,  $a'_i := a_i + 1$ .

*Шаг 4.* Выбирается случайное время до следующего появления пакета в  $i$ -м потоке  $\tau_a$ ,  $\mathbb{P}\{\tau_a \leq \tau\} = F_{X_i}(\tau)$ , на момент  $t_0 + \tau_a$  назначается следующее событие появления нового пакета в  $i$ -м потоке.

**Завершение обслуживания пакета.** Если в момент времени  $t_0$  завершается обслуживание  $k$ -го пакета в узле  $i$ , то выполняется следующее:

*Шаг 1.* Если  $i < N$ , то пакет обрабатывается в зависимости от того, занят ли прибор на следующей станции, и есть ли место в ее очереди:

- Если  $n_{i+1} = 0$ , то есть обслуживающий прибор узла  $i + 1$  свободен, то вычисляется случайная длительность обслуживания  $\tau_s$ ,  $\mathbb{P}\{\tau_s \leq \tau\} = F_{Y_{i+1}}(\tau)$ , на момент времени  $t_0 + \tau_s$  назначается событие окончания обслуживания на  $i + 1$ -м приборе, а число пакетов в системе увеличивается на единицу:  $n'_{i+1} := n_{i+1} + 1$ .

- Если  $1 \leq n_{i+1} \leq M_{i+1}$ , то есть обслуживающий прибор занят, но в очереди есть место, то пакет помещается в очередь  $i + 1$ -й станции и увеличивается счетчик числа пакетов в системе:  $n'_{i+1} := n_{i+1} + 1$ .
- Если  $n_{i+1} = M_{i+1}$ , то есть в очереди нет места, то пакет теряется, а счетчик потерянных пакетов  $l_{i+1}$  увеличивается на единицу:  $l'_{i+1} := l_{i+1} + 1$ .

*Шаг 2.* Если  $i = N$ , то в  $\Delta$  добавляется величина  $t_0 - T_k^a$ , то есть время с появления пакета до настоящего времени. *Шаг 3.* Число пакетов на  $i$ -й станции уменьшается:  $n'_i := n_i - 1$ . *Шаг 4.* Если в очереди  $i$ -й станции есть еще пакеты, то есть  $n_i > 0$ , то вычисляется случайная длительность обслуживания  $\tau_s$ ,  $\mathbb{P}\{\tau_s \leq \tau\} = F_{Y_i}(\tau)$ , на момент времени  $t_0 + \tau_s$  назначается событие окончания обслуживания на  $i$ -м приборе.

### 4.3.2 Расчет характеристик

В момент времени  $t$  межконцевую среднюю задержку  $\bar{T}(t)$  можно оценить как

$$\bar{T}(t) = \frac{1}{|\Delta(t)|} \sum_{\delta \in \Delta(t)} \delta,$$

а вероятность успешной доставки пакета  $P_s(t)$  — как долю доставленных пакетов из всех пакетов, успевших к моменту времени  $t$  покинуть систему.

$$P_s(t) = 1 - \frac{\sum_{i=1}^N l_i(t)}{K(t) - \sum_{i=1}^N n_i}$$

Симуляция производится либо до достижения модельным временем заданного значения  $T_{max}$ , либо до того, как отклонение искомым характеристик не становится меньше заданной величины  $\varepsilon$ . Для реализации последнего критерия, можно вычислять оценки  $\bar{T}(t)$  и  $P_s(t)$  в моменты модельного времени  $t_1, t_2, \dots$  и останавливать симуляцию в момент времени  $t_n$ , если  $(\bar{T}(t_n) - \bar{T}(t_{n-1})) / \bar{T}(t_n) < \varepsilon$  и  $(P_s(t_n) - P_s(t_{n-1})) / P_s(t_{n-1}) < \varepsilon$ .

Для повышения достоверности результатов можно провести симуляцию несколько раз, и в качестве результатов эксперимента использовать усредненные значения по всем проведенным симуляциям.

### 4.3.3 Программная реализация

Выбор конкретных алгоритмов и методов реализации имитационной модели зависит от того, используется ли готовая система имитационного моделирования, или же модель

пишется без использования готовых систем. Пакеты для моделирования систем массового обслуживания входят в такие системы, как OMNeT++, MatLab, AnyLogic и другие.

В диссертационном исследовании автором была разработана высокоэффективная реализацию модели сети с МАР-потоками и РН-распределениями на языке C++, и разработан интерфейс для ее подключения в код на Python с помощью Cython. Реализация модели, а также все приведенные в исследовании численные эксперименты, доступны на GitHub: <https://github.com/larioandr/thesis-queues>. Все алгоритмы для расчета характеристик в имитационной модели реализованы так, чтобы использовать ограниченный объем памяти. Кроме реализации имитационной модели, пакет содержит классы для представления МАР-потоков, РН-распределений, систем МАР/РН/1/М и методы для аналитического расчета их характеристик.

#### 4.3.4 Преимущества и недостатки метода

Главное преимущество метода Монте-Карло для вычисления характеристик сети массового обслуживания — его простота. Так как при выполнении модели не происходит явного построения выходных потоков, нет и экспоненциального роста размера задачи при увеличении числа станций. Точность получаемых результатов связана только со временем, которое выделено для проведения эксперимента. Наконец, в реальной имитационной модели собирается гораздо больше характеристик сети, чем было описано выше. Например, вычисляются оценки моментов и коэффициентов корреляции в выходных потоках и средние длины очередей.

Метод также обладает серьезными недостатками, основной из которых — невозможность эффективного повторного использования результатов: даже при небольшом изменении входных данных (например, размера сети или одного из входных потоков) требуется заново выполнять весь эксперимент. Если МАР-потоки или РН-распределения имеют большие размерности, для генерации случайных величин нужно больше времени. Также много времени требуется, если сеть содержит большое число станций.

Главный недостаток метода Монте-Карло — зависимость точности результатов от числа промоделированных событий. Это означает, что какой бы эффективной не была реализация, для получения качественных результатов все равно потребуется сгенерировать определенное число случайных величин. Например, при моделировании сети с десятью узлами для получения высокой точности с погрешностью в пределах 5% приходится генерировать порядка 100000 пакетов. Таким образом, у метода Монте-Карло оказывается весьма ограниченный потенциал к ускорению вычислений. Так, можно ускорить вычисление с нескольких секунд до ста миллисекунд, за счет использования более эффективных алгоритмов обработки статистики, более быстрых структур данных и других оптимизаций, но нельзя получить время, меньшее, чем требуется для генерации  $N$  экспоненциально-распределенных случайных величин, причем это число  $N$  может иметь порядок  $10^5 - 10^7$ . Как будет показано далее, в

ходе численного эксперимента было обнаружено, что использование аппроксимаций выходящих потоков позволяет получить результаты в среднем даже быстрее, хотя и с несколько меньшей точностью.

Для получения оценок исследуемых характеристик за существенно меньшее время (порядка микросекунд) нужно использовать радикально иной подход, позволяющий отказаться от необходимости генерировать большое число случайных величин. В качестве такого подхода можно использовать, например, машинное обучение. Подробное рассмотрение данного метода выходит за рамки диссертационной работы.

#### 4.4 Расчет характеристик сети методом понижения размерности выходящих потоков

В качестве альтернативы методу Монте-Карло, для расчета характеристик открытой сети массового обслуживания с узлами МАР/РН/1/М в настоящей работе предлагается использовать метод итерационного расчета с понижением размерности выходящих потоков. Идея метода заключается в том, чтобы заменять МАР-потоки обслуженных пакетов другими МАР-потоками, имеющими те же или близкие значения первых  $K$  моментов и коэффициентов корреляции с лагами  $1, 2, \dots, L$ , но обладающими меньшим порядком. В этом случае экспоненциального роста размерности исходящих потоков нет: исходящий поток из любой станции имеет порядок не выше  $(M_i + 2)V_i(W + W_i)$ , где  $M_i$  — емкость очереди на  $i$ -й станции,  $V_i$  — порядок РН-распределения  $i$ -го обслуживающего прибора,  $W_i$  — порядок внешнего МАР-потока, поступающего на  $i$ -ю фазу, а  $W$  — максимальный порядок аппроксимирующих МАР-потоков. Если кросс-трафика нет и  $i > 1$ , то в этой формуле  $W_i = 0$ , а если  $i = 1$ , то  $W = 0$ .

Если для работы алгоритма восстановления МАР-потока  $\mathcal{A}_o$  нужны  $K$  первых моментов и  $L$  первых коэффициентов корреляции, то шаг 2 схемы расчета, описанной в разделе 4.2.3 заменяется шагом 2\*:

*Шаг 2\**. Если  $i = 1$ , то положим  $\hat{X}_i = X_1$ . В противном случае ( $i > 1$ ):

1. Для потока  $Z_{i-1} \sim \text{МАР}(D'_0, D'_1)$  вычисляем  $P$  и  $\bar{\alpha}$  согласно (4.3).

2. Вычисляем  $\bar{\mu} \in \mathbb{R}^K$ ,  $\mu_i = i! \bar{\alpha} (-D'_0)^{-i} \bar{\Gamma}$

3. Вычисляем  $\bar{\nu} \in \mathbb{R}^L$ ,  $\rho_i = \frac{\bar{\theta} P^i (-D'_0)^{-i} \bar{\Gamma} - 1}{\mu_i (2\bar{\theta} (-D'_0)^{-i} \bar{\Gamma} - 1)}$ .

4. Находим  $\tilde{Z}_{i-1} = \mathcal{A}_o(\bar{\mu}, \bar{\rho})$

5. Если в сети есть кросс-трафик, то полагаем  $\hat{X}_i = \tilde{Z}_{i-1} \oplus X_i$ , а в противном случае  $\hat{X}_i = \tilde{Z}_{i-1}$ .

При расчета коэффициентов корреляции можно использовать стационарное распределение  $\bar{\theta}$ , найденное на шаге 4 на предыдущей итерации.

Для понижения размерности системы будем использовать четыре варианта метода моментов. Выбор этих методов обусловлен их вычислительной простотой, а также тем, что

различия в точности позволят лучше определить влияние моментов и коэффициента корреляции на суммарные задержки.

1. Аппроксимацию по среднему  $m_1$  экспоненциальным распределением;
2. Аппроксимацию по среднему  $m_1$  и коэффициенту вариации  $c$  с помощью экспоненциальных и гиперэкспоненциальных распределений и распределения Эрланга;
3. Аппроксимацию РН-распределениями по среднему  $m_1$  и коэффициентам вариации  $c$  и асимметрии  $\gamma$ ;
4. Аппроксимацию МАР-поток по  $m_1$ ,  $c$ ,  $\gamma$  и коэффициенту корреляции  $\rho_1$ .

В первых трех вариантах МАР-поток строится по РН-распределению, его корреляция с любым положительным лагом равна нулю. Матрица  $D_1$  вычисляется с помощью (4.6) (см. рис. 4.2). В четвертом варианте, аппроксимирующий МАР-поток строится по трем моментам и коэффициенту корреляции. Сначала, по первым трем моментам, найдем стационарное РН-распределение, матрицей которого будет матрица  $D_0$  искомого МАР-потока, с помощью метода, предложенного в работе Johnson и Taafe [111]. Затем, с помощью метода, предложенного Horvath, Bucholz и Telek [149], найдем подходящую матрицу  $D_1$ . Этот метод аппроксимации прост в расчете, позволяет быстро находить потоки, хотя их размерность может быть не самой низкой.

Каждый метод будем применять к выходящим (обслуженным) потокам. Кроме того, их можно применять и к входящему МАР-потoku, однако, в этом случае результаты оказываются хуже. Численные результаты будут приведены далее в этой главе, а сейчас рассмотрим каждый из методов аппроксимации подробнее. Будем обозначать среднее значение, интенсивность, коэффициент вариации, коэффициент асимметрии и коэффициент корреляции исходного (аппроксимируемого) потока  $X \sim MAP(D_0, D_1)$  как  $m_1$ ,  $\lambda = 1/m_1$ ,  $c$ ,  $\gamma$  и  $\rho_1$ , а те же характеристики построенного (аппроксимирующего) потока  $\tilde{X} \sim MAP(\tilde{D}_0, \tilde{D}_1)$  как  $\tilde{m}_1$ ,  $\tilde{\lambda} = 1/\tilde{m}_1$ ,  $\tilde{c}$ ,  $\tilde{\gamma}$  и  $\tilde{\rho}_1$ .

#### 4.4.1 Аппроксимация по среднему

Самый простой способ аппроксимировать поток  $X \sim MAP(D_0, D_1)$  с интенсивностью  $\lambda = 1/MX$  — заменить его экспоненциальным распределением  $X' \sim Exp(\lambda)$ . Представление  $\tilde{X}$  в виде МАР-потока будет иметь следующий вид:

$$\tilde{D}_0 = (-\lambda) \quad \tilde{D}_1 = (\lambda) \quad (4.15)$$

Такой поток будет обладать коэффициентом вариации  $\tilde{c} = \tilde{\sigma}/M\tilde{X} \equiv 1$ , коэффициентом асимметрии  $\tilde{\gamma} \equiv 2$  и коэффициентом корреляции  $\tilde{\rho}_1 \equiv 0$ .

Главное преимущество этого подхода — высокая скорость, так как для заменяемого потока достаточно вычислить стационарное распределение (см. (4.4)) и интенсивность (см. (4.5)). Кроме того, размерность построенного МАР-потока всегда равна единице, поэтому

размерность выходящего потока из системы  $M/PN/1/M$  равна  $(M + 2)V$ . Однако, если коэффициент вариации существенно отличается от единицы, такая замена может привести к очень существенной ошибке. Поэтому желательно рассматривать большее число моментов. Кроме того, известно [206], что наличие положительной корреляции существенно влияет на характеристики системы, в частности, увеличивает время ожидания, в том числе при фиксированной средней интенсивности входящего потока. Этот метод будем рассматривать как наименее точный.

#### 4.4.2 Аппроксимация по двум моментам

Для аппроксимации МАР-потока  $X \sim MAP(D_0, D_1)$  другим потоком  $\tilde{X}$  с таким же средним  $\tilde{m}_1 = m_1$  и близким коэффициентом вариации  $\tilde{c} \approx c$  можно подобрать простое РН-распределение по этим двум моментам. В зависимости от значения коэффициента вариации, будем использовать экспоненциальные распределения, гиперэкспоненциальные распределения или распределения Эрланга (см. рис. 4.4). Выбор этих распределений обусловлен хорошо известным фактом, что у экспоненциального распределения коэффициент вариации  $c$  всегда равен единице, у распределения Эрланга  $c \leq 1$ , а у гиперэкспоненциального распределения всегда  $c \geq 1$ .

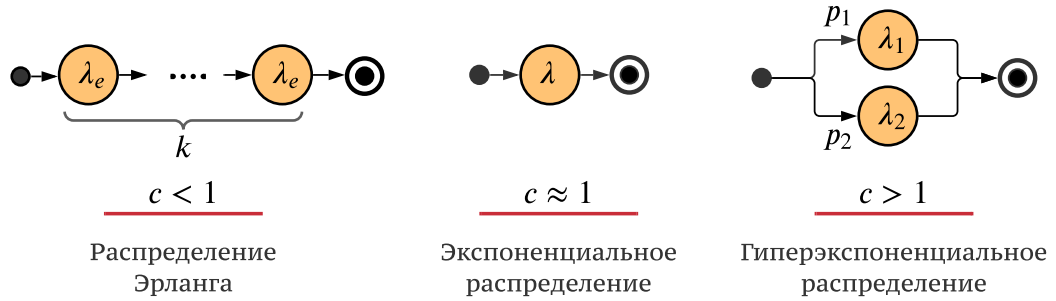


Рисунок 4.4 — РН-распределения для аппроксимации потоков по двум моментам

Если  $c \approx 1$ , то  $\tilde{X} \sim Exp(\lambda)$ . Матрицы МАР-потока  $MAP(\tilde{D}_0, \tilde{D}_1)$  определяются с помощью выражения (4.6).

Если  $c > 1$ , то  $\tilde{X}$  — гиперэкспоненциальное распределение с вероятностями  $\bar{p} = (p_1, p_2)$  и интенсивностями  $\bar{\lambda} = (\lambda_1, \lambda_2)$ . Отметим, что интенсивности  $\lambda_1, \lambda_2$  и вероятности  $p_1$  и  $p_2 = 1 - p_1$  определяются неоднозначно, так как ограничение на среднее и коэффициент вариации дают только два уравнения при трех неизвестных. Будем использовать следующее решение:

$$\lambda_2 = \frac{1}{m_1} + 1, \quad \lambda_1 = \frac{m_1}{b\lambda_2 - m_1}, \quad p_1 = \frac{m_1^2}{b\lambda_2^2 - 2m_1\lambda_2 + 1},$$

где  $b = m_1^2(c^2 + 1)/2$ . Матрицы МАР-потока будут иметь вид:

$$\tilde{D}_0 = \begin{pmatrix} -\lambda_1 & 0 \\ 0 & -\lambda_2 \end{pmatrix}, \quad \tilde{D}_1 = \begin{pmatrix} p_1\lambda_1 & p_2\lambda_1 \\ p_1\lambda_2 & p_2\lambda_2 \end{pmatrix}.$$

Наконец, если  $c < 1$ , то  $\tilde{X}$  – распределение Эрланга с числом состояний  $k = [1/c^2]$  и параметром (интенсивностью каждого состояния)  $\lambda_e = k/m_1$ . Матрицы МАР-потока:

$$\tilde{D}_0 = \begin{pmatrix} -\lambda_e & \lambda_e & 0 & \cdots & 0 \\ 0 & -\lambda_e & \lambda_e & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda_e \end{pmatrix}, \quad \tilde{D}_1 = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \lambda_e & 0 & \cdots & 0 \end{pmatrix}.$$

Отметим, что коэффициент вариации  $\tilde{c}$  может не совпадать в точности с исходным  $c$  из-за округления.

### 4.4.3 Аппроксимация по трем моментам

Для большей точности результатов можно учитывать не только среднее значение  $m_1$  и коэффициент вариации  $c$  МАР-потока  $X$ , но и его коэффициент асимметрии  $\gamma$ . Существует ряд исследований, посвященных построению РН-распределений по трем и более моментам [111; 140–145; 207]. Для упрощения задачи, поиск проводится не на всем классе РН-распределений, а на некотором его подмножестве. Например, в работах [141; 142] используются ациклические РН-распределения (АРН), а в работе [111] используется распределение, представленное смесью двух распределений Эрланга с одинаковым числом фаз ( $ME_n(2)$ ).

В общем случае РН-распределение можно найти для любого набора значений  $m_1 > 0$ ,  $c > 0$  и  $\gamma > c - 1/c$ , которыми может обладать какое-либо непрерывное положительное распределение [111]. Однако, порядок такого РН-распределения может быть очень большим. Например, в работе [151] было доказано, что при  $c < 1$  самым малым порядком среди всех распределений с таким коэффициентом вариации обладает распределение Эрланга. При этом порядок распределений Эрланга равен  $1/c^2$ , то есть растет квадратично при приближении коэффициента вариации к нулю. Таким образом, если хочется получить РН-распределение с ограниченным числом фаз, то неизбежно приходится также ограничивать и область значений параметров  $m_1, c, \gamma$  (особенно в области  $c < 1$ ). Если же нужно иметь возможность получить РН-распределение для произвольных  $m_1, c, \gamma$ , то приходится смириться с неограниченным ростом сложности такого распределения.

В диссертационной работе будут использоваться два простых метода. Во-первых, попытаемся построить ациклическое РН-распределение второго порядка (АСРН(2)) в каноническом виде (см. рис. 4.5, левая схема) с помощью метода, описанного в работе Telek и Heindl [142]. Если моменты не попадают в область существования АСРН(2) (см. рис. 4.6), то используем более универсальный метод, продолженный Johnson и Taafe [111], с помощью которого РН-распределение ищется в виде смеси двух распределений Эрланга  $ME_n(2)$  (см. рис. 4.5, правая схема). Опишем эти методы подробнее.

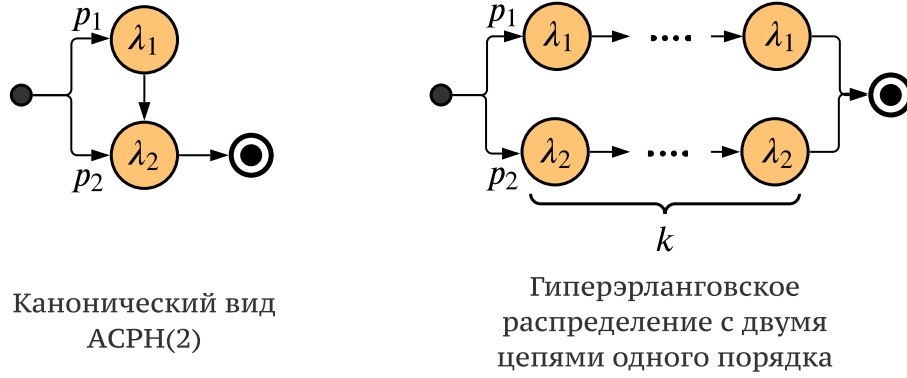


Рисунок 4.5 — Ациклическое РН-распределение АСРН(2) и гиперэрланговское распределение  $ME_n(2)$ , используемые при аппроксимации потоков по трем моментам

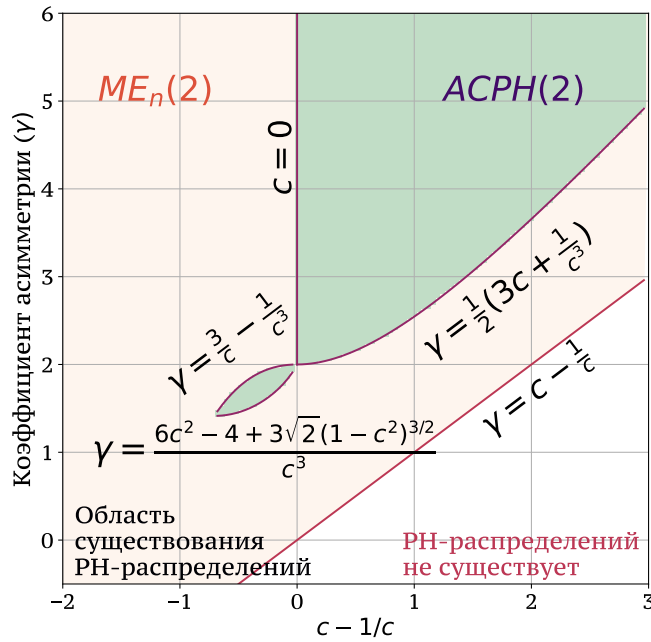


Рисунок 4.6 — Области существования РН-распределений. Зеленым цветом показана область существования АСРН(2) [142], розовым – произвольных РН-распределений, в частности  $ME_n(2)$  [111]. Ниже линии  $\gamma = c - 1/c$  РН-распределений не существует

Ациклические РН-распределения второго порядка АСРН(2) существуют в ограниченной области [142]. Границы области можно выразить через коэффициенты асимметрии  $\gamma$  и вариации  $c$ :

$$\frac{6c^2 - 4 + 3\sqrt{2}(1-c^2)^{3/2}}{c^3} \leq \gamma \leq \frac{3c^2 - 1}{c^3}, \quad \frac{1}{\sqrt{2}} \leq c \leq 1 \quad (4.16)$$

$$\frac{1}{2}\left(3c + \frac{1}{c^3}\right) \leq \gamma, \quad c > 1$$

При  $c < 1/\sqrt{2}$  распределений АСРН(2) не существует. Область существования показана на рис. 4.6.

Распределение АСРН(2) определяется тремя параметрами  $\lambda_1$ ,  $\lambda_2$  и  $p_1$  (вероятность  $p_2 = 1 - p_1$ ), см. рис. 4.5. Их можно вычислить следующим образом:

$$p = \frac{-B + 6m_1D + \sqrt{A}}{B + \sqrt{A}}, \quad \lambda_1 = \frac{B - \sqrt{A}}{C}, \quad \lambda_2 = \frac{B + \sqrt{A}}{C} \quad (4.17)$$



где  $D = 2m_1^2 - m_2$ ,  $C = 3m_2^2 - 2m_1m_3$ ,  $B = 3m_1m_2 - m_3$  и  $A = B^2 - 6CD$ . Здесь  $m_i = \mathbb{M}X^i$  –  $i$ -й момент случайной величины. Моменты  $m_2$  и  $m_3$  можно выразить через коэффициент вариации  $c$  и асимметрии  $\gamma$  как  $m_2 = m_1c^2 + m_1^2$  и  $m_3 = m_1^3(c^3\gamma + 3c^2 + 1)$ . Матрица  $S$  и вектор начального распределения  $\bar{\tau}$  для АСРН(2) будут иметь следующий вид:

$$S = \begin{pmatrix} -\lambda_1 & \lambda_1 \\ 0 & -\lambda_2 \end{pmatrix}, \quad \bar{\tau} = (p_1 \quad 1 - p_1). \quad (4.18)$$

Если коэффициенты  $c$  и  $\gamma$  аппроксимируемого потока не удовлетворяют ограничениями (4.16), то будем искать РН-распределение в классе  $\text{ME}_n(2)$ . В отличие от узкого класса АСРН(2), распределения  $\text{ME}_n(2)$  существуют для любого значения  $\gamma > c - 1/c$ , то есть их область существования совпадает с областью существования произвольных РН-распределений [111]. Соотношение между областями существования АСРН(2) и  $\text{ME}_n(2)$  показано на рис. 4.6. Распределения  $\text{ME}_n(2)$  представляют собой смесь двух распределений Эрланга одинакового порядка  $n$  с вероятностями  $p_1$  и  $p_2 = 1 - p_1$ . Распределения Эрланга имеют параметры  $\lambda_1$  и  $\lambda_2$  (см. рис. 4.5). Минимальный порядок  $n^*$  распределений Эрланга в  $\text{ME}_n(2)$  для заданных значений  $c$  и  $\gamma$  определяется согласно утверждению 4 из работы [111]:

$$n^* = \lceil \max\left\{ \frac{1}{c^2}, \frac{-\gamma + 1/c^3 + 1/c + 2c}{\gamma - (c - 1/c)} \right\} \rceil \quad (4.19)$$

После выбора  $n \geq n^*$  можно вычислить параметры  $\lambda_1$ ,  $\lambda_2$  и  $p_1$ . Способ вычисления описан в теореме 3 [111]:

$$\lambda_i^{-1} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}, \quad p_1 = \frac{m_1/n - \lambda_2^{-1}}{\lambda_1^{-1} - \lambda_2^{-1}}, \quad (4.20)$$

где

$$\begin{aligned} A &= n(n+2)m_1y \\ B &= -\left( nx + \frac{n(n+2)}{n+1}y^2 + (n+2)m_1^2y \right) \\ C &= m_1x \\ x &= m_1m_3 - \frac{n+2}{n+1}m_2^2 \\ y &= m_2 - \frac{n+1}{n}m_1^2 \end{aligned} \quad (4.21)$$

Матрица  $S$  и вектор начального распределения  $\bar{\tau}$  для РН-распределения из класса  $\text{ME}_n(2)$  будут иметь вид:



тремя моментами и коэффициентом корреляции между соседними интервалами между возникновением событий в МАР-потоке. Как и ранее, ключевая идея здесь заключается в том, чтобы понизить размерность МАР-потока: если учитывать только ограниченный набор характеристик, то есть шанс получить МАР-поток меньшего размера.

Для поиска аппроксимирующего МАР-потока будем пользоваться методом, предложенным Horvath и соавторами в работе [149]. Согласно этому подходу, матрицы  $D_0$  и  $D_1$  ищутся независимо по значениям моментов и коэффициентам корреляции. Метод основан на том, что стационарное распределение МАР-потока, определяющее его моменты, есть  $PH(D_0, \bar{\alpha})$ , где  $\bar{\alpha}$  — стационарное распределение вложенной по моментам появления событий цепи с матрицей  $P = (-D_0)^{-1}D_1$ . Поэтому можно найти матрицу  $D_0$  и вектор  $\bar{\alpha}$ , используя любой из ранее описанных методов для поиска РН-распределений, а информацию о значениях коэффициентов корреляции использовать для поиска матрицы  $D_1$ .

В общем случае поиск элементов матрицы  $D_1 \in \mathbb{R}^{W \times W}$  по известным коэффициентам корреляции с лагами  $i = 1, 2, \dots, L$  можно сформулировать как задачу оптимизации:

$$D_1 = \arg \min \sum_{i=1}^L w_i \left( \frac{\lambda \bar{\pi} (-D_0)^{-k} D_1^k (-D_0)^{-1} \bar{\mathbf{1}} - 1}{2 \bar{\pi} (-D_0)^{-1} \bar{\mathbf{1}} - 1} - \rho_i \right)^2 \quad (4.23)$$

с ограничениями

$$\begin{aligned} D_1 \bar{\mathbf{1}} &= -D_0 \bar{\mathbf{1}} \\ \bar{\alpha} (-D_0)^{-1} D_1 &= \bar{\alpha} \\ \{D_1\}_{ij} &\geq 0, \quad i, j = \overline{1, W}. \end{aligned} \quad (4.24)$$

Здесь  $w_i$ ,  $i = 1, 2, \dots, L$  — веса, с помощью которых можно регулировать важность приближения того или иного момента. Отметим, что ограничения (4.24) линейны относительно элементов матрицы  $D_1$ . Кроме ограничений (4.24) также нужно убедиться, что управляющая цепь с генератором  $D = D_0 + D_1$  неприводима.

Ограничением применения метода является то, что диапазон возможных значений  $\rho_i$  определяется выбором стационарного РН-распределения. В общем случае, для заданных  $m_1$ ,  $c$  и  $\gamma$  РН-распределение определяется неоднозначно, его вид зависит от используемых методов. Возможна ситуация, когда РН-распределение найти удалось, а матрицу  $D_1$  для МАР-потока с заданными коэффициентами корреляции  $\rho_i$  — нет, хотя МАР-поток с искомыми параметрами существует.

Пусть  $L = 1$ . Функция

$$\rho(D_1) = \frac{\lambda \bar{\pi} (-D_0)^{-1} D_1 (-D_0)^{-1} \bar{\mathbf{1}} - 1}{2 \bar{\pi} (-D_0)^{-1} \bar{\mathbf{1}} - 1},$$

принимаяющая значение коэффициента корреляции, представляет собой линейную комбинацию элементов матрицы  $D_1$ . Значит, для нахождения минимального  $\underline{\rho}_1$  и максимально  $\hat{\rho}_1$  значения  $\rho_1$  при заданной матрице  $D_0$  можно использовать симплекс-метод:

$$\begin{aligned} \hat{\rho}_1 &= \max \rho(D_1) \\ \underline{\rho}_1 &= \min \rho(D_1) \end{aligned} \quad (4.25)$$

при ограничениях (4.24). Пользуясь выпуклостью задачи линейного программирования, если заданное значение  $\rho_1 \in [\underline{\rho}_1, \hat{\rho}_1]$ , то матрицу  $D_1$  можно найти, решая систему линейных алгебраических уравнений относительно элементов матрицы  $D_1$ :

$$\begin{cases} D_1 \bar{\mathbf{1}} & = -D_0 \bar{\mathbf{1}} \\ \bar{\alpha} (-D_0)^{-1} D_1 & = \bar{\alpha} \\ \frac{\lambda \bar{\pi} (-D_0)^{-1} D_1 (-D_0)^{-1} \bar{\mathbf{1}} - 1}{2\bar{\pi} (-D_0)^{-1} \bar{\mathbf{1}} - 1} & = \rho_1 \end{cases} \quad (4.26)$$

Отметим, что если порядок искомого МАР-потока  $W > 1$ , то у системы будет бесконечное множество решений. Если  $\rho_1 > \hat{\rho}$  или  $\rho_1 < \underline{\rho}_1$ , то будем брать ту матрицу  $D_1$ , которая соответствует найденному ранее экстремуму.

На практике иногда возникает ситуация, когда результатом решения проблемы (4.23) является такая матрица  $D_1$ , что цепь Маркова с генератором  $D = D_0 + D_1$  не является неприводимой. Для быстрой проверки неприводимости удобно использовать алгоритм Тарьяна [208] для поиска сильно-связных компонент графа. Сильно-связная компонента — это такой набор вершин графа, в котором из каждой вершины можно попасть в любую другую вершину. Если цепь состоит из единственной сильно-связной компоненты, то она является неприводимой. Для повышения качества результатов, при построении графа будем игнорировать переходы между состояниями цепи, вероятности которых крайне малы. Чтобы подчеркнуть важность проверки на неприводимость отметим, что в ходе численных экспериментов, которые будут обсуждаться далее, около 16% всех построенных МАР-потоков имели не неприводимые цепи.

Если полученная цепь  $D = D_0 + D_1$  не является неприводимой, то есть несколько возможных действий. Во-первых, так как проблемы (4.23) и (4.26) могут иметь множество решений, можно попытаться найти другую матрицу  $D_1$  с той же матрицей  $D_0$  и вектором  $\bar{\alpha}$ . Во-вторых, можно попытаться найти другое распределение  $PH(D_0, \bar{\alpha})$  с теми же моментами. Для этого можно использовать другой метод, или ослабить требования точности на третий и/или второй моменты. В-третьих, можно найти другую матрицу  $D_1$ , для которой  $\rho(D_1)$  будет близко, но не равно в точности значению  $\rho_1$ . В нашей работе используем третий подход.

В завершение отметим, что существуют и другие подходы к построению МАР-потоков по моментам и коэффициентам корреляции, например [146; 147]. Кроме того, МАР-потоки можно строить по выборке (ее можно получить из аппроксимируемого потока), используя EM-процедуру [130–132].

#### 4.4.4 Сложность и применимость метода

Хотя метод понижения размерности позволяет находить оценки характеристик сетей массового обслуживания с произвольно большим числом узлов, он, как и метод Монте-Карло, не может обеспечить очень быстрый расчет. Проблема состоит в том, что размерности

потоков нельзя сделать сколь угодно низкими. Хорошо известно [151], что если коэффициент вариации  $c$  искомого распределения меньше единицы, то самое малое число состояний  $K = \lceil 1/c^2 \rceil$  будет у распределения Эрланга. Таким образом, неограниченного роста размерности невозможно избежать даже тогда, когда для приближения используется всего два первых момента. Пусть  $\hat{c} = \min\{c_1, c_2, \dots, c_{N-1}\}$  — минимальный коэффициент вариации среди всех выходных потоков кроме последнего узла, который здесь не играет роли, и  $\hat{W} = \max\{\lceil 1/\hat{c}^2 \rceil, W\}$ . Тогда сложность алгоритма можно оценить как  $\mathcal{O}(N(\hat{W}MV)^3)$ . Конечно, это гораздо лучше экспоненциального роста сложности точного алгоритма, но все равно очень много. Еще стоит отметить, что хотя порядок РН-распределений слабее зависит от коэффициента асимметрии, чем от коэффициента вариации (см., например, [111]), для точного совпадения третьего момента и коэффициента корреляции также может потребоваться увеличить размерность. Наконец, поиск матрицы  $D_1$  даже по коэффициенту корреляции с единичным лагом требует в общем случае решать задачу оптимизации, и это приходится делать для каждого выходного потока.

#### 4.5 Моделирование задержки в канале

В диссертационной работе предметом исследования является многошаговая беспроводная сеть сбора данных с камер или RFID-считывателей, работающая под управлением протокола IEEE 802.11 и передающая трафик в одном направлении. Будем считать, что все станции этой сети работают в одном канале, антенны у всех станций имеют круговую диаграмму, и станции расположены таким образом, что передачу любой станции могут получить только ее непосредственные соседи (см. рис. 4.7). В такой сети коллизия на станции  $S_i$  возникнет при одновременной передаче ее соседей, то есть когда станция  $S_{i-1}$  будет вести передачу станции  $S_i$ , а станция  $S_{i+1}$  — станции  $S_{i+2}$ : из-за круговой диаграммы и использования общего канала, последнюю передачу также получит станция  $S_i$ .

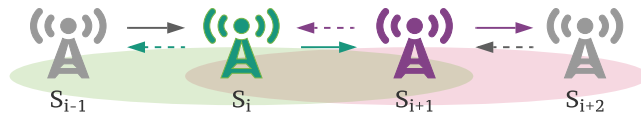


Рисунок 4.7 — Области видимости станций многошаговой беспроводной сети

Для моделирования длительности передачи пакетов в беспроводном канале с помощью РН-распределений нужно знать моменты этого распределения или иметь выборку длительностей. В диссертационном исследовании будем пользоваться методом восстановления РН-распределения по трем моментам, описанном в разделе 4.4.3, основанном на работах Johnson и Taafe [111] и Telek и Heindl [142].

Для получения оценок моментов распределения длительностей передачи пакетов можно использовать аналитические или имитационные модели каналов. Например, в работах

[200; 201] время передачи моделировалось с помощью полумарковского процесса для насыщенного режима, а также использовались простые имитационные модели каналов CSMA/CA для ненасыщенного режима. Если в качестве модели канала используется полумарковский процесс, то через вычисление функции генерации моментов можно получить значения математического ожидания и дисперсии для построенного этого процесса (см., например, работу Sakurai [81]). В результатах, представленных в диссертационном исследовании, использовалась более точная имитационная модель канала IEEE 802.11 из системы моделирования NS-3.

Длительность передачи в беспроводном канале определяется не только настройками приемников и передатчиков, но и вероятностью коллизий, которая зависит от числа конкурирующих станций. В исследуемой многошаговой сети станции работают в одном канале, поэтому вероятность коллизий зависит от положения станции внутри сети. Таким образом, для получения достаточно точных оценок нужно использовать разные РН-распределения на разных приборах.

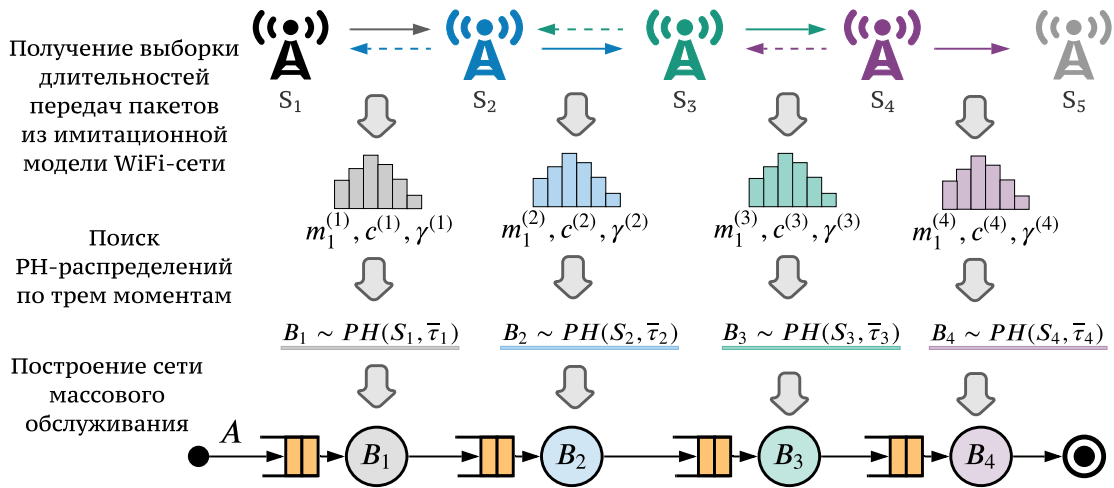


Рисунок 4.8 — Схема моделирования каналов беспроводной сети

Рассмотрим сеть, состоящую из пяти станций (см. рис. 4.8). Возможны четыре различных случая:

1. Канал  $S_1 \rightarrow S_2$ : канал занят, если станция  $S_2$  ведет передачу; возможна коллизия с передачами станции  $S_3$ .
2. Канал  $S_2 \rightarrow S_3$ : канал занят, если станции  $S_1$  или  $S_3$  ведут передачу; возможна коллизия с передачами станции  $S_4$ .
3. Канал  $S_3 \rightarrow S_4$ : канал занят, если станция  $S_2$  или  $S_4$  ведут передачу; станция  $S_5$  — конечная, далее не передает, поэтому коллизий нет.
4. Канал  $S_4 \rightarrow S_5$ : канал занят, если станция  $S_3$  ведет передачу; коллизий нет.

Для каждого из этих каналов строится отдельное РН-распределение  $B_1, B_2, B_3$  и  $B_4$  (номер распределения совпадает с номером станции-отправителя). Если сеть содержит меньше станций, то и распределений потребуется меньше.

Если добавятся промежуточные станции, то новые каналы будут испытывать нагрузку, аналогичную каналу  $S_2 \rightarrow S_3$ , поэтому сеть можно моделировать с помощью тех же распределений  $B_1, B_2, B_3$  и  $B_4$  (см. рис. 4.9).

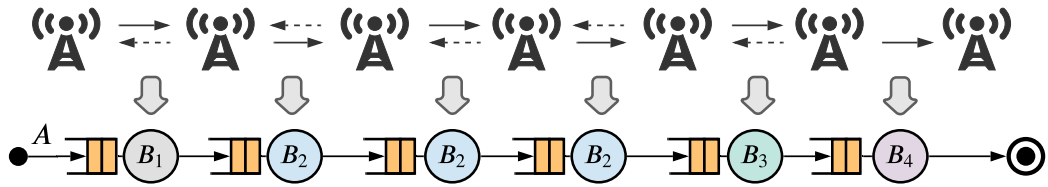


Рисунок 4.9 — Схема моделирования беспроводной сети с произвольным числом станций

#### 4.6 Численное исследование эффективности метода аппроксимации потоков

Для исследования эффективности метода вычисления оценок характеристик тандемной сети массового обслуживания, был проведен ряд численных экспериментов на случайном наборе данных. Входящий набор данных включал 2500 сетей со случайными значениями среднего времени обслуживания, коэффициентов вариации и асимметрии интервалов между пакетами и времен обслуживания, коэффициента корреляции входящего МАР-потока с единичным лагом, случайными емкостями очередей и случайным числом станций в сети. На всех наборах были найдены решения методом Монте-Карло и всеми методами аппроксимации потоков, описанных в разделе 4.4. На тех входных данных, на которых это было возможно, было найдено точное решение. Исходный код модели и численных экспериментов, а также документация, доступны на GitHub: <https://github.com/larioandr/thesis-queues>.

Сначала рассмотрим, как был сгенерирован набор входных данных для исследования методов аппроксимации выходящих потоков. Затем приведем результаты измерения ошибок, возникающих при использовании этих методов. В завершение покажем, как меняется точность метода Монте-Карло при увеличении числа сгенерированных пакетов.

##### 4.6.1 Построение набора данных для исследования

Для проведения численных экспериментов были сгенерированы 2000 входных параметров со случайными средними временами обслуживания, коэффициентами вариации интервалов между пакетами  $c_a$  и обслуживания  $c_s$ , коэффициентами асимметрии поступления  $\gamma_a$  и обслуживания  $\gamma_s$ , коэффициентами корреляции входящего потока  $\rho_l$ , числом узлов в сети и емкостью очередей. Интенсивность (и, соответственно, средний интервал) входящего потока во всех случаях полагалась равной единице, поэтому коэффициент загрузки первого узла был равен  $m_s/m_a = m_s$ . Границы значений параметров для стандартного генератора показаны в табл. 6. Отметим, что так как емкости очередей в системе ограничены, и переполняющие очередь пакеты теряются, исследовать системы с коэффициентом загрузки  $m_s/m_a > 1$  корректно. Для построения РН-распределений времени обслуживания по выбранным случайным параметрам  $m_s, c_s, \gamma_s$  использовались методы, описанные в разделах 4.4.1, 4.4.2 и 4.4.3, а для построения МАР-потоков по  $m_a \equiv 1, c_a, \gamma_a, \rho_1$  — метод из раздела 4.4.3.

Таблица 6 — Граничные значения параметров для исследования методов аппроксимации потоков

Параметр	Стандартный генератор		Простые распределения	
	Мин.	Макс.	Мин.	Макс.
$m_s$	0,1	1,5	0,1	1,5
$c_s, c_a$	0,5	4,0	0,5	4,0
$\gamma_{a,s}$	$c_{a,s} - 1/c_{a,s}$	6.0	Не используется	
$\rho_1$	Границы рассчитываются как решение задачи (4.25)			
Размер сети $N$	1	10	3	10
Емкость очереди $M$	0	10	0	3

Сложность итерационного алгоритма для точного расчета характеристик сети растет экспоненциально с ростом числа узлов. Для получения точного решения требовалось, чтобы порядок матриц выходящего потока с последнего узла был достаточно мал. В противном случае матрицы могут просто не поместиться в оперативной памяти или потребуют слишком много времени для выполнения операции обращения и получения решения системы линейных уравнений. Следуя этой логике, для получения точного решения использовались только те входные данные, на которых матрица потока обслуженных заявок на последнем узле имеет порядок не выше 8000. После такого отсева оставалось слишком мало входных наборов, поэтому дополнительно были сгенерированы 500 наборов входных параметров, в которых использовались более простые распределения и меньшие очереди: РН-распределение времени обслуживания и стационарное РН-распределение МАР-потоков строились по первым двум моментам (см. раздел 4.4.2), а не трем, длина очереди выбиралась из меньшего диапазона, а число узлов ограничивалось снизу, см. табл. 6.

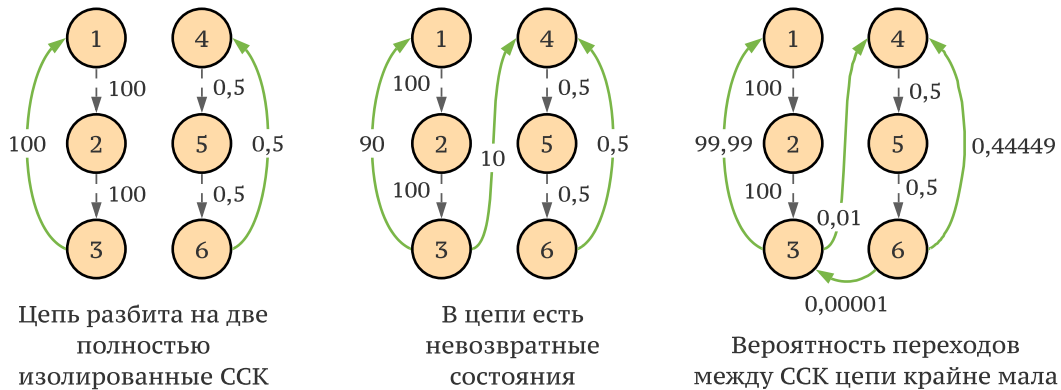


Рисунок 4.10 — Примеры отсеиваемых МАР-потоков. На стрелках показана интенсивность переходов. Сплошные линии — переходы матрицы  $D_1$ , пунктирные —  $D_0$

В итоге были получены 2500 входных наборов. Как было отмечено в разделе 4.4.3, иногда построенные МАР-потоки не были неприводимыми. В некоторых случаях пространство состояний распадалось на две сильно связанные компоненты, соответствующие цепочкам Эрланга. Примеры «плохих» МАР-потоков показаны на рис. 4.10. Если бы такие потоки



использовались при расчете методом Монте-Карло, были бы получены некорректные результаты. Например, запуская имитационную модель (реализацию метода Монте-Карло) два раза, и выбирая в качестве начального состояния МАР-потока состояние сначала из первой, а затем из второй компоненты, по сути рассматривались бы два разных потока, так как попасть из состояний одной компоненты в другую невозможно, и, таким образом, были бы получены разные результаты. Для того, чтобы быстро отсеять «плохие» потоки, был использован алгоритм Тарьяна поиска сильно-связных компонентов в графе [208]. Множество вершин такого графа совпадает с множеством состояний цепи, а дуга между состояниями  $i$  и  $j$  в графе есть только в том случае, если вероятность перехода во вложенной цепи  $\mathbb{P}\{X_{n+1} = j | X_n = i\} > 10^{-3}$ . Нижняя граница нужна, чтобы отсеять случаи, когда переходы между компонентами есть, но крайне маловероятны: такие переходы срабатывают редко, а переход по ним может существенно изменить оценки характеристик, поэтому для получения устойчивых результатов нужно моделировать огромное число событий.

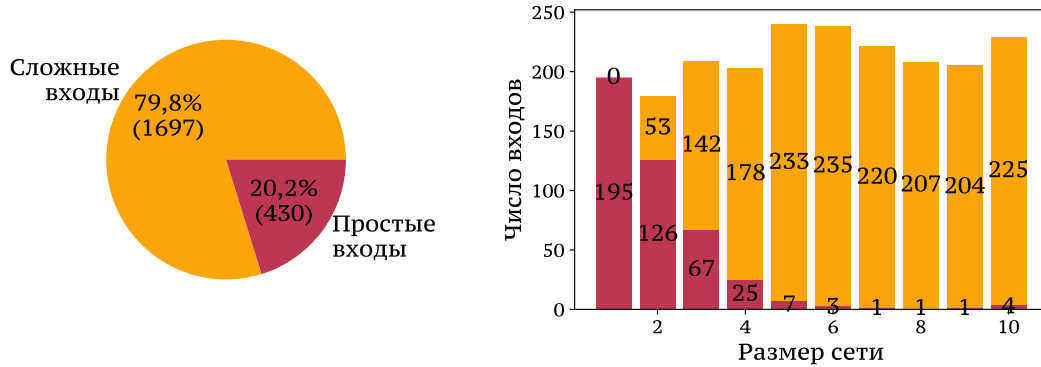


Рисунок 4.11 — Разбиение наборов входных данных на простые и сложные

В результате отсеивания было удалено 373 входных набора, то есть 14,9 % данных. Оставшиеся наборы были поделены на «простые» и «сложные»: к простым наборам были отнесены те, для которых порядок выходящего потока, равный  $\hat{W}_N = W(V(M+2))^N$ , не превосходил 8000, а все остальные — к сложным. Ожидаемо, большинство простых наборов соответствовали сетям малого размера, см. рис. 4.11. Так как для больших сетей простых наборов оказалось слишком мало, в дальнейшем точное решение искалось только для тех простых наборов, в которых длина сети не превышала 5.

На рис. 4.12 показаны диаграммы разброса различных параметров в наборах простых входных данных. Можно обратить внимание на три особенности. Во-первых, наборы распределены практически равномерно относительно коэффициента загрузки  $m_s/m_a$ . Во-вторых, у большей части наборов коэффициенты вариации  $c_a$  и  $c_s$  больше единицы. Это можно объяснить тем, что при  $c < 1$  размерность РН-распределений не меньше, чем  $\lceil 1/c^2 \rceil$ , поэтому и  $\hat{W}_N$  оказывается в среднем больше, чем для распределений с  $c \geq 1$ . В-третьих, подавляющее большинство МАР-потоков имеют коэффициент корреляции  $\rho_1 \geq 0$ , причем у значительной части  $\rho_1 \approx 0$ . Это можно объяснить тем, что для тех матриц  $D_0$ , которые строились с помощью методов, описанных в разделах 4.4.2 и 4.4.3, нижняя граница  $\rho_1 \approx 0$ , а верхняя меняется от 0 до 0,8. Отметим, что среди наборов достаточно таких, у которых коэффициент корреляции  $\rho_1 > 0$ .

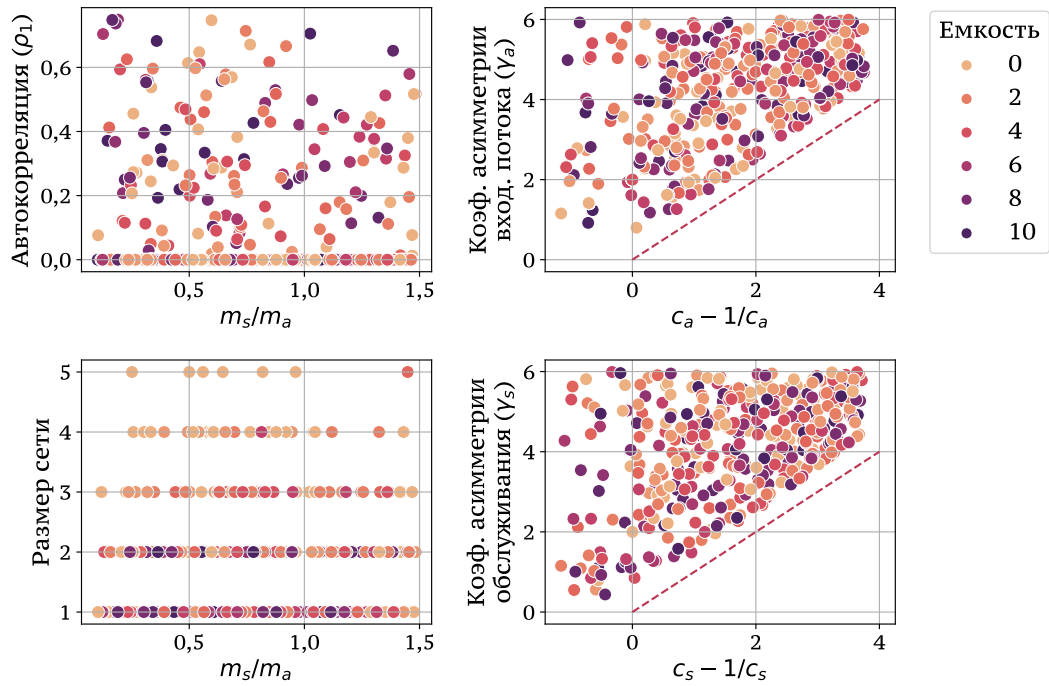


Рисунок 4.12 — Диаграммы разброса для различных сочетаний параметров в наборах простых входных данных

#### 4.6.2 Использование методов аппроксимации выходящего потока

С помощью алгоритма, приведенного в разделе 4.2.3, были найдены точные решения для всех простых входных данных, для которых порядок выходного МАР-потока с последнего узла сети не превышает 8000, а с помощью методов аппроксимации потоков и метода Монте-Карло — решения для всех входных наборов. Используя точные решения, для простых наборов были измерены относительные ошибки оценки трех параметров: суммарной задержки, вероятности доставки пакета и среднего числа пакетов на последнем узле сети.

Отметим, что суммарная задержка вычислялась как сумма средних задержек на всех узлах сети, что, вообще говоря, не равно межконцевой задержке. Для вычисления последней следовало бы учитывать задержки только тех пакетов, которые не были отброшены на каком-либо узле. Точная формула для расчета межконцевой задержки сложнее (см. [109, р. 429]).

Сначала были вычислены оценки исследуемых характеристик, заменяя выходящие потоки потоками Пуассона по среднему значению, и также заменяя входящий МАР-поток потоком Пуассона. Разброс оценок, полученных таким образом, показан на рис. 4.13.

Вполне ожидаемо, аппроксимация входящего потока ведет к очень серьезным отклонениям, особенно вероятности доставки и оценки размера последней системы. В обоих случаях чаще всего получается завышенная оценка, особенно для сетей малого размера. Отметим, что разброс оценок суммарной задержки не так велик. Также отметим, что разброс оценок в случае, когда аппроксимировались только выходящие потоки, оказался относительно небольшим.

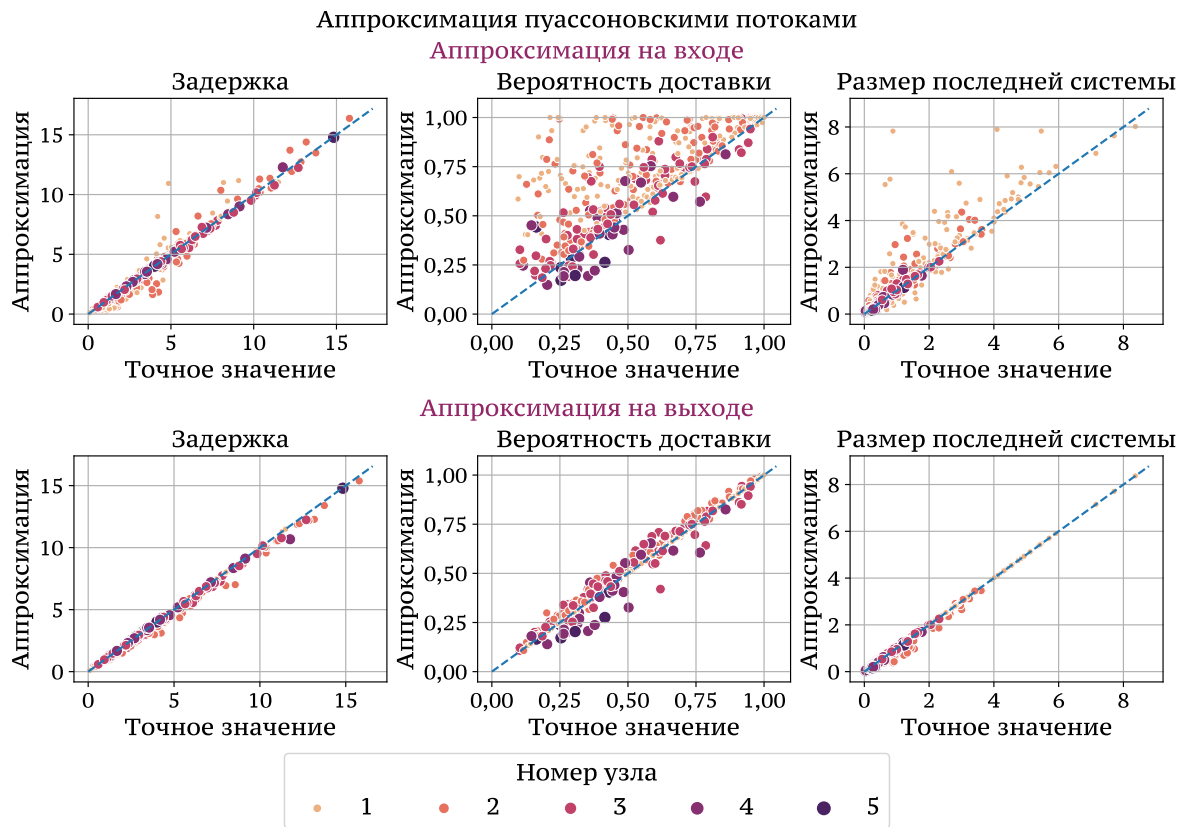


Рисунок 4.13 — Разброс оценок, полученных аппроксимацией потоков экспоненциальным распределением

Далее было проведено сравнение с оценками, полученными аппроксимацией потоков с помощью РН-распределений по среднему и коэффициенту вариации (см. рис. 4.14). Как и в случае с аппроксимацией потоками Пуассона, ошибки при аппроксимации входящего МАР-потока оказались существенно больше ошибок, полученных в случае аппроксимации только выходящих потоков. В то же время отметим, что разброс немного снизился по сравнению с аппроксимациями потоками Пуассона.

На рис. 4.15 показаны результаты, полученные аппроксимацией МАР-потоков РН-распределениями по трем моментам. В этом случае, как и ранее, разброс при аппроксимации входящего потока выше разброса в случае аппроксимации только выходящих потоков, однако он существенно меньше, чем тот, что был получен предыдущими методами. Это наблюдение говорит о важности учета коэффициента асимметрии и, в то же время, о том, что отказ от учета коэффициента корреляции входящего МАР-потока все же ведет к росту ошибок. Стоит отметить, что разброс оценок при аппроксимации только выходящих потоков очень мал.

Чтобы учесть корреляцию, использовался метод аппроксимации потоков новыми МАР-потоками, построенными по трем моментам и коэффициенту корреляции, который был описан в разделе 4.4.3. Разброс оценок показан на рис. 4.16. Полученный разброс мало отличается как в случае аппроксимации исходного МАР-потока, так и в случае аппроксимации только выходящих МАР-потоков.

Отметим, что последний результат может быть не совсем точным, так как для генерации МАР-потоков и их аппроксимации использовались одинаковые алгоритмы. Другими

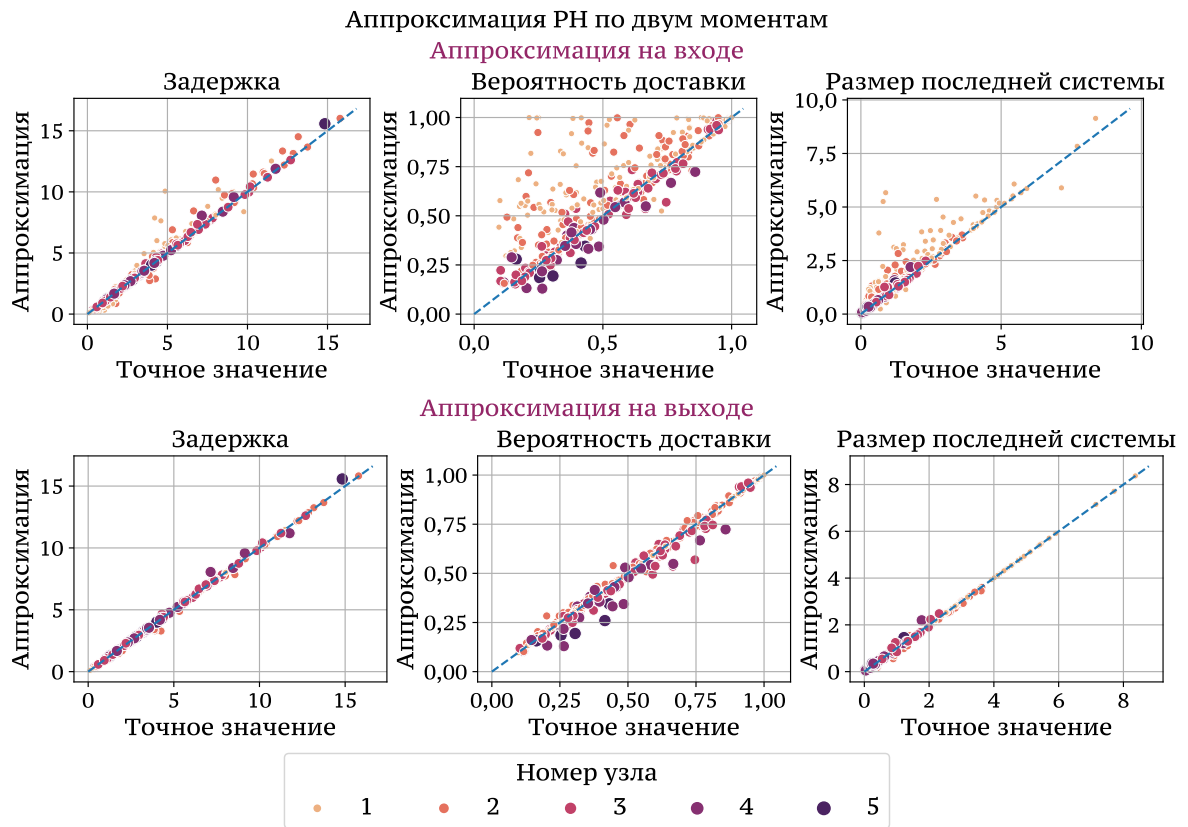


Рисунок 4.14 — Разброс оценок, полученных аппроксимацией потоков РН-распределениями по двум моментам

словами, исходные МАР-потоки имели достаточно «хорошую» структуру, чтобы быть точно аппроксимируемыми. Вероятно, ошибка при аппроксимации исходных МАР-потоков может увеличиться на более реалистичных данных (например, если исходные МАР-потоки обладают положительными коэффициентами корреляции с большими лагами или существенно иной структурой переходов в матрице  $D_0$ ).

На рис. 4.17 показан разброс оценок, полученных методом Монте-Карло. Для их получения в каждой сети моделировалось по 100 тысяч пакетов. Точность результатов оказывается наиболее высокой.

Общий разброс ошибок, полученных всеми перечисленными методами, показан на рис. 4.18 с помощью диаграмм типа ящички с усами. На каждом ящичке с усами средняя линия соответствует медиане ( $Q_2$ ), нижняя и верхняя границы прямоугольника — первый ( $Q_1$ ) и третий ( $Q_3$ ) квартили, а границы усов определяются, как наименьшее значение выборки, не меньшее  $Q_1 - 1.5(Q_3 - Q_1)$ , и наибольшее значение, не большее  $Q_3 + 1.5(Q_3 - Q_1)$ . Точки, лежащие над верхним усом — выбросы.

Из приведенных на рис. 4.18 данных можно сделать несколько выводов. Во-первых, ожидаемо, метод Монте-Карло дает наилучшую оценку, причем среднее не значительно выше медианы. Это говорит о том, что даже имеющиеся выбросы относительно невелики и не утягивают среднее значение вверх слишком сильно. Средняя погрешность по всем трем метрикам находится в пределах 1,5%. Во-вторых, аппроксимация входящего МАР дает существенно большую ошибку, особенно при использовании одного или двух моментов, что

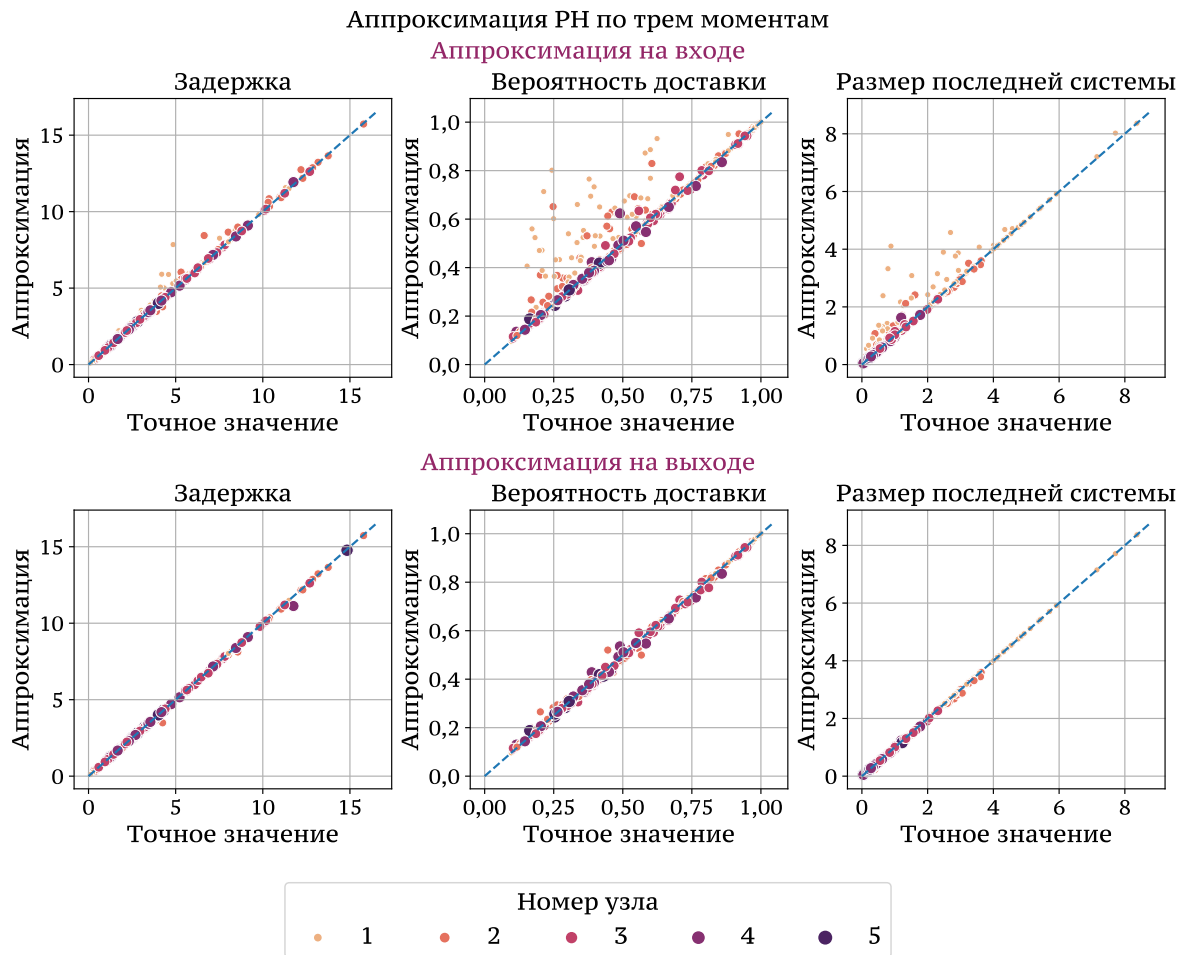


Рисунок 4.15 — Разброс оценок, полученных аппроксимацией потоков РН-распределениями по трем моментам

подтверждает важность учета коэффициента корреляции. В-третьих, ошибки в оценке суммарной задержки оказались невысокими даже при аппроксимации потоками Пуассона — меньше 10% в среднем для сетей длиной два и более узла. В-четвертых, вероятность доставки и средний размер последней системы удается оценить с медианной погрешностью менее 10% при аппроксимации МАР-потоков РН-распределениями по трем моментам и, естественно, МАР-потоками по трем моментам и коэффициенту корреляции. В-пятых, погрешности в оценке размеров системы и вероятности доставки становятся тем меньше, чем больше узлов в сети. Наконец, в-шестых, во всех случаях при аппроксимации потоков возникают редкие выбросы, которые очень сильно повышают среднее значение. Другими словами, в подавляющем большинстве случаев удастся получить оценки с небольшой погрешностью (например, около 5% для аппроксимации РН по трем моментам), но в редких случаях ошибка аппроксимации настолько велика, что средняя ошибка по всему множеству входных наборов достигает 20%.

Рост ошибки при аппроксимации входящего МАР-потока, высокая точность Монте-Карло и улучшение приближений по мере увеличения числа аппроксимируемых моментов были ожидаемы. В то же время, снижение ошибки при увеличении длины сети, а также практически идентичная точность при аппроксимации МАР-потоками и РН-распределениями по трем моментам, выглядят несколько неожиданно. Чтобы лучше понять причины этих

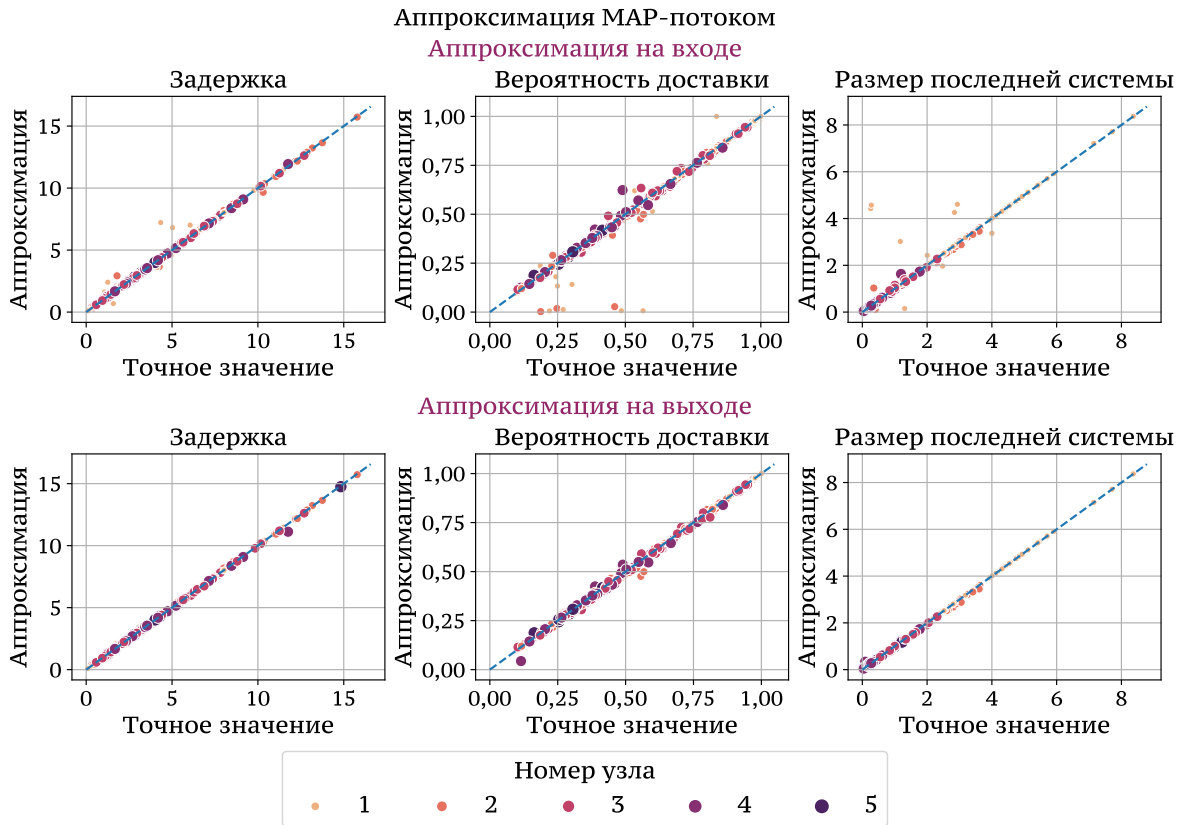


Рисунок 4.16 — Разброс оценок, полученных аппроксимацией МАР-потоков МАР-потоками по трем моментам и коэффициенту корреляции

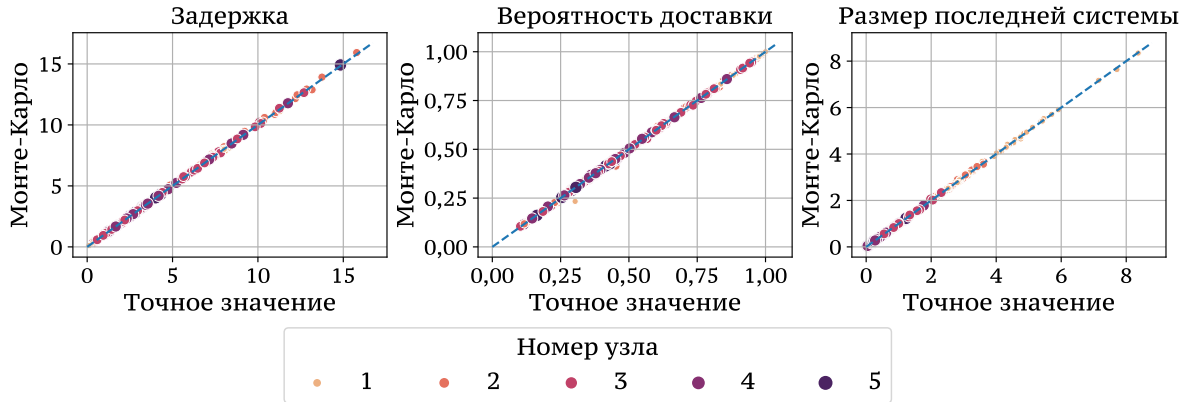
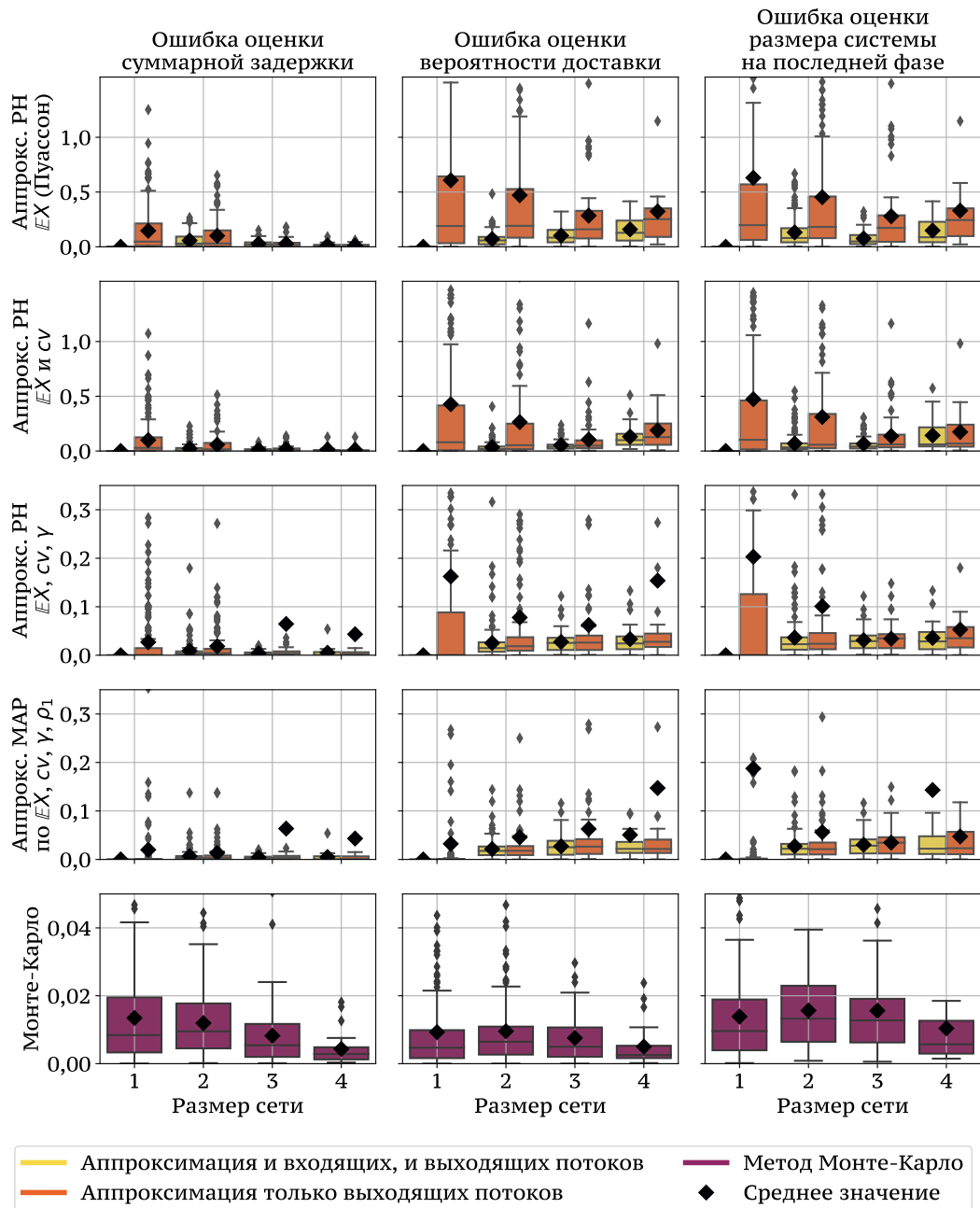


Рисунок 4.17 — Разброс оценок, полученных методом Монте-Карло

наблюдений, рассмотрим, как меняется коэффициент корреляции и коэффициент вариации в выходном потоке. На рис. 4.19 показан разброс коэффициентов корреляции и вариации во входном и выходном потоках. Во-первых, при положительной корреляции входящего потока, в выходящем потоке она снижается. Это правило нарушает только одно наблюдение. При нулевой корреляции во входящем потоке, в выходящем потоке она также снижается в большинстве случаев, хотя иногда (при высокой загрузке и положительной вариации и обслуживания, и входящего потока) может немного повыситься. Во-вторых, коэффициент вариации также меняется: если во входящем потоке вариация была больше единицы, то при относительно небольшой загрузке она уменьшается. Если же коэффициент вариации был меньше единицы (например, в случае распределений Эрланга), то вариация повышается,



На каждом ящике с усами средняя линия соответствует медиане ( $Q_2$ ), нижняя и верхняя границы прямоугольника – первый ( $Q_1$ ) и третий ( $Q_3$ ) квантили, а границы усов определяются, как наименьшее значение выборки, не меньше  $Q_1 - 1.5(Q_3 - Q_1)$ , и наибольшее значение, не большее  $Q_3 + 1.5(Q_3 - Q_1)$ . Точки, лежащие над верхним усом - выбросы.

Рисунок 4.18 — Точность оценок, полученных с помощью методов аппроксимации потоков и метода Монте-Карло

причем тем больше, чем выше загрузка. Такую связь с загрузкой можно объяснить тем, что когда поступления происходят чаще обслуживания (система насыщена), на выходе наблюдается скорее распределение времени обслуживания, чем распределение интервалов между поступлениями.

Еще более интересно отметить, что на последующих узлах сети изменения коэффициентов корреляции и вариации существенно снижаются. Результаты, вычисленные для первых

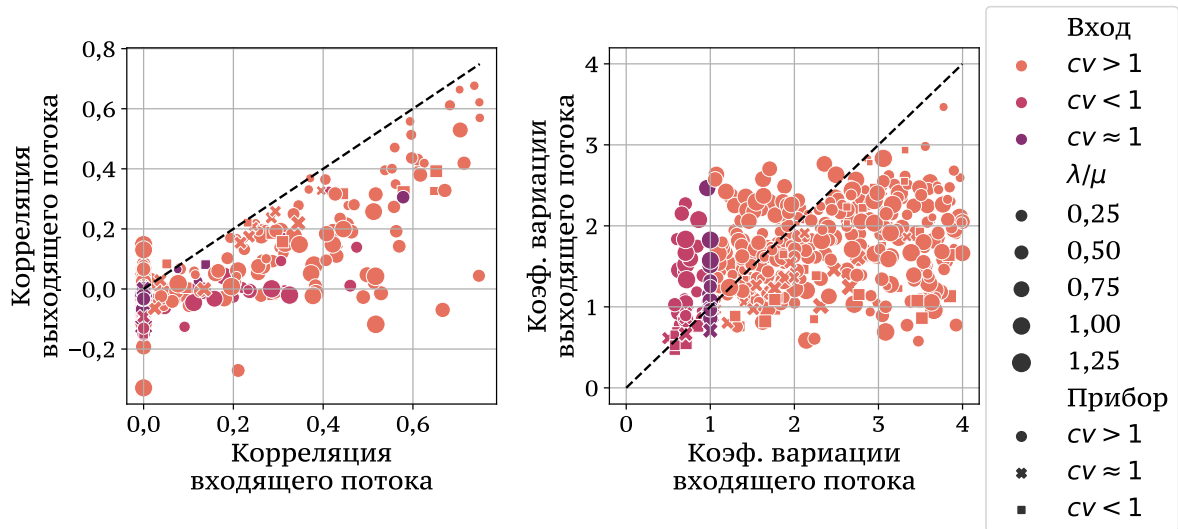


Рисунок 4.19 — Изменение коэффициентов корреляции и вариации в выходном потоке для системы MAP/PH/1/M

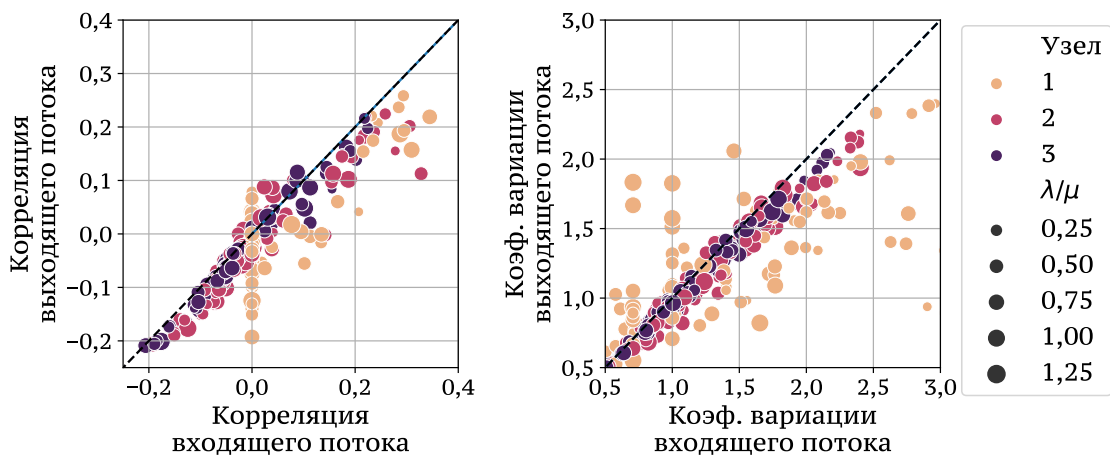


Рисунок 4.20 — Изменение коэффициентов корреляции и вариации в выходящих потоках на разных узлах сети

трех станций, показаны на рис. 4.20. Коэффициент корреляции также имеет тенденцию к снижению, однако его изменение становится тем меньше, чем дальше от начала сети находится узел. Коэффициент вариации также меняется тем меньше, чем дальше узел. Это позволяет понять, почему ошибки при аппроксимации выходящих потоков уменьшаются с ростом длины сети: ошибка при аппроксимации потоков на выходе с первого и второго узла может быть большой, однако далее она снижается, и общая ошибка растет медленнее по мере увеличения длины сети.

Так как наиболее точные результаты из всех методов аппроксимации были получены для аппроксимации выходящих потоков PH-распределениями по трем моментам и MAP-потоками, также была исследована точность аппроксимации относительно метода Монте-Карло. Разброс оценок для всех (и простых, и сложных) входных данных показан на рис. 4.21.

Судя по полученным результатам, аппроксимацию PH-распределениями по трем моментам можно использовать для получения приближенных оценок характеристик тандемной сети.



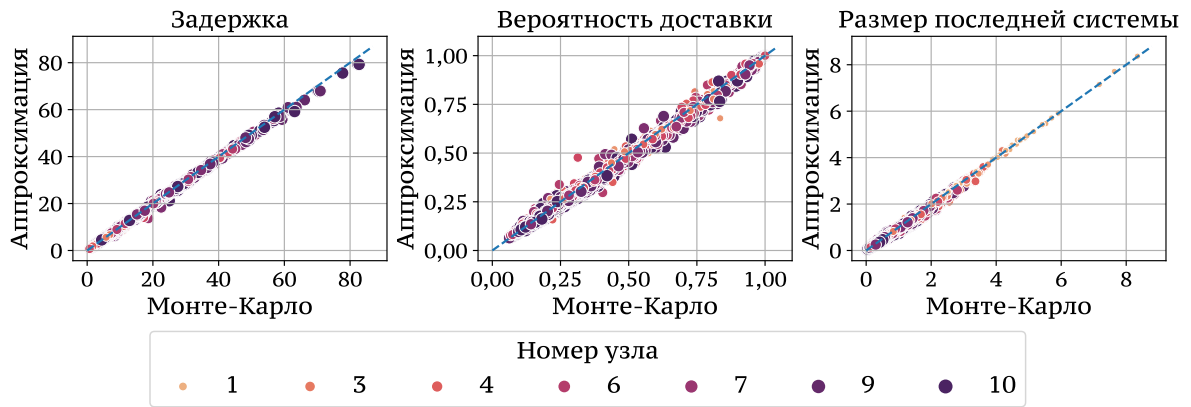


Рисунок 4.21 — Сравнение оценок, полученных методами Монте-Карло и аппроксимацией только выходящих потоков РН-распределениями по трем моментам

### Скорость вычислений и сходимость метода Монте-Карло

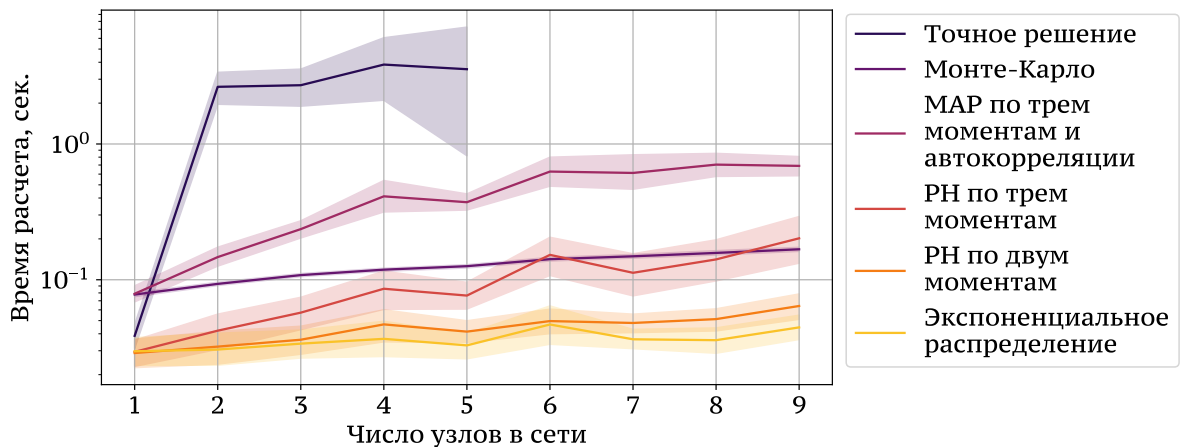


Рисунок 4.22 — Длительность расчета точного решения, а также расчетов с помощью различных методов аппроксимации выходящих потоков и метода Монте-Карло

Для всех методов аппроксимации, точного решения и метода Монте-Карло были проведены измерения времени, необходимого для получения оценок, результаты показаны на рис. 4.22. Можно отметить, во-первых, что точное решение требует больше всего времени. Это связано с необходимостью обращать матрицы и решать системы линейных уравнений очень высокого порядка (размеры матриц были ограничены 8000-ми строк/столбцов). Во-вторых, аппроксимация MAP-потоками также работает долго. Причина — необходимость решать задачу оптимизации для поиска элементов матрицы  $D_1$  (поиск минимума и максимума допустимых коэффициентов корреляции для заданной матрицы  $D_0$ ). В-третьих, остальные методы аппроксимации работают быстрее метода Монте-Карло. Так, если для сетей с 9 узлами метод Монте-Карло требует порядка 200–300 мс., то аппроксимация РН-распределениями по трем моментам — около 150 мс., а аппроксимация РН-распределениями по одному или двум моментам — менее 50 мс. Отметим, что все численные эксперименты проводились на рабочей станции с процессором Intel Core i7-9700К под управлением ОС Ubuntu

Linux. Код имитационной модели компилировался с помощью компилятора GCC с уровнем оптимизации O3. Расчет методом аппроксимации потоков был реализован на Python с использованием библиотек NumPy и SciPy.

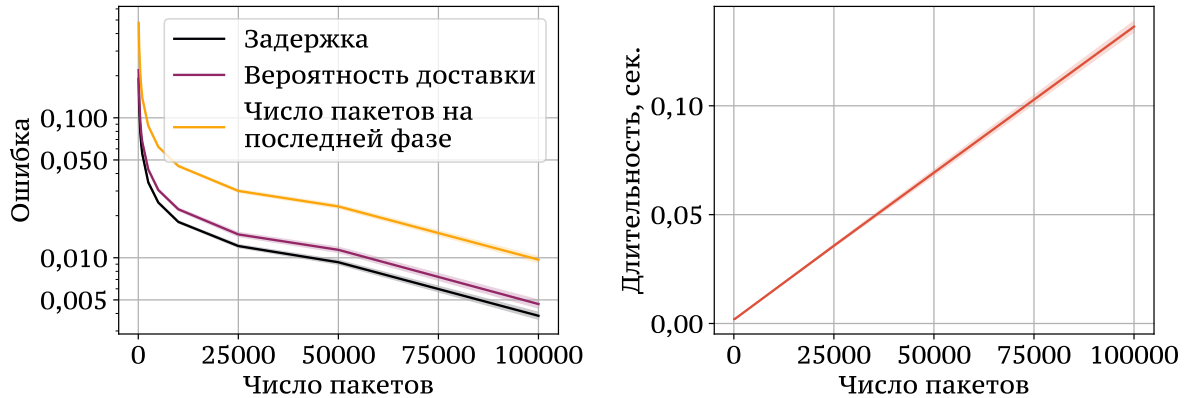


Рисунок 4.23 — Сходимость метода Монте-Карло и скорость выполнения расчетов

Наконец, отметим, что 100 тыс. пакетов для метода Монте-Карло является даже избыточным. На рис. 4.23 показано, как меняется точность результатов от числа промоделированных пакетов, а также то, как меняется время, расходуемое на выполнение метода. Так, для получения точности 5 % оказывается достаточно промоделировать около 50 тыс. пакетов. Ограничить число пакетов представляется разумным в свете того, что необходимое время зависит от числа пакетов почти линейно.

#### 4.7 Численный расчет характеристик многошаговой беспроводной сети

В численном эксперименте были исследованы характеристики многошаговых беспроводных сетей с помощью метода, описанного в разделе 4.1. Предполагалось, что сеть работает в ad-hoc режиме по стандарту IEEE 802.11 DCF без RTS/CTS. Параметры настройки протокола приведены в табл. 7. Беспроводная сеть моделировалась в системе моделирования NS-3, в каждом запуске моделировалась передача 500'000 пакетов.

Входящий трафик моделировался пуассоновскими потоками с четырьмя интенсивностями  $\lambda = 100, 200, 500, 750, 1000$ . Учитывая, что средний размер пакетов, поступающих от пользователей, был равен 11200 бит, эти потоки соответствовали трафику 1,2, 2,4, 6, 9 и 12 мегабит в секунду.

##### 4.7.1 Характеристики каналов калибровочной сети

Для получения РН-распределений, моделирующих время передачи пакетов в каналах, с помощью имитационной модели в NS-3 была собрана статистика длительностей передачи

Параметр	Значение	Единицы
Заголовок PHY	128	бит
Заголовок MAC DATA	208	бит
Кадр MAC ACK	96	бит
Слот ( $\sigma$ )	9	мкс
DIFS	34	мкс
SIFS	16	мкс
CWmin	16	-
CWmax	1024	-
Скорость канала	54	Мбит/с
Заголовок IP	160	бит
Средний размер пакета данных ( $\bar{\xi}$ )	11200	бит
Минимальная интенсивность трафика ( $B_{\min}$ )	1,2	Мбит/с
Максимальная интенсивность трафика ( $B_{\max}$ )	12	Мбит/с

Таблица 7 — Параметры беспроводных каналов и трафика для эксперимента

пакетов в каждом из четырех каналов калибровочной сети, показанной на рис. 4.8. Характеристики длительностей передач пакетов для  $\lambda = 500$  показаны на рис. 4.24.

Можно видеть, что по мере удаления от источника коэффициент вариации падает, распределение становится все ближе к константе. Как можно видеть из рис. 4.25, это наблюдение справедливо практически для всех интенсивностей, кроме  $\lambda = 750$ .

С помощью метода восстановления РН-распределения по трем моментам, описанного в работе [111], были построены РН-распределения для каждого канала калибровочной сети и рассчитаны межконцевые задержки с помощью сети массового обслуживания  $M/PH_1/1/N \rightarrow \bullet/PH_2/1/N \rightarrow \bullet/PH_3/1/N \rightarrow \bullet/PH_4/1/N$ . Эти значения сравнивались со значениями задержек, полученных из имитационной модели калибровочной сети в NS-3. Для того, чтобы исследовать целесообразность использования сложных моделей РН-распределений, задержки также были найдены с помощью простой модели типа  $M/M_1/1/N \rightarrow \dots \rightarrow \bullet/M_4/1/N$ , в которой обслуживание моделировалось экспоненциальными распределениями, которые были найдены по средним длительностям передач пакетов в каналах. Результаты сравнения приведены на рис. 4.26.

Для интенсивностей  $\lambda = 100, 200, 1000$  ошибки не превышают 10 %, причем для  $\lambda = 100, 200$  тандемы с РН-распределениями времени обслуживания оказываются в полтора раза точнее, а для  $\lambda = 1000$  немного более точный результат дают экспоненциальные распределения. Для  $\lambda = 500$  ошибка составила 15 % для экспоненциальных распределений и 24 % для РН-распределений. Что касается  $\lambda = 750$ , ошибка достигла почти 96 %. Возможная причина — особенности работы модели NS-3. Так, в выборке длительностей передач оказывались пакеты, которые доставлялись более секунды, хотя они должны были бы быть отброшены из-за исчерпания попыток передач. В дальнейшем рассматривались только  $\lambda = 100, 200, 500, 1000$ .

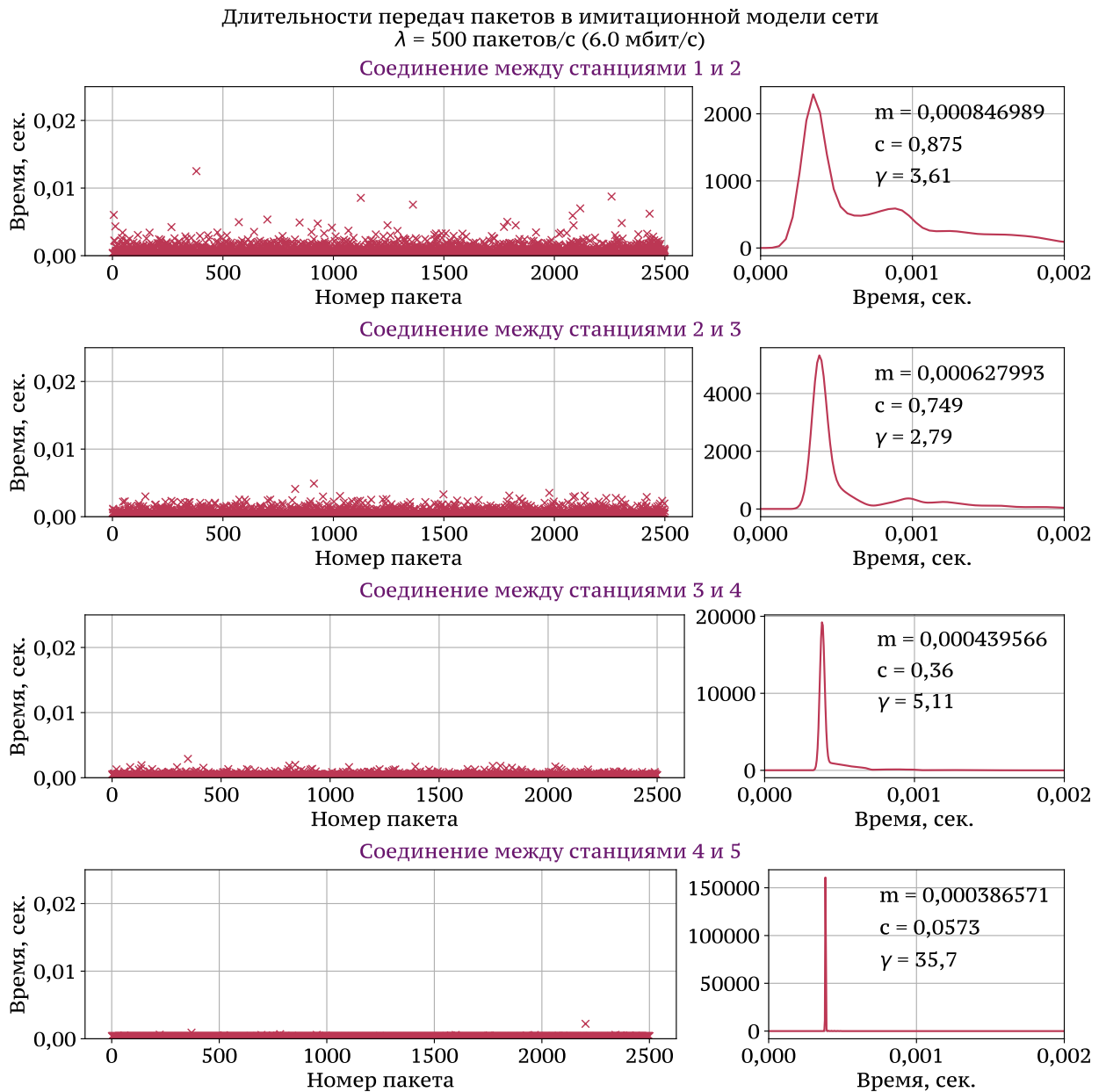


Рисунок 4.24 — Распределение длительностей передач пакетов в каналах калибровочной сети для  $\lambda = 500$

#### 4.7.2 Расчет межконцевых задержек в сети произвольного размера

Для расчета межконцевых задержек в сети, состоящей из 10 станций, использовалась методика, описанная в разделе 4.5: первый и два последних канала моделировались распределениями, полученными для каналов  $S_1 \rightarrow S_2$ ,  $S_3 \rightarrow S_4$ ,  $S_4 \rightarrow S_5$  калибровочной сети, а все промежуточные — с помощью распределения, построенного для канала  $S_2 \rightarrow S_3$  калибровочной сети (см. рис. 4.9). Как и ранее, использовались две модели сетей массового обслуживания: с PH-распределениями и экспоненциальными распределениями. Результаты сравнения с данными, полученными из модели NS-3, показаны на рис. 4.27.

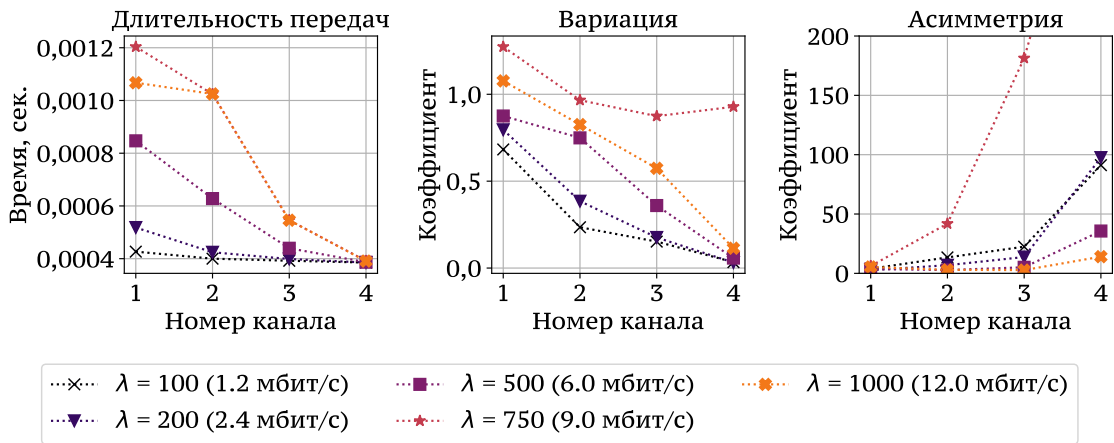


Рисунок 4.25 — Средние длительности передач, коэффициенты вариации и асимметрии для каналов калибровочной сети

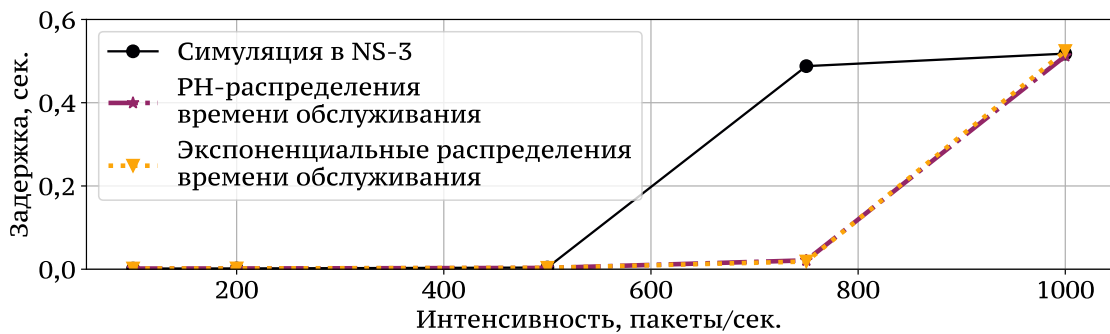


Рисунок 4.26 — Межконцевые задержки в калибровочной сети

Полученные результаты показывают, что обе модели дают неплохое приближение межконцевой задержки, причем сеть массового обслуживания с PH-распределениями точнее аппроксимирует беспроводную сеть. Стоит отметить, что для расчета характеристик сети массового обслуживания требовалось значительно меньше времени, чем на выполнение имитационной модели в NS-3. В то же время, ошибка растет по мере роста размера сети, особенно при высокой интенсивности входящего потока. Причем если в сети массового обслуживания рост задержки практически линеен, то в беспроводной сети скорость роста задержки снижается по мере роста размера сети.

Для того, чтобы объяснить это наблюдение, на рис. 4.28 показаны характеристики каналов беспроводной сети, состоящей из 10 станций. Можно видеть, что первые каналы испытывают наибольшую нагрузку — средние длительности передач в них самые большие, как и потери пакетов. По мере передачи трафика далее, часть пакетов теряется и на вход следующих станций поступает все более разреженный трафик, в результате чего конкуренция снижается и скорость передачи возрастает. Таким образом, из-за того, что для моделирования промежуточных каналов использовалось распределение времени передачи канала  $S_2 \rightarrow S_3$  калибровочной сети, была введена систематическая ошибка.

Можно немного изменить методику выбора распределений, чтобы лучше учесть этот эффект просеивания трафика по мере удаления от источника. Будем моделировать первые четыре канала точно теми же распределениями, что были получены для калибровочной сети,

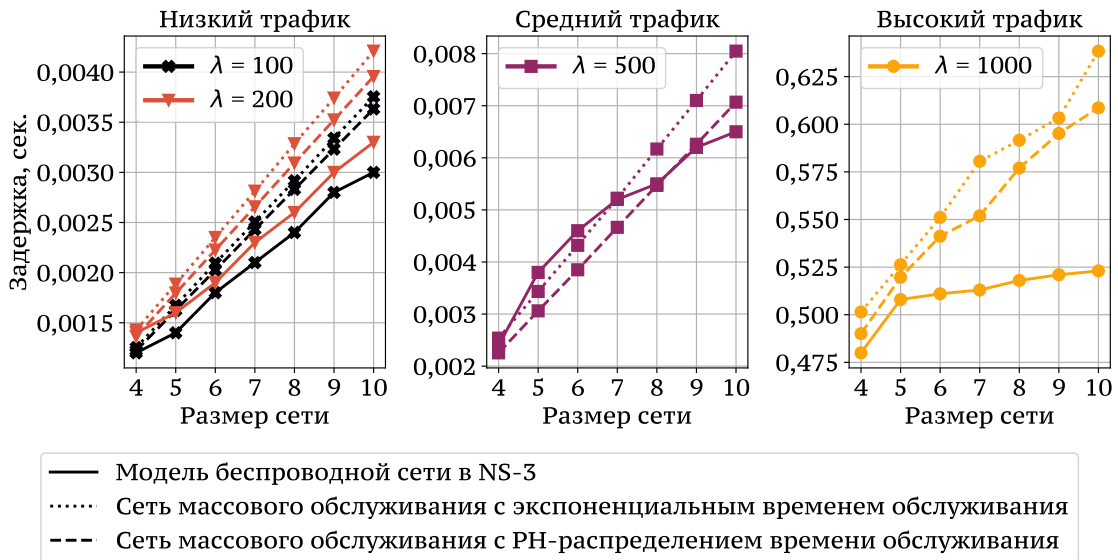


Рисунок 4.27 — Межконцевые задержки в сети из 10 станций, рассчитанные с помощью оригинальной методики

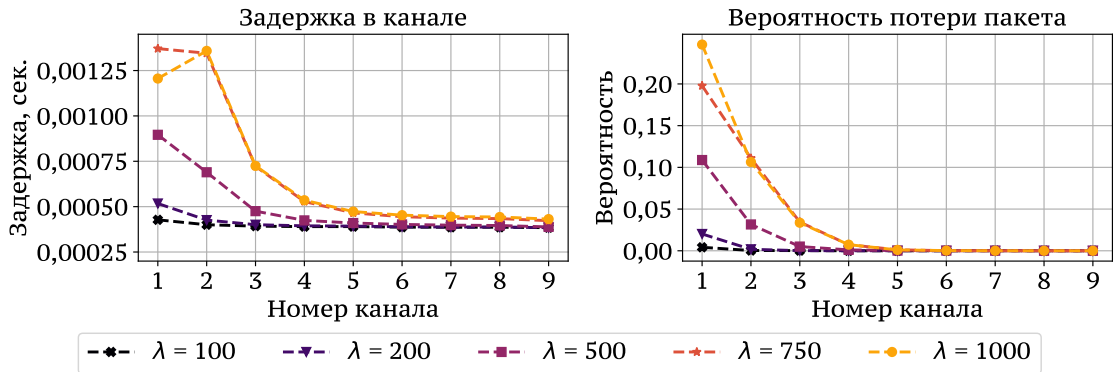


Рисунок 4.28 — Характеристики распределений длительностей передачи пакетов в каналах беспроводной сети из 10 станций

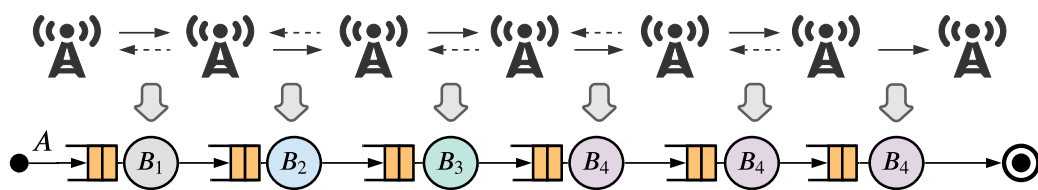


Рисунок 4.29 — Улучшенная схема моделирования беспроводной сети с произвольным числом станций

а все последующие каналы — распределением для канала  $S_4 \rightarrow S_5$ . Новая схема моделирования показана на рис. 4.29.

С помощью обновленной схемы расчета были получены более точные оценки, используя, как и ранее, РН-распределения и экспоненциальные распределения времени передачи пакетов. Результаты показаны на рис. 4.30. Можно отметить, что теперь для  $\lambda = 500$  сеть с экспоненциальными распределениями дала даже лучший результат, чем сеть с РН-распределениями. В остальных случаях использование РН-распределений по-прежнему дает более высокую точность.

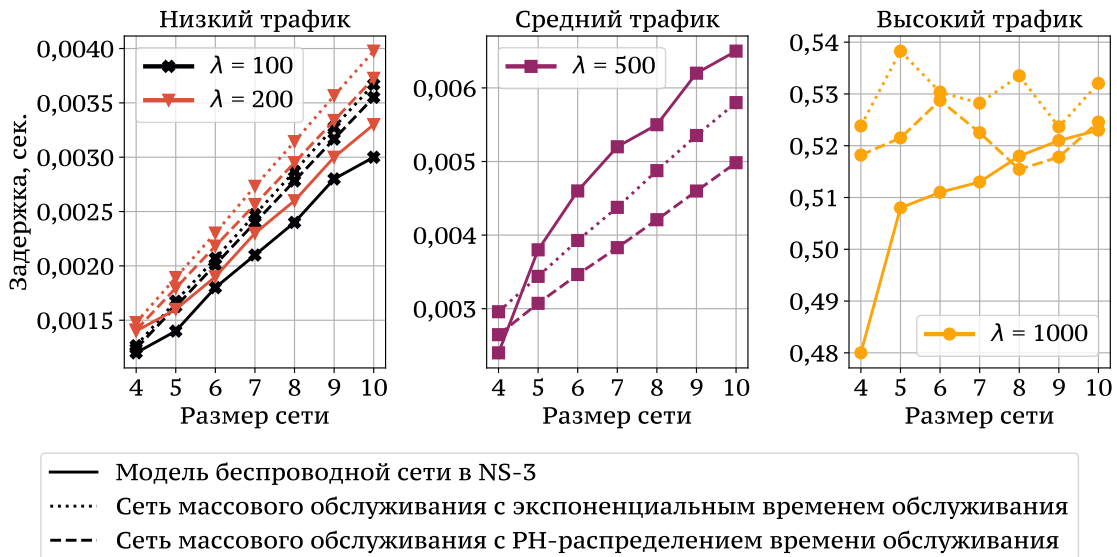


Рисунок 4.30 — Межконцевые задержки в сети из 10 станций, рассчитанные с помощью улучшенной методики

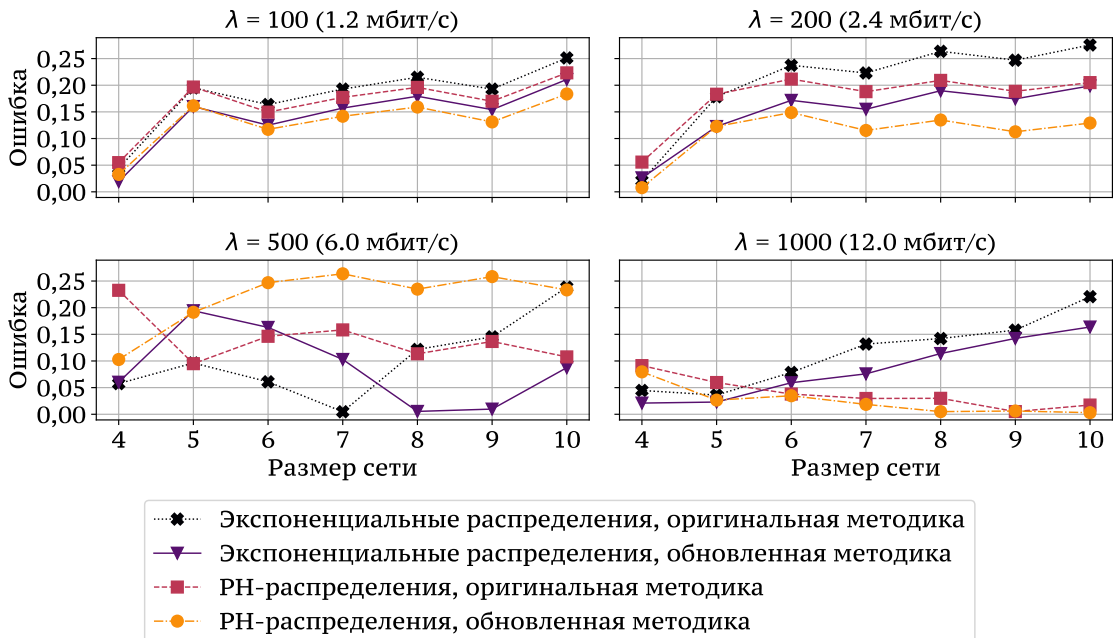


Рисунок 4.31 — Ошибки оценок межконцевых задержек беспроводной сети с помощью моделей сетей массового обслуживания

Ошибки, полученные с помощью исходной и улучшенной методики, приведены на рис. 4.31. Обновленная методика дает более точное приближение межконцевых задержек, независимо от того, используется ли для моделирования длительностей передач пакетов экспоненциальное или PH-распределение. Ошибка, полученная с помощью обновленной методики, не превосходит 20 %, кроме тандема с PH-распределениями при  $\lambda = 500$ . Во всех остальных случаях тандем с PH-распределениями, построенный по обновленной методике, дает самое лучшее приближение. При  $\lambda = 500$  самые точные приближения были получены с помощью тандема с экспоненциальными распределениями, построенного по обновленной методике, и тандема с PH-распределениями, построенного по оригинальной методике.

## 4.8 Заключение

В главе были представлены следующие результаты.

1. Предложен метод нахождения приближенных значений характеристик открытых сетей массового обслуживания с узлами MАР/РН/1/Ν с помощью аппроксимации потоков обслуженных пакетов РН-распределениями и MАР-потоками меньшего порядка. Приведены результаты численного исследования метода, показывающие высокую точность при использовании РН-распределений, построенных по трем моментам исходного потока.
2. Предложена методика построения и выбора распределений для моделирования многошаговых беспроводных сетей IEEE 802.11 произвольного размера с помощью тандемных сетей массового обслуживания.
3. Представлены результаты численного исследования предложенной методики моделирования многошаговых беспроводных сетей, показывающие высокую точность получаемых оценок межконцевых задержек.



## Глава 5. Разработка и экспериментальное внедрение системы радиочастотной идентификации

Важная задача, возникающая при построении системы радиочастотной идентификации транспортных средств — необходимость объединять в реальном времени данные со множества считывателей, которые могут быть размещены на большой территории. Для решения этой задачи была разработана распределенная платформа для интеграции и управления RFID-считывателями. Платформа позволяет интегрировать разрозненные считыватели в единую масштабируемую систему обработки данных.

В 2014–2015 годах был проведён эксперимент в городе Казань, в ходе которого было проведено испытание разработанной платформы и RFID-считывателей для идентификации меток, размещённых на рейсовых автобусах. Весной 2020 года совместно с АО «Микрон» в рамках НИР были проведены испытания на полигоне в городе Казань для проверки системы при идентификации меток на автомобилях, движущихся со скоростями до 170 км/ч. Все испытания завершились успешно. Летом 2021 года начались испытания на платном участке центральной кольцевой автодороги (ЦКАД), которые к моменту написания диссертации еще не завершились.

В этой главе приводится описание разработанной платформы и проведенных экспериментальных исследований. В разделе 5.1 описывается структура платформы управления считывателями и ее основные компоненты, в разделе 5.2 — разработанные протоколы, а в разделе 5.3 — модули системы. Далее, в разделе 5.4, описывается аппаратная реализация RFID-считывателей и особенности программной реализации. Раздел 5.5 посвящен описанию экспериментальных исследований разработанных RFID-считывателей и системы управления. Заключение в разделе 5.6 завершает главу.

Результаты, представленные в главе, были опубликованы в работах, индексируемых Scopus/WoS [209;210]. По результатам эксперимента 2020 года был сделан доклад на форуме Kazan Digital Week 2020.

### 5.1 Архитектура системы управления считывателями

Радиочастотная идентификация транспорта может применяться для идентификации автомобилей в системах регистрации нарушений правил дорожного движения, для оплаты проезда по платным дорогам, доступа на закрытые территории, а также использоваться при поиске угнанных машин и розыске преступников. Для быстрой реализации различных приложений программное обеспечение должно предоставлять возможность простого добавления новых сервисов в систему, а также обеспечивать масштабируемость.

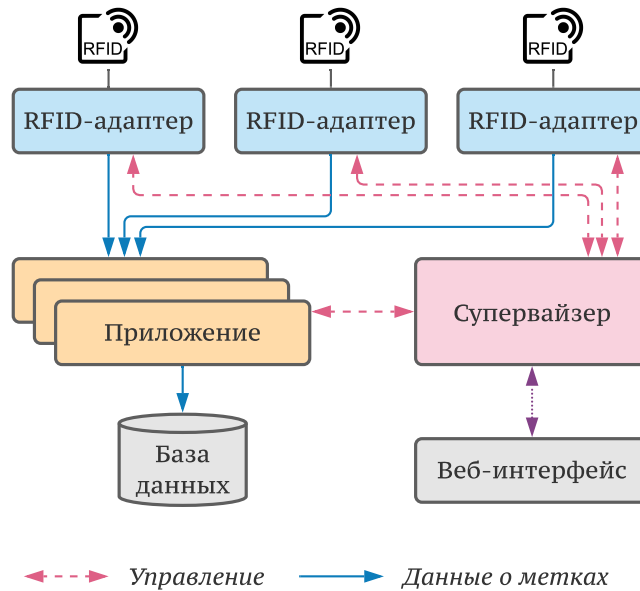


Рисунок 5.1 — Компоненты системы управления

Система управления считывателями имеет модульную структуру (см. рис. 5.1) и включает три типа модулей: супервайзеры, RFID-адаптеры и приложения. Супервайзеры (SVR) отвечают за хранение и управление конфигурацией системы. RFID-адаптеры управляют работой радиомодулей RFID-считывателей, реализуют чтение и запись меток. Приложения обрабатывают потоки прочитанных меток, поступающих от RFID-адаптеров и других приложений. Приложения могут использоваться для объединения потоков от нескольких источников, фильтрации или преобразования протоколов. Кроме перечисленных модулей, в состав системы входят инструменты администрирования — консоль и веб-интерфейс.

Каждый модуль системы поддерживает множество объектов, каждый из которых является либо параметром, либо процедурой. Обработка объектов ведётся по запросам, приходящим от пользователей или других модулей системы. Множество объектов полностью определяется типом модуля. Например, каждый RFID-адаптер поддерживает параметры «температура радио-модуля», «значение Q», «номер сессии» и процедуры «выключить радио-модуль», «включить радио-модуль». RFID-адаптер, специализированный на поддержке определенного считывателя, может предоставлять объекты, специфичные для конкретной модели считывателя. Каждый модуль получает свои настройки от супервайзера сразу после включения.

Для работы с объектом пользователь или модуль должны отправить запрос супервайзеру через один из поддерживаемых им протоколов управления (рассматриваются ниже). Супервайзер поддерживает множество собственных объектов, например — «число активных пользователей» или «статус подключений модулей», а также выполняет проксирование объектов всех подключенных модулей. Когда супервайзер получает запрос на объект, который не принадлежит ему самому, он перенаправляет запрос модулю, которому этот объект принадлежит, и отвечает клиенту, перенаправляя ответ, полученный от модуля.

В системе используется разделение потоков данных и управления: все управление производится через супервайзер, а данные передаются напрямую между RFID-адаптерами и

приложениями. Такое разделение позволяет, с одной стороны, снизить нагрузку на супервайзер и увеличить масштабируемость, а с другой — использовать более простые протоколы между компонентами. Так как основное назначение системы — непрерывное чтение меток проезжающих автомобилей, передача данных о прочитанных метках организована в виде потоков. Приложение, которое хочет получать данные о метках от RFID-адаптера или другого приложения должно подписаться на поток.

Пользователи могут настраивать систему и взаимодействовать с RFID-адаптерами и приложениями. В системе определены четыре уровня доступа: наблюдатели, операторы, администраторы и суперпользователи. Наблюдатели могут подписываться на потоки меток, но не могут записывать метки или менять настройки оборудования. Операторы имеют право настраивать RFID-адаптеры и выполнять операции чтения/записи меток, но не могут менять настройки остальных компонентов системы. Администраторы имеют право настраивать модули, но не могут записывать метки. Суперпользователи имеют права на любые действия с системой. Помимо уровней доступа пользователей с каждым компонентом связан специальный системный уровень доступа, который позволяет ему работать с определенным набором объектов других компонентов. Например, приложения могут читать, но не имеют права записывать параметры RFID-адаптеров.

Поскольку все модули системы взаимодействуют друг с другом по сети, физически они могут располагаться где угодно. Типичные примеры размещений модулей показаны на рис. 5.2.

Для обычного RFID-считывателя, построенного на базе среднего процессора (например, одноядерный ARM A7 или ARM A8), обладающего небольшим объемом оперативной памяти порядка 256–512 МБ и имеющего в составе один радиомодуль, достаточно установить супервайзер, RFID-адаптер, а также приложение для подключения к потоку прочитанных меток снаружи (TFPD) и модуль, предоставляющий доступ к настройке считывателя через веб-интерфейс или консоль. Такое размещение компонентов позволяет передавать данные обо всех прочитанных метках в центр обработки данных и, при необходимости, реализовывать операции чтения/записи через подключение внешних модулей и пользователей. Именно такая конфигурация считывателей использовалась во всех проведенных экспериментах.

Если считыватель имеет более мощное вычислительное ядро, на нём можно разместить несколько приложений. Например, на нём можно установить приложения для фильтрации прочитанных меток или нахождения соответствий с номерными знаками, распознанными камерами. Если считыватель включает несколько радиомодулей, то потребуются нескольких RFID-адаптеров. Хотя каждый компонент системы не требователен к вычислительным ресурсам по отдельности, для выполнения задач обработки потоков меток могут потребоваться дополнительные ресурсы. Желательно использовать 2-х или 4-х ядерный процессор ARM и 512 — 1024 МБ оперативной памяти.

В наиболее сложном случае система может работать в распределенном режиме, а различные ее компоненты размещаться на считывателях и серверах, при этом считыватели могут быть сделаны чрезвычайно «тонкими» — на них достаточно разместить RFID-адаптер, который будет подключаться по сети к супервайзеру, работающему на внешнем сервере

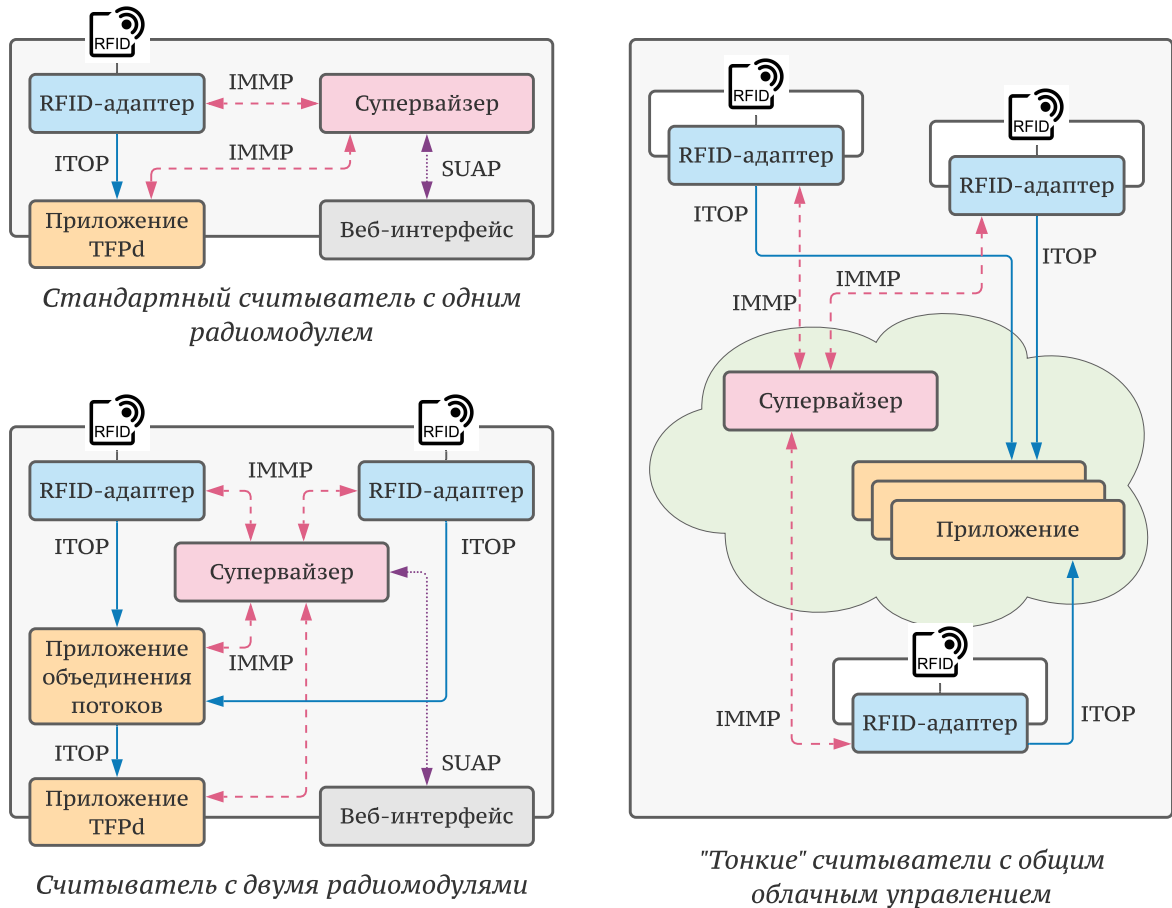


Рисунок 5.2 — Примеры размещения модулей системы

или в облаке. Приложения также могут работать на отдельных физических или виртуальных внешних серверах. В этом варианте считыватели могут быть реализованы на самых простых вычислительных компонентах, что позволит снизить их стоимость, а приложения и супервайзер выполняются в центре обработки данных на мощном вычислительном оборудовании, благодаря чему им можно поручить работу с большим числом считывателей. Такой вариант размещения наиболее гибок и позволяет построить распределенную систему, состоящую из сотен считывателей, каждый из которых может администрироваться из единой консоли.

## 5.2 Протоколы взаимодействия компонентов системы

В системе управления передача данных и управления разделены. Для управления системой были разработаны протоколы IMMP (Internal Modules Management Protocol) и SUAP (Simple User Access Protocol), а для передачи потоков меток и работы с радиомодулями — протоколы ITOP (Internal Tags Operation Protocol) и TFP (Tag Flow Protocol). Протоколы IMMP и ITOP используются для управления и передачи внутри системы, а протоколы SUAP и TFP — для работы с супервайзером со стороны интерфейсов управления и полу-

чения потоков прочитанных меток во внешние информационные системы. Все протоколы, используемые в системе, показаны на рис. 5.3.

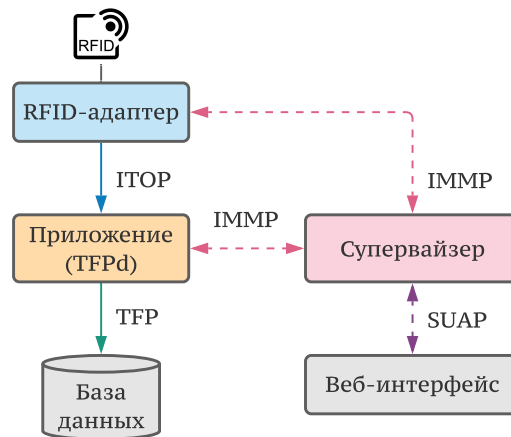


Рисунок 5.3 — Протоколы передачи данных и управления

Все протоколы имеют клиент–серверную архитектуру, соединение создаётся клиентом. Каждый запрос должен подтверждаться получателем с помощью специального сообщения Response, содержащего результат обработки запроса или код ошибки. Запросы упорядочиваются с помощью порядкового номера ReqSN (Request Sequence Number): клиент и сервер поддерживают счетчики, перед отправкой значение счетчика увеличивается и помещается в соответствующее поле в запросе. В ответе на запрос поле ReqSN должно совпадать со значением счетчика из запроса. Этот механизм позволяет реализовать асинхронную обработку запросов — отправителю запроса не нужно ждать ответа для того, чтобы передать новый запрос. При этом отправитель может быть уверен, что запросы будут обработаны в порядке их передачи: если получатель видит, что номер запроса меньше последнего обработанного, запрос будет проигнорирован. С каждым запросом также связано максимальное ожидание ответа, после которого отправитель считает, что запрос не был выполнен.

Формат сообщений, кодируемых с помощью ASCII-строк, аналогичен форматам, используемым в HTTP v1.1 и SIP.

### 5.2.1 IMMP — протокол управления модулями

Протокол IMMP работает поверх транспортного уровня TCP и реализует функции получения модулями своей начальной конфигурации от супервайзера, получения и установки значений параметров модулей, а также удалённый вызов процедур. В роли сервера всегда выступает супервайзер, а клиента — один из модулей системы.

В протоколе определены семь запросов: HELLO, ACK, BYE, GET, SET и CALL. Запрос HELLO отправляется клиентом для создания соединения. Если соединение может быть установлено, сервер (супервайзер) сначала передает ответ с кодом успеха (200), а затем — запрос ACK, в котором передает конфигурацию модуля и случайный ключ сессии (SKey, Session

Key), по которому в дальнейшем он будет авторизовывать запросы этого клиента. Запросы GET и SET используются для чтения и записи значений объектов-параметров, а запрос CALL — для удаленного вызова процедур. Запросы GET, SET и CALL могут передаваться как клиентом, так и сервером.

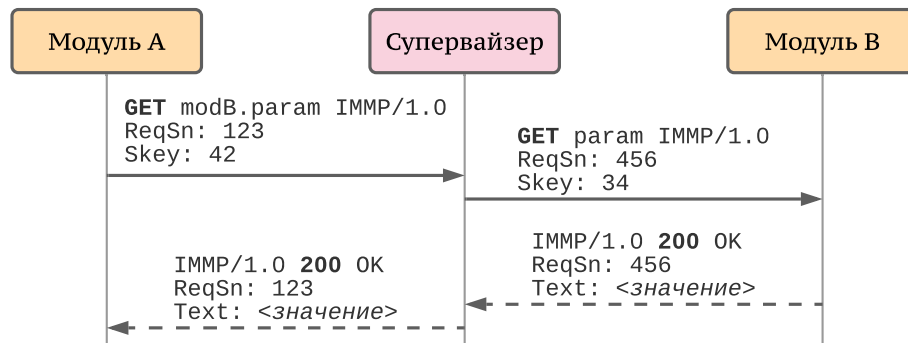


Рисунок 5.4 — Работа супервайзера в режиме IMMP-прокси

Если супервайзеру нужно выполнить какую-либо операцию над объектом другого модуля, он передает соответствующий запрос (GET, SET или CALL). Если же модулю А нужно выполнить действие над объектом модуля В, то супервайзер выступает в роли прокси-сервера: модуль А передает запрос супервайзеру, который тот передает модулю В, дожидается ответа и пересылает его обратно модулю А (см. рис. 5.4).

Пример соединения показан на рис. 5.5. Клиент передает запрос HELLO со своим именем, сервер отвечает кодом 200. Затем супервайзер готовит стартовую конфигурацию клиента и передает ее в запросе АСК, получение которого клиент должен подтвердить. Такое разделение ответов сервера возникает из-за того, что при высокой нагрузке на супервайзер подготовка стартовой конфигурации может занять время, которое может превышать время ожидания ответа на запрос HELLO. После получения ответа на HELLO клиент может ждать гораздо дольше получения своей конфигурации в АСК. Помимо начальной конфигурации, в АСК супервайзер передает ключ сессии SKey, который используется в дальнейшем обмене сообщениями. После обмена запросами HELLO и АСК начинается основная часть сессии. В ней клиент и сервер обмениваются запросами GET, SET и CALL. Завершает сессию в этом примере клиент отправкой запроса BYE.

### 5.2.2 SUAP — протокол подключения интерфейсов управления

Протокол SUAP представляет собой упрощенную версию IMMP, работающую через UDP. Он предназначен для подключения пользовательских интерфейсов к супервайзеру. В функции SUAP входят получение и установка значений параметров и удаленный вызов процедур. Протокол не содержит запроса АСК и не является симметричным — запросы GET, SET и CALL могут передаваться только от клиента серверу. Кроме этого, есть некоторые отличия в кодировании ответов: если в IMMP значение параметра в ответе на запрос GET

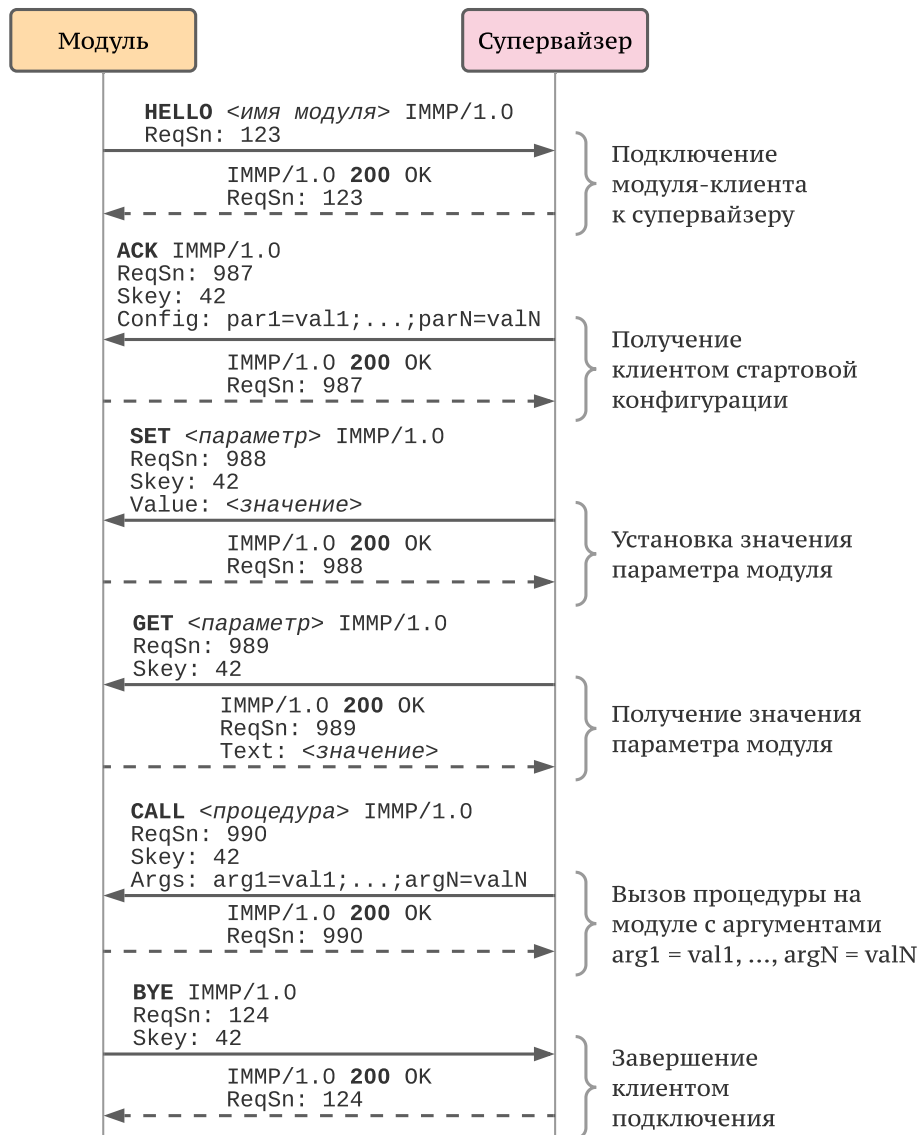


Рисунок 5.5 — Пример IMMP-сессии

передается в поле заголовка Text, то в SUAP значение параметра содержится в теле ответа, отделенном пустой строкой от последнего заголовка.

Аутентификация пользователей производится с помощью логина и пароля. Пароль может передаваться либо нешифрованным текстом, либо в виде хеша. В отличие от IMMP, вместо запроса HELLO клиент в начале сессии должен отправить запрос LOGIN, а для завершения сессии — запрос LOGOUT. Ключ сессии (SKey) используется для авторизации запросов клиента и передается супервайзером в ответе на запрос LOGIN. Пример сессии SUAP показан на рис. 5.6.

Отметим, что использование SKey для авторизации запросов не является безопасным: если злоумышленник сможет перехватить UDP-пакеты SUAP-сессии, то он сможет получить значение SKey и выполнять операции над системой с уровнем доступа скомпрометированного пользователя. Вопросы обеспечения защиты информации в системе управления требуют дальнейшей разработки, возможные решения — использование DTLS (Datagram Transport Layer Security), замена SUAP на HTTPS и реализация REST API на супервайзере или замена SUAP на gRPC с защитой SSL/TLS.

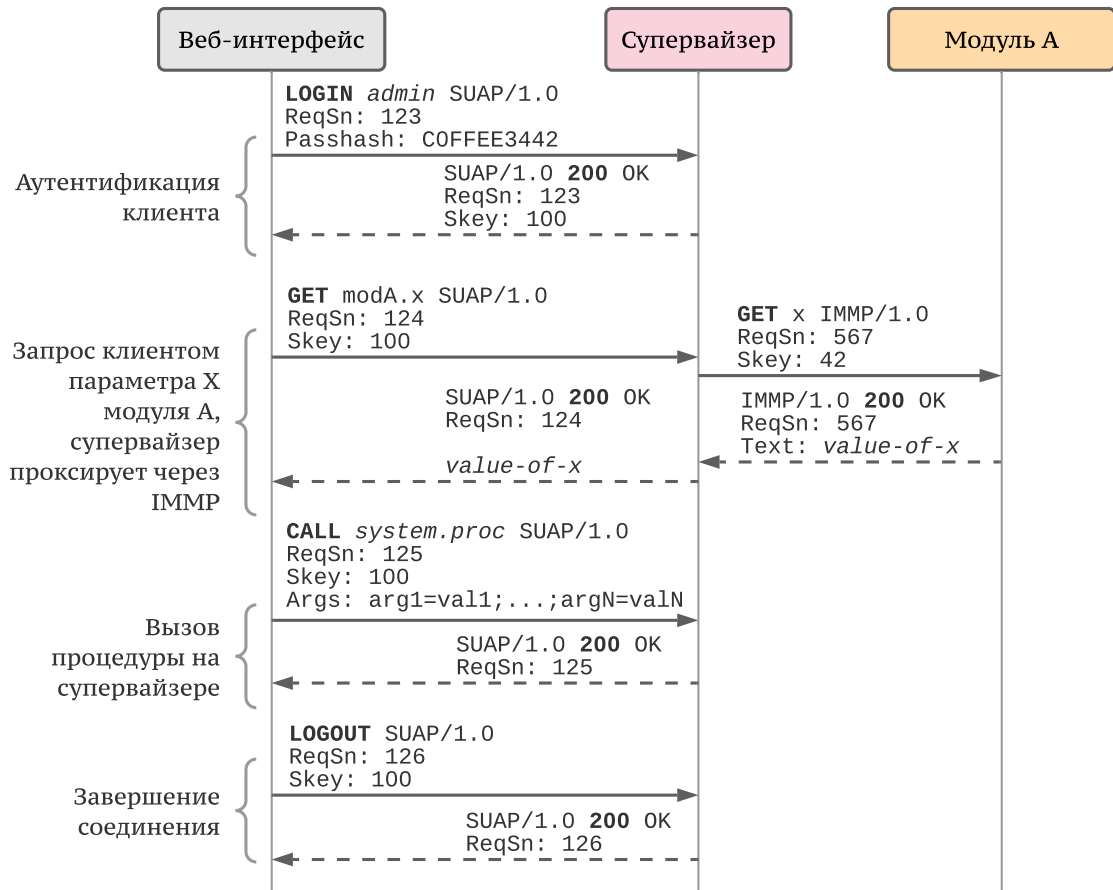


Рисунок 5.6 — Пример SUAP-сессии

### 5.2.3 ITOP — протокол работы с RFID-адаптерами

Протокол предназначен для связи между модулями системы (приложениями и RFID-адаптерами), желающими осуществлять действия над метками. Он позволяет читать и записывать память меток, а также подписываться на потоки считанных меток. Подписка может быть позднее отменена, также она завершается при закрытии соединения любой из сторон.

В протоколе определены семь запросов: HELLO, BYE, READMEM, WRITEMEM, WRITEEPC, SUBSCRIBE, UNSUBSCRIBE. Сервер может передать запрос BYE, остальные запросы передаются только клиентом. Запросы HELLO и BYE используются для создания и завершения соединений. SUBSCRIBE и UNSUBSCRIBE используются для создания и отмены подписки, а запросы READMEM, WRITEMEM и WRITEEPC — для чтения или записи данных на заданную метку.

Когда клиент хочет установить ITOP-соединение, он передает серверу запрос HELLO. В этот запрос клиент включает значение SKey, полученное ранее при создании IMMP-или SUAP-соединения. Значение SKey используется для авторизации сервером действий клиента. Например, администраторы и наблюдатели не имеют права записывать метки, а единственная разрешенная им операция — создание подписки на получение потока меток. Для авторизации сервер перед выполнением любой из операций READMEM, WRITEMEM,



WRITEEPC или SUBSCRIBE вызывает процедуру супервайзера «system.auth-tagop» с помощью CALL-запроса IMMP и передает в качестве ее аргументов ключ SKey клиента и тип операции. Если супервайзер подтвердит право клиента на запрошенное действие, то сервер выполнит операцию, в противном случае сообщит клиенту о ее недоступности. Для завершения соединения клиент или сервер отправляют запрос BYE. Пример сессии между веб-интерфейсом, пользователь которого ранее авторизовался с помощью SUAP и получил ключ SKey, и RFID-адаптером показан на рис. 5.7.

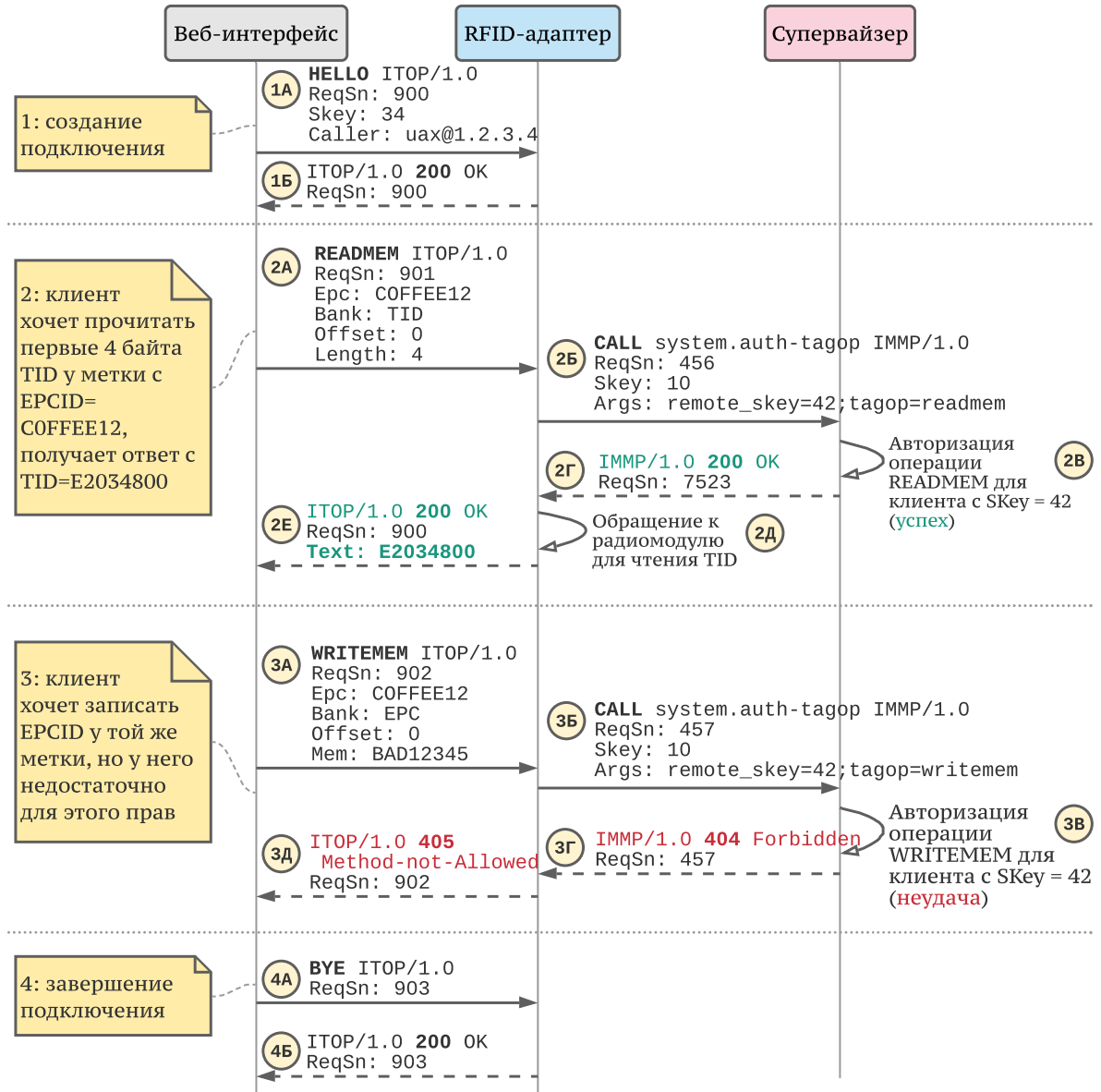


Рисунок 5.7 — Пример соединения ITOP между веб-интерфейсом и RFID-адаптером

Кроме чтения и записи меток по запросу протокол ITOP позволяет создавать подписки на считанные метки. Если клиент желает непрерывно получать информацию о прочитанных метках от сервера, он передает запрос SUBSCRIBE. Как только считывается (если сервер — RFID-адаптер) или принимается (если сервер — приложение) новая метка, сервер асинхронно передает специальное сообщение NOTIFY клиенту, содержащее информацию о метке. Когда клиент желает остановить получение потока, он отменяет подписку передачей запроса

UNSUBSCRIBE. Если клиент (например, веб-интерфейс) хочет читать метки только в течение заданного интервала, то он может добавить в запрос SUBSCRIBE заголовок Continuous со значением false и указать длительность чтения в миллисекундах в заголовке Duration. В этом случае клиенту не нужно явно завершать подписку отправкой UNSUBSCRIBE, она завершится автоматически по истечении заданного времени. Пример создания подписок двумя приложениями показан на рис. 5.8. В этом примере приложение А создает подписку с автоматическим продлением и завершает ее явно, а приложение В читает метки только в течение 0,5 секунд.

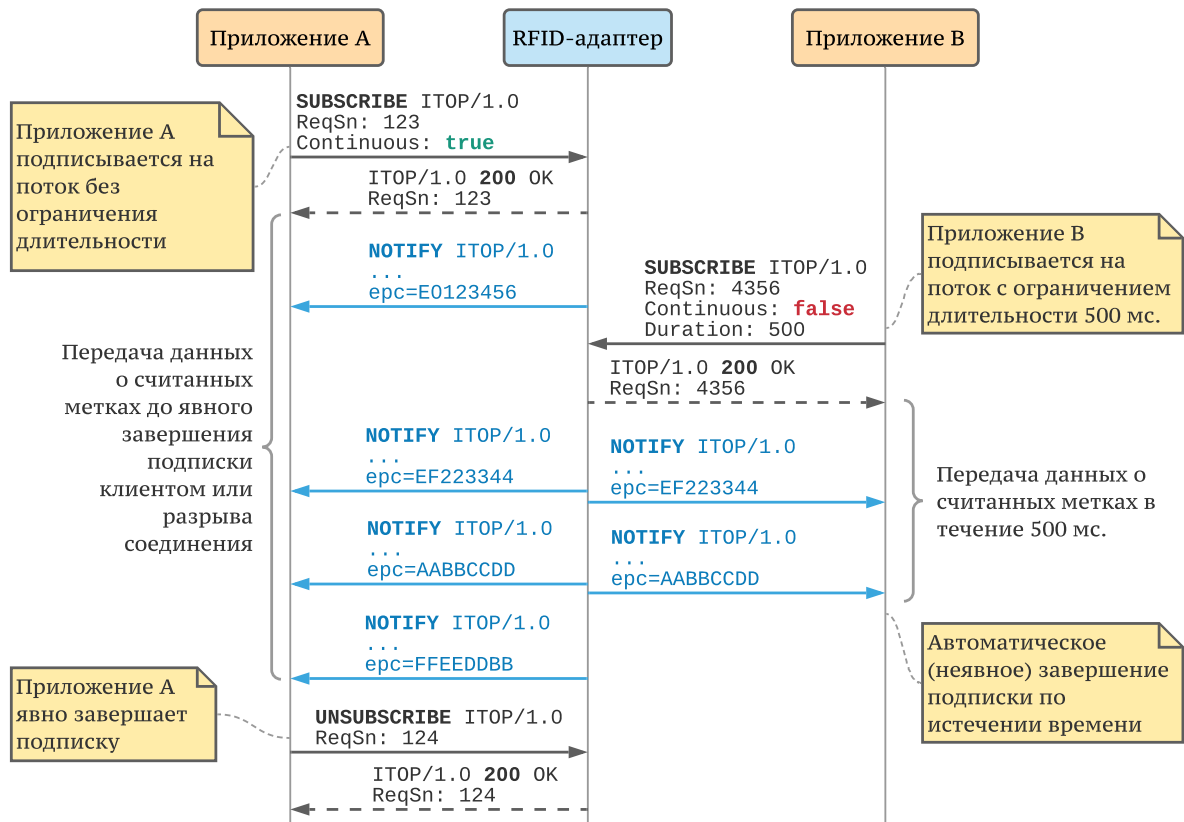


Рисунок 5.8 — Пример создания подписок в соединении ITOP

Тело сообщения NOTIFY может содержать следующие поля:

- epc: значение EPC прочитанной метки;
- tid: значение банка TID прочитанной метки;
- power: мощность принятого сигнала;
- antenna: номер антенны, на которой была считана метка;
- frequency: частота, на которой работал считыватель;
- count: число раз, которое метка была считана;
- um: содержимое банка памяти USER MEMORY;
- source: имя RFID-адаптера, считавшего метку;
- id: идентификатор метки.

Поле id может заполняться данными, специфичными для приложения, например — значением EPC или номером транспортного средства, полученным приложением с помощью

обращения к базе данных по TID или в результате декодирования EPCID. Поле `um` используется только при чтении банка пользовательской памяти, а поле `source` нужно только при использовании нескольких RFID-адаптеров. Если какие-то из полей не содержат данных, их можно опустить. Значение каждого поля передается в отдельной строке, общее число строк содержится в заголовке `Content-length`.

Если в роли сервера выступает не RFID-адаптер, а приложение, то оно может не поддерживать некоторые из операций, например `WRITEMEM`, `WRITEEPC` или `READMEM`. В этом случае в ответ на недоступные запросы оно будет передавать клиенту код 405 (`Method-not-Allowed`).

Так как радиомодуль не может одновременно выполнять несколько операций, ключевая сложность в реализации ИТОР-сервера для RFID-адаптера заключается в правильном мультиплексировании. Для этого потребовалось разработать алгоритм обработки запросов клиентов, позволяющий исключить блокировки и обеспечить такой уровень задержек, чтобы даже при наличии активных подписок другие пользователи могли выполнять свои операции чтения, записи или изменения настроек радиомодуля. Подробнее реализацию RFID-адаптера рассмотрим в разделе [5.3.2](#).

#### 5.2.4 TFP — протокол потокового чтения меток

Протокол TFP является упрощенной версией протокола ИТОР и предназначен для получения данных о метках из пользовательских интерфейсов или внешних клиентов. Например, TFP-клиенты могут записывать данные о прочитанных метках в базу данных или отправлять во внешнюю информационную систему по REST API или gRPC. Специальное приложение TFP Daemon (TFPD) выступает в роли моста между ИТОР и TFP: с одной стороны это приложение выступает в роли сервера протокола TFP, с другой — клиента ИТОР. В протоколе TFP поддерживается только механизм создания подписок, прочие функции (запись меток, чтение заданных областей памяти и пр.) он не реализует. Протокол поддерживает запросы `HELLO`, `BYE`, `SUBSCRIBE` и `UNSUBSCRIBE`, данные о метках передаются, как и в ИТОР, в сообщениях `NOTIFY`. Пример сессии TFP приведен на рис. [5.9](#).

Использование TFP-сервера позволяет снизить нагрузку на RFID-адаптер и на супервизор при одновременном подключении множества клиентов. Сервер TFP создает ИТОР-подписку только в том случае, если к нему подключен хотя бы один клиент и этот клиент подписан. Если подписанных клиентов несколько, TFP-серверу достаточно одной ИТОР-подписки, данные из которой он будет пересылать всем клиентам. Когда последний клиент отменяет подписку или отключается, TFP-сервер завершает ИТОР-подписку и, если других запросов по ИТОР-соединениям у RFID-адаптера нет, он может выключить радиомодуль до появления новых клиентов. Пример работы TFP-сервера с несколькими клиентами приведен на рис. [5.10](#).

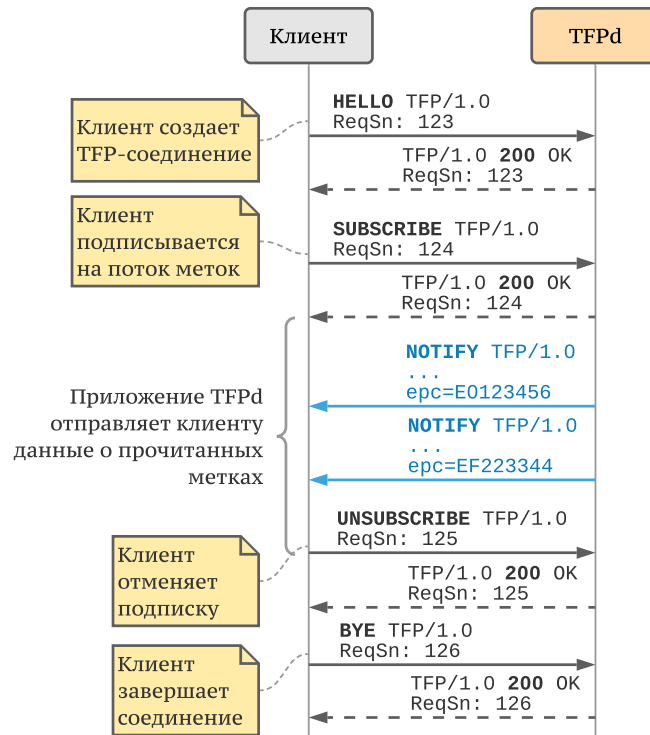


Рисунок 5.9 — Пример подключения клиента по протоколу TFP

### 5.3 Основные компоненты системы

Ключевыми компонентами системы являются супервизор (SVR), модуль управления радиомодулем RFID (G2RD), приложение TFPD и утилита UAX, выполняющая функции SUAP- и ITOP-клиента. Также в состав системы управления входят служебные компоненты — клиент TFPC, веб-интерфейс и консольные интерфейсы, которые используют UAX для отправки запросов SVR по SUAP и RFID-адаптеру по ITOP, а также специальная программа RRRD (Road RFID Reader Daemon), управляющая запуском остальных компонентов. Рассмотрим эти компоненты подробнее.

#### 5.3.1 Супервизор SVR

Супервизор управляет работой всех остальных модулей. Его основные задачи: аутентификация пользователей и модулей, передача запросов получения и установки параметров и вызова процедур на модулях системы, хранение и передача начальной конфигурации подключенным модулям, а также реализация логического интерфейса для взаимодействия с операционной системой (например, настройка сетевого интерфейса). SVR выполняет роль IMMP-сервера, при получении запросов относительно параметров других модулей он выполняет функции прокси-сервера (см. рис. 5.4). Также SVR выполняет роль SUAP-сервера для обслуживания запросов со стороны пользовательских интерфейсов.

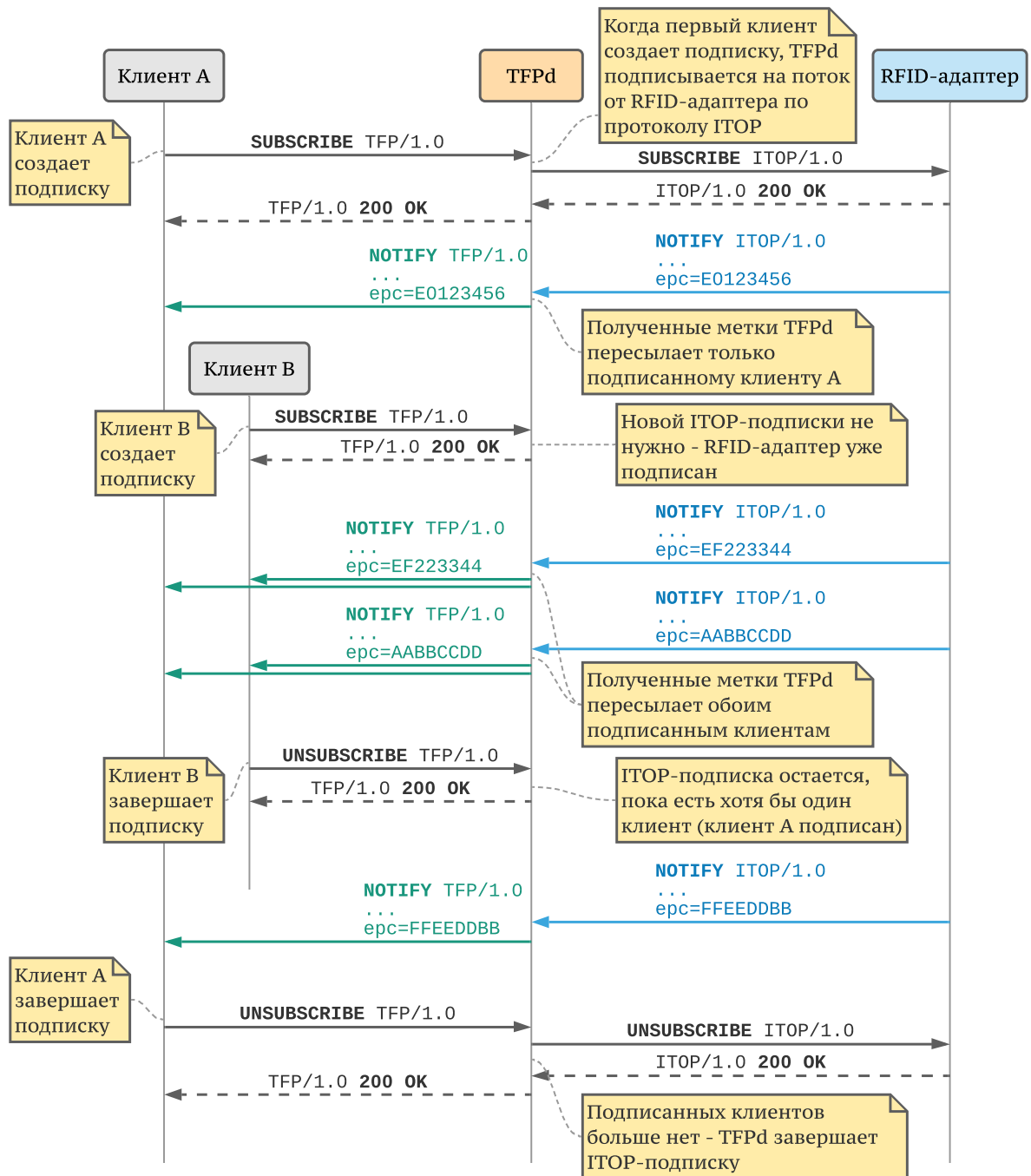


Рисунок 5.10 — Работа TFP-сервера при создании подписок от двух клиентов

Нагрузка на супервизор может быть достаточно высокой, особенно если в системе содержится несколько радиомодулей (и, соответственно, несколько модулей G2RD) и множество приложений. Для обеспечения масштабируемости SVR спроектирован многопоточным: все запросы, поступающие по IMMP или SUAP, поступают в очередь, откуда извлекаются и обрабатываются потоками-работчими. Число потоков-работчих можно настраивать. Взяв запрос, рабочий обрабатывает его синхронно. Если при этом требуется обратиться к другому модулю, он сам отправляет запрос этому модулю по IMMP и дожидается ответа. Из-за этой особенности реализации даже при небольшой загрузке целесообразно использовать несколько потоков-работчих (по-умолчанию, в системе используется 9 рабочих). Упрощенная программная архитектура супервизора показана на рис. 5.11.

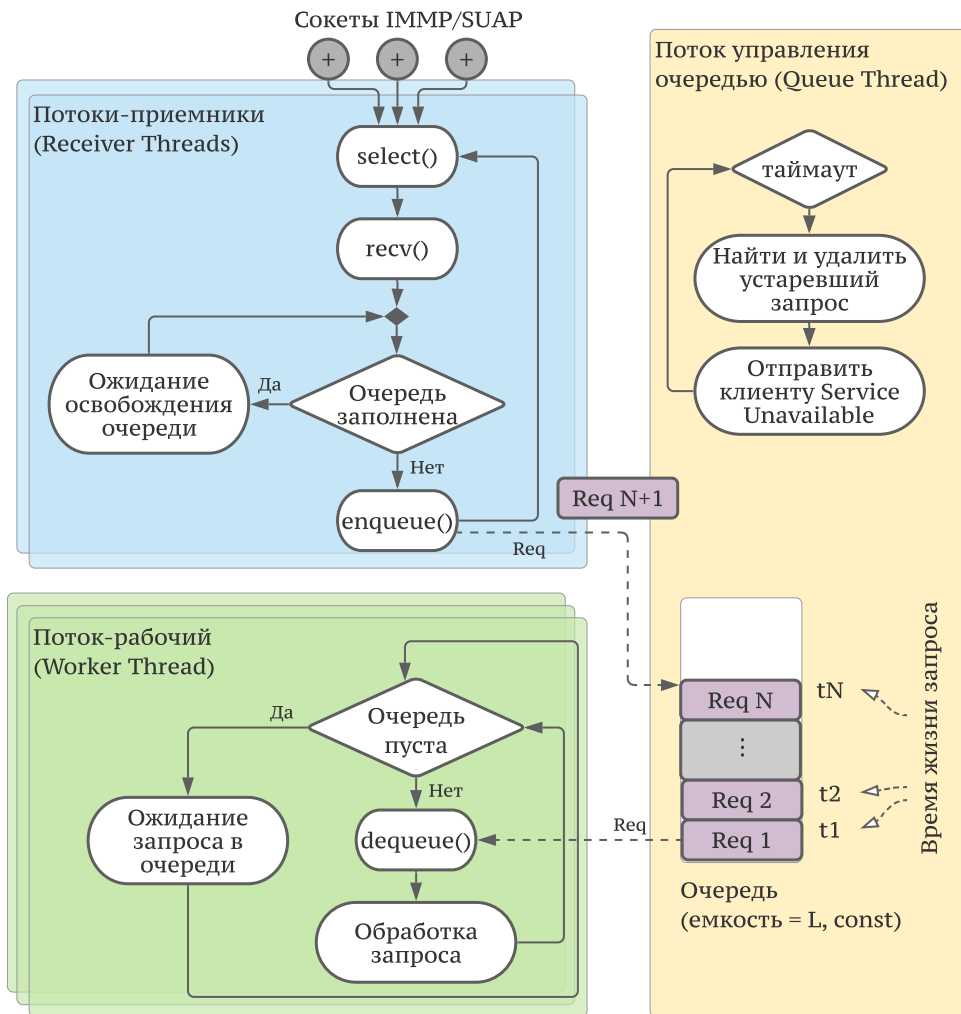


Рисунок 5.11 — Архитектура супервизора SVRD

Попав в очередь, запрос не ждет бесконечно долго, пока рабочий не освободится. Вместо этого в системе определено максимальное время ожидания обслуживания, по истечении которого запрос удаляется из очереди. Отслеживанием истечения времени ожидания занимается отдельный поток, управляющий очередью (см. рис. 5.11). Очередь имеет фиксированную емкость.

Управление перегрузками в системе реализовано на стороне отправителя. Если очередь заполнена, поток-приемник ждет освобождения места без ограничения времени. Так как у всех запросов есть предельные времена ожидания, рано или поздно место в очереди освободится, поток-приемник не зависнет навсегда. Отправитель IMMP- или SUAP-запроса ждет ответа в течение ограниченного времени. Если ответа в течение этого времени не приходит, он сигнализирует о перегрузке системы. Таким образом, результат обработки, отправленный слишком поздно, будет просто проигнорирован.

Отметим, что особенность помещения запросов в очередь, когда поток-приемник ждет освобождения очереди, может быть оптимизирована в следующих версиях системы: желательно синхронизировать предельное время нахождения запроса на стороне приемника (и в очереди, и до этого, в ожидании освобождения очереди) с предельным временем ожидания ответа отправителем запроса. Это позволит быстрее снижать нагрузку при перегрузке, уда-

для устаревшие запросы еще до их попадания в очередь. В качестве альтернативы, можно передавать отправителю ответ с кодом 411 (Not Ready) сразу при обнаружении заполненной очереди потоком-приемником.

Супервизор хранит конфигурацию других модулей, которые он передает в сообщении IMMР АСК при подключении к нему (см. рис. 5.5). Если во время работы настройки модуля меняются, они хранятся в конфигурации времени исполнения и сбрасываются при перезапуске модуля. Для переноса параметра из конфигурации времени исполнения в хранимую конфигурацию на супервизоре нужно выполнить процедуру «system.store-config», передав соответствующий CALL-запрос.

### 5.3.2 RFID-адаптер G2RD

Модуль RFID-адаптера (G2RD) осуществляет взаимодействие с радиомодулем RFID. Он поддерживает два протокола: IMMР (в роли клиента) и ITOP (в роли сервера). Как и другие модули системы, после включения он подключается по IMMР к супервизору и получает от него стартовую конфигурацию, а в дальнейшем получает от супервизора запросы на получение или изменение параметров и вызов процедур. По протоколу ITOP к G2RD подключаются другие модули (TFPD, веб-интерфейс и прочие), которым нужно выполнить те или иные операции над метками.

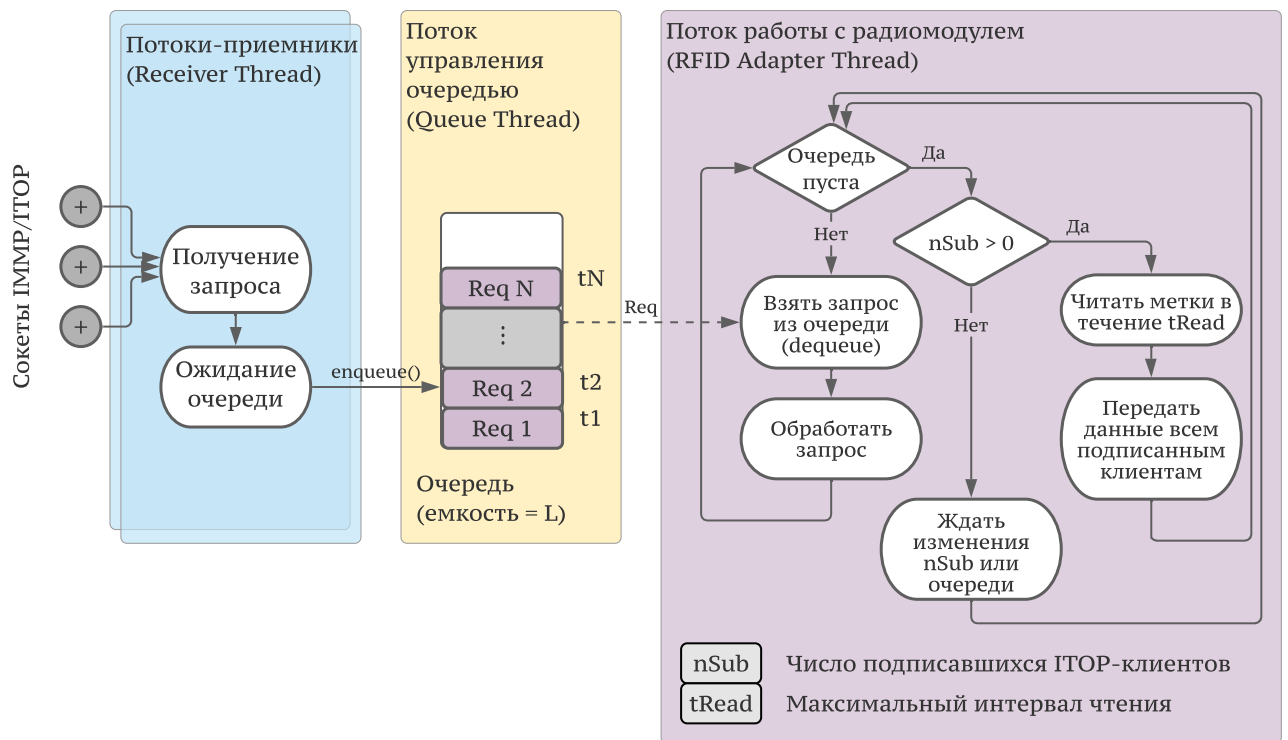


Рисунок 5.12 — Архитектура RFID-адаптера G2RD

Основная сложность в реализации G2RD заключается в том, что сам радиомодуль RFID может одновременно выполнять только одну операцию (читать или записать метку,

получить или установить параметр). То есть радиомодуль является разделяемым ресурсом, доступ к которому должен быть синхронизирован. Если другой модуль подписался на поток меток, и радиомодуль будет постоянно занят чтением, то запросы на его настройку или запись меток не смогут быть выполнены до тех пор, пока подписка не будет остановлена, что не всегда возможно. Так как запросы к G2RD по IMMP приходят от потоков-рабочих супервайзера, которые работают синхронно, блокировка запросов в G2RD привела бы каскадно к повышению загрузки SVR — его потоки-рабочие были бы заблокированы до истечения предельного времени обработки запроса. Для того, чтобы избежать такой ситуации и дать возможность работать со считывателем одновременно различным модулям, потребовалось обеспечить приоритизацию операций.

На рис. 5.12 показана архитектура RFID-адаптера. Потоки-приемники и поток управления очередью схожи с аналогичными потоками, используемыми в SVR (см. рис. 5.11). Потоки-приемники получают, десериализуют и помещают в очередь запросы, пришедшие по протоколам IMMP и ITOP, а поток управления очередью отслеживает устаревшие запросы и удаляет их. Поток-рабочий, который обрабатывает запросы, в G2RD один, так как именно он обеспечивает взаимодействие с радиомодулем. Этот поток отслеживает число подписок, интервалы чтения, указанные в запросах ITOP SUBSCRIBE, и ведет учет подписавшихся ITOP-сессий. Если в очереди есть хотя бы один запрос, поток-рабочий выполняет его. Если запросов нет, но есть активные подписки, поток-рабочий инициирует чтение в течение кванта времени. Этот квант выбирается максимальным среди всех квантов, указанных в запросах SUBSCRIBE, и не превосходит 500 мс. После окончания чтения, поток опять проверяет очередь запросов и продолжает чтение только в том случае, если очередь по-прежнему пуста. Таким образом исключается блокировка на непрерывном чтении, а обработка запросов приобретает более высокий приоритет. Если очередь пуста и подписок нет, поток спит до тех пор, пока не появятся запросы или подписки. В это время радиомодуль RFID неактивен и не расходует энергию.

### 5.3.3 Приложение TFPD и клиент TFPC

Модуль-приложение TFPD используется в качестве прокси-сервера между клиентами, подключающимися по протоколу TFP, и RFID-адаптером, с которым TFPD взаимодействует по протоколу ITOP (см. рис. 5.10). Приложение TFPD позволяет снять часть нагрузки с RFID-адаптера, так как при подключении нескольких клиентов оно будет поддерживать единственную сессию с RFID-адаптером и отправлять данные во все клиентские TFP-сессии. Как и другие модули системы, начальную конфигурацию TFPD получает от SVR сразу после включения и подключения к SVR по протоколу IMMP. Конфигурация TFPD включает, среди прочего, максимальное число одновременных клиентских соединений.

В состав системы входит утилита TFPC, выполняющая функции клиента TFP. Она позволяет подключаться к TFPD и записывать полученные метки в базу или файл. Основное



назначение TFPC — подключение к системе управления из внешней информационной системы и запись прочитанных меток в базу данных, однако она также может быть использована для кэширования меток на считывателе. Такое решение использовалось, в частности, в ходе эксперимента в городе Казань, когда сеть периодически оказывалась перегружена данными с видеокамер, и соединение с центром обработки данных обрывалось.

Изначально TFPC, как и остальные модули системы, был реализован на языке C++. При подготовке к эксперименту в 2020 году была также выполнена его реализация на языке Python 3.

### 5.3.4 Утилита UAX и интерфейсы управления

В системе были реализованы два интерфейса управления: веб-интерфейс и интерфейс командной строки. Обоим интерфейсом нужно отправлять супервайзеру по протоколу SUAP запросы на получение или установку параметров модулей и запрашивать удаленное выполнение процедур, а также взаимодействовать с RFID-адаптером по протоколу ITOP. Обработка команд в любом из интерфейсов управления сводится к четырем шагам: проверке корректности аргументов команды, формированию и отправке запроса по SUAP или ITOP, ожиданию ответа от SVR или G2RD соответственно и выводу результата пользователю. Для унификации и упрощения разработки интерфейсов управления была реализована утилита UAX (User Agent Executor), которая в качестве аргумента получает строку с командой, обменивается данными с нужным модулем системы и выводит результат в поток стандартного вывода. Таким образом, любому интерфейсу достаточно сформировать строку с командой, запустить утилиту UAX в отдельном процессе и получить результат в его выводе, а по коду завершения определить успешность выполнения команды.

Команды UAX имеют вид «*операция*:*[модуль.объект|действие]* *аргументы*», где «*операция*» — одна из строк «**get**», «**set**», «**call**», «**rfid**», «**login**» или «**logout**». Операции «**get**», «**set**» и «**call**» должны включать адрес в виде названия модуля и имени объекта, например «**tfpd.conns-num**» (число подключений к TFPD) или «**g2rd.power-read**» (мощность, на которой радиомодуль читает метки). Операция «**rfid**» должна включать одно из доступных действий: «**subscribe**», «**readmem**», «**writemem**» или «**writeepc**». Во всех командах после описания действия и адреса объекта нужно передать список аргументов в формате «*ключ=значение*»: в команде «**login**» нужно передать имя пользователя и пароль, а во всех остальных — как минимум ключ «**skey**», который был получен с помощью команды «**login**». Результаты выполнения команд выводятся в стандартный поток. Всем операциям кроме «**rfid**» соответствуют запросы SUAP (см. раздел 5.2.2), а операции «**rfid**» — запросы ITOP (см. раздел 5.2.3), соответствующие выбранному действию. Отметим, что команда «**rfid:subscribe**» позволяет читать метки только в течение ограниченного времени, от 200 до 1000 миллисекунд, за счет установки заголовка «**Continuous: false**» в сообщении ITOP SUBSCRIBE (см. рис. 5.8).

В следующем примере администратор подключается к SVRD, получает ключ SKey и запрашивает число текущих подключений к TFPD, температуру радиомодуля и мощность считывателя. Затем он изменяет текущую мощность считывателя и сохраняет новое значение, чтобы оно использовалось после перезагрузки. После этого пользователь завершает сессию.

```

$ uax "login: username=admin password=qwerty"
42
# Получить число активных подключений к TFPD:
$ uax "get:tfpd.conns-num skey=42"
5 tfpd.conns-num: 1
# Получить температуру радиомодуля:
$ uax "get:g2rd.temperature skey=42"
g2rd.temperature: 27
# Получить текущую мощность радиомодуля (100*дБм):
10 $ uax "get:g2rd.power-read skey=42"
g2rd.power-read: 2900 # 29 дБм
# Установить значение мощности чтения радиомодуля:
$ uax "set:g2rd.power-read skey=42 value=3150"
$ uax "get:g2rd.power-read skey=42"
15 g2rd.power-read: 3150 # 31.5 дБм
# Сохранить новое значение мощности радиомодуля
$ uax "call:system.store-config skey=42 plist=g2rd.power-read"
# Выйти из системы (сделать невалидным ключ):
$ uax "logout: skey=42"

```

## 5.4 Реализация RFID-считывателей и системы управления

### 5.4.1 Аппаратная реализация считывателей

Экспериментальные RFID-считыватели были выпущены во всепогодном исполнении с питанием через Ethernet (PoE). К каждому считывателю можно подключить до четырех антенн.

Аппаратная структура считывателя показана на рис. 5.13. Считыватель состоит из одноплатного компьютера, радиомодуля RFID и модуля питания. Из-за всепогодного исполнения использовалось пассивное охлаждение, тепло от радиомодуля и процессора отводилось через радиатор на металлический корпус. Одноплатный компьютер построен на четырехъядерном процессоре Exynos4412 Prime 1,7 ГГц ARM Cortex-A9, имеет 2 ГБ оперативной памяти LP-DDR2, интерфейс FastEthernet 100 Мб/с и сменную карту постоянной памяти

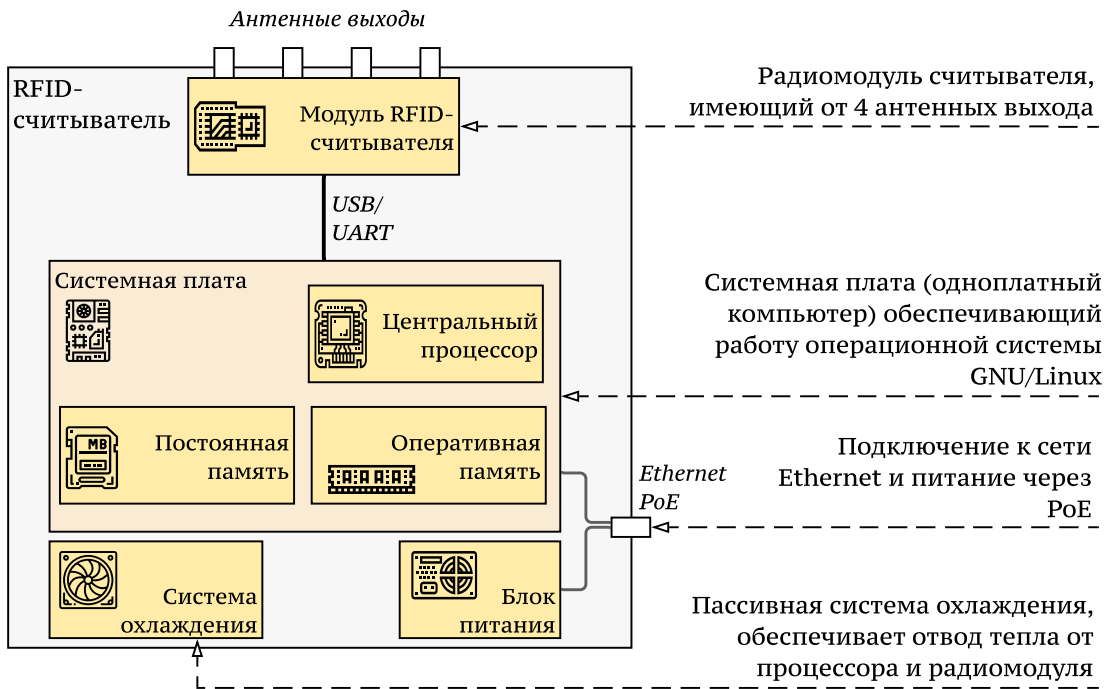


Рисунок 5.13 — Аппаратная структура RFID-считывателя

на 8 ГБ. Питание осуществлялось через интерфейс Ethernet. Радиомодуль RFID имеет четыре антенных порта, он подключается к управляющему компьютеру через интерфейс USB. Модуль обеспечивает выходную мощность до 31,5 дБм.

#### 5.4.2 Особенности программной реализации

Все описанные ранее в разделах 5.2 и 5.3 программные модули и протоколы, кроме веб-интерфейса и части RFID-адаптера, взаимодействующей со считывателем, были реализованы на языке C++. Часть RFID-адаптера, работающая непосредственно с радиомодулем считывателя, была реализована на C. Для реализации веб-интерфейса использовался фреймворк Flask на языке Python 3.

Все компоненты системы работают под управлением операционной системы Arch Linux. Для взаимодействия с сервисами операционной системы (например, настройки сетевого адаптера, журнала syslog или перезагрузки) были написаны скрипты на языках Bash и Python, которые вызываются SVR при обращении к соответствующим процедурам или свойствам. Например, для получения текущего IP-адреса считывателя нужно отправить запрос IMMP GET «system.network-address» на SVR. Получив его, поток-работчий в SVR выполнит в отдельном процессе скрипт, который просмотрит текущие сетевые интерфейсы и вернет IP-адрес, связанный с интерфейсом Ethernet. Для установки статического IP-адреса нужно передать команду IMMP CALL «system.network-setup-static» с параметрами «address», «netmask» и «gateway», получив которую SVR выполнит другой скрипт, настраивающий IP-адрес, маску и шлюз. За счет такого подхода систему достаточно просто адаптировать под

разные дистрибутивы Linux, в которых могут использоваться различные версии системных служб. Так, система была успешно портирована на Ubuntu и Armbian.

Выделим несколько особенностей, общих для всех модулей кроме веб-интерфейса, реализованного на Python. Во-первых, в реализациях очередей, клиентов и серверов практически не используется динамическая память (функции `malloc/free`, операторы `new/delete`, функции `make_shared<T>()` и пр.). Вместо этого под каждое сообщение или очередь заранее статически выделяется область памяти, заведомо превышающая предельно допустимый размер хранимого объекта. Этот подход позволяет не только повысить производительность за счёт отсутствия работы с выделением и освобождением памяти, но и существенно повысить надёжность за счёт снижения вероятности появления ошибок утечки памяти или обращений к ранее освобождённой памяти.

Во-вторых, большинство модулей, включающих серверы каких-либо протоколов (IMMP, ITOP и пр.), имеют в своем составе очереди, с помощью которых обработка запросов от различных клиентов мультиплексируется и может осуществляться даже одним потоком. Опрос сокетов также производится с мультиплексированием с помощью функции `pselect`.

В-третьих, все модули реализованы в виде многопоточных приложений с использованием библиотеки `pthread` (POSIX Threads).

При разработке модуля G2RD требовалось использовать программный интерфейс радиомодуля, написанный на языке C. Для того, чтобы иметь возможность быстрой замены модулей, была разработана промежуточная библиотека (также на языке C), реализующая основные функции: чтение метки в течение заданного интервала, запись значения EPCID на метку, запись значения в заданный банк памяти, получение и запись значений параметров радиомодуля. Эту библиотеку несложно адаптировать к новым радиомодулям, при наличии соответствующей спецификации протокола взаимодействия или API.

## 5.5 Экспериментальные испытания

В общей сложности было проведено три экспериментальных испытания разработанной системы радиочастотной идентификации автомобилей. Первое масштабное испытание системы состоялось в 2014–2015 годах, когда в городе Казань более 700 автобусов были оснащены метками, на улицах установлены считыватели и в течение нескольких зимних месяцев производилась регистрация проездов. Следующий эксперимент состоялся весной 2020 года на полигоне в Казани, в нем проверялась успешность идентификации автомобилей, едущих со скоростями до 170 км/ч, совершающими обгон и прочие маневры. Наконец, третий эксперимент начался летом 2021 года на платном участке центральной кольцевой автодороги вокруг Москвы (ЦКАД), в нем метками были оснащены 9 автомобилей аварийных комиссаров. Цель последнего эксперимента — проверить применимость UHF RFID в системе бесконтактной оплаты проезда по платной дороге. На момент написания диссертации этот эксперимент еще не завершен.

### 5.5.1 Испытания в Казани в 2014 году

В конце осени 2014 года был запущен большой эксперимент в городе Казань (республика Татарстан). В ходе эксперимента номерами с радиометками были оснащены 740 автобусов и были оборудованы четыре точки радиочастотной идентификации. Оборудование для двух из них было разработано коллективом при активном участии автора (проектирование архитектуры системы управления и протоколов; разработка программного обеспечения системы управления; установка, настройка программ в ЦОДе и настройка считывателей; анализ полученных результатов и расчёт вероятности идентификации).

Каждый считыватель оснащался четырьмя антеннами, по две антенны на каждую из полос. Одна антенна в каждой паре была направлена на чтение передних номеров, вторая — задних. Угол наклона антенн был приблизительно равен  $45^\circ$ . Считыватели размещались на тех же опорах, на которых были установлены камеры видеофиксации: на Кремлевской дамбе на штанге над дорогой, на высоте 5,8 м, а на пересечении ул. Гаврилова и просп. Ямашева — под сводом путепровода, немного ниже.

Номера с метками для этого эксперимента были разработаны немецкой компанией Topnjes Group и выпущены российской фирмой ООО «Знак». Чипы меток были производства компании NXP. На метках были записаны уникальные значения EPCID, а для повышения достоверности использовалась идентификация и по EPCID, и по TID. Номера TID на метках имели класс E0 [1], то есть первые 8 бит содержали шестнадцатиричное значение E0, следующие 8 бит — код производителя, а последние 48 бит — уникальный номер метки.

Схема размещения считывателей и программных компонентов показана на рис. 5.14. На считывателях были установлены супервайзеры (SVR), приложения TFPD, RFID-адаптеры и веб-серверы. Сервер был размещен в центре обработки данных (ЦОД) ГИБДД города Казань, подключение к считывателям происходило посредством протокола TFP. На сервере в ЦОД был установлен клиент TFPC и база данных MySQL, в которую записывались прочитанные метки. Позднее на считыватели пришлось добавить модули TFPC для кэширования — из-за перегрузок в сети (она использовалась для передачи видеотрафика от камер) TFP-соединения часто разрывались и данные терялись. Для каждой метки в базу данных записывались время, идентификатор считывателя, значения EPC и TID, уровень сигнала от метки, число прочтений, номер антенны. Также в отдельной таблице хранились соответствия значений TID и номерных знаков, поэтому была возможность выгрузки информации о времени проезда автобусов напрямую из базы данных. Схема базы данных приведена на рис. 5.14.

Эксперимент длился до начала 2015 года, в течение поздней осени и зимы. Задачей эксперимента было практически обосновать применимость технологии радиочастотной идентификации для регистрации проездов автомобилей на нормальных скоростях в условиях городского потока при плохих погодных условиях.

Параметры настройки радиомодулей RFID-считывателей приведены в табл. 8. Отметим две особенности. Во-первых, у меток всегда читались ровно 64 бита (4 машинных слова) банка TID, а длина значения EPCID не ограничивалась. Во-вторых, опрос всегда проводился

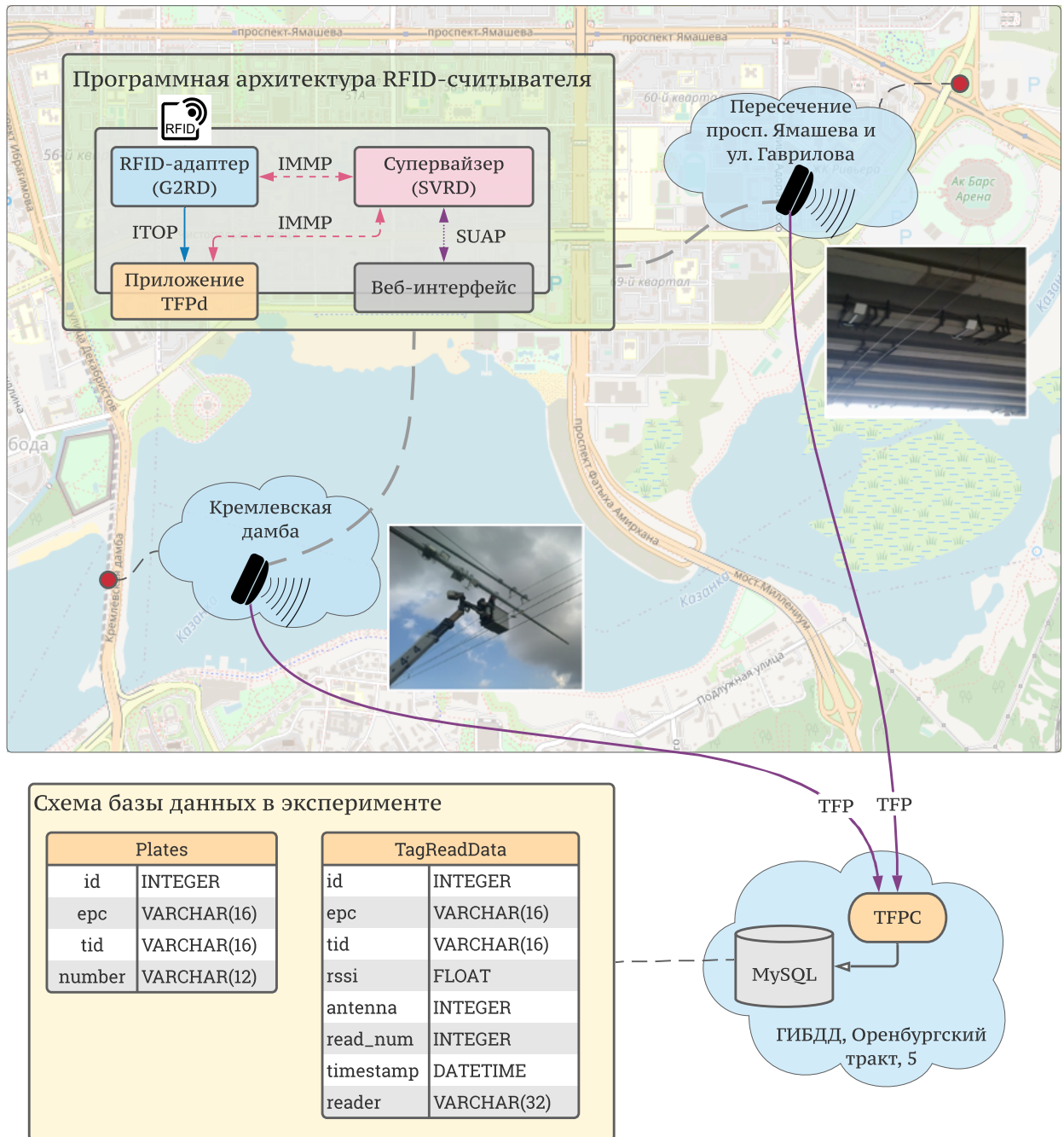


Рисунок 5.14 — Схема размещения считывателей и программных компонентов распределенной системы в Казани в 2014 году

по флагу А. Как показали более поздние исследования, результаты которых были приведены в главе 3, это не всегда оптимальная стратегия, и можно немного улучшить идентификацию, если периодически инвертировать флаг. В то же время, считыватели периодически переключали антенны, поэтому сброс питания так или иначе производился. Настройки параметров Q, TagI и M оказались достаточно удачными, что позднее было показано и теоретически (см. главу 2).

Для оценки вероятности идентификации на протяжении нескольких дней силами ГИБДД Казани были собраны данные о проездах автобусов, с помощью визуального наблюдения. Эти данные обрабатывались и сравнивались с информацией, полученной от

Таблица 8 — Параметры RFID-считывателей в Казани в 2014 году.

Параметр	Значение
Интервал Tagi	12,5 мкс
Показатель числа слотов (Q)	2
Число символов на бит (M)	4
Использовать расширенную преамбулу (TRext)	0 (нет)
Длина EPCiD	любая
Длина TID	64 бита
Сессия	S0
Флаг опроса меток	Всегда А
Выходная мощность радиомодуля	31,5 дБм
Усиление антенн	8 дБ
Общие потери в кабелях и разъемах	2,6 дБ

считывателей, с поправкой на погрешности (например, время могло немного отличаться). Результаты несколько различались между точками и составили от 92 % идентифицированных транспортных средств на Кремлевской дамбе до 96 % на пересечении ул. Гаврилова и просп. Ямашева. Еще одно наблюдение из эксперимента — в 2014 году практически все считанные метки были метками в номерах, то есть прочтения «случайных» меток были очень редкими и при выборе параметра Q можно было руководствоваться тем, что коллизии практически не происходят даже при малом числе слотов. Как показали более поздние эксперименты, которые будут описаны далее, к 2021 году это предположение перестало быть верным.

Следует отметить, что одной из основных проблем при проведении эксперимента оказалась высокая перегруженность оптических каналов связи между считывателями и центром обработки данных, которые также использовались для передачи уличного видеонаблюдения. Перегрузки приводили к тому, что значительную часть времени соединений между считывателями и сервером не было, но за счет использования кэширования удалось избежать потери данных о проездах. В то же время, для реального использования системы радиочастотной идентификации и обеспечения оперативного воздействия при обнаружении нарушений правил дорожного движения или, например, выявления похищенных автомобилей может требоваться либо предоставление гарантированной пропускной способности в существующих проводных каналах, либо построение беспроводных многошаговых сетей.

### 5.5.2 Испытания в Казани в 2020 году

Следующие испытания состоялись в Казани в 2020 году на полигоне ИТС ООО «Казань-Телематика» в рамках НИР по применению RFID для идентификации транспорта. Эти испытания длились несколько дней, их целью было проверить работоспособность системы при проезде автомобилей на высоких скоростях и при совершении маневров. Метки для этого эксперимента были произведены АО «Микрон».

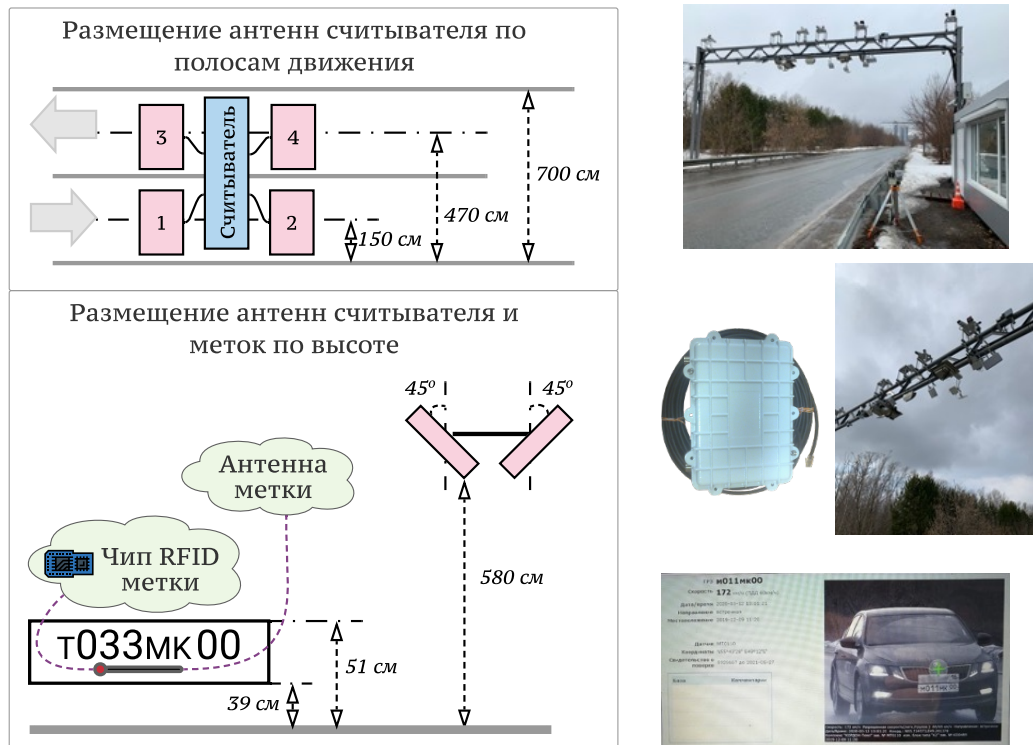


Рисунок 5.15 — Схема размещения RFID-оборудования в Казани в 2020 году

Считыватель размещался на той же опорной конструкции, на которой размещены системы видеofиксации нарушений. Схема расположения антенн считывателя, а также размещение метки на номерном знаке и фотографии с полигона приведены на рис. 5.15. Настройки считывателя были теми же, что и в эксперименте 2014 года (см. табл. 8).

Во время эксперимента проверялась успешность идентификации метки при проезде с заданной скоростью от 40 км/ч до 170 км/ч (самый быстрый проезд был на скорости 172 км/ч), при обгоне одним автомобилем другого и при следовании автомобилей друг за другом в потоке. Во всех случаях метки были успешно идентифицированы. Для точного измерения скорости движения использовался комплекс «Кордон-М» фирмы ООО «Симикон».

В отличие от предыдущего эксперимента 2014 года, метки АО «Микрон» в этом эксперименте имели класс E2 [1], то есть первый байт содержал шестнадцатиричное значение E2, следующие три бита — служебные флаги, потом 9-битный идентификатор производителя MDID (Mask Designer Identifier), 12-битный номер модели и уже после него уникальный номер метки. Общая длина идентификатора, таким образом, составляла 96 бит (6 машинных слов). К сожалению, эта особенность стала известна уже во время эксперимента, поэтому



различить метки по TID не удалось — первые 64 бита у всех меток совпадали. В дальнейшем эта проблема была исправлена и уже в следующем эксперименте считыватели были настроены на 96-битные TID.

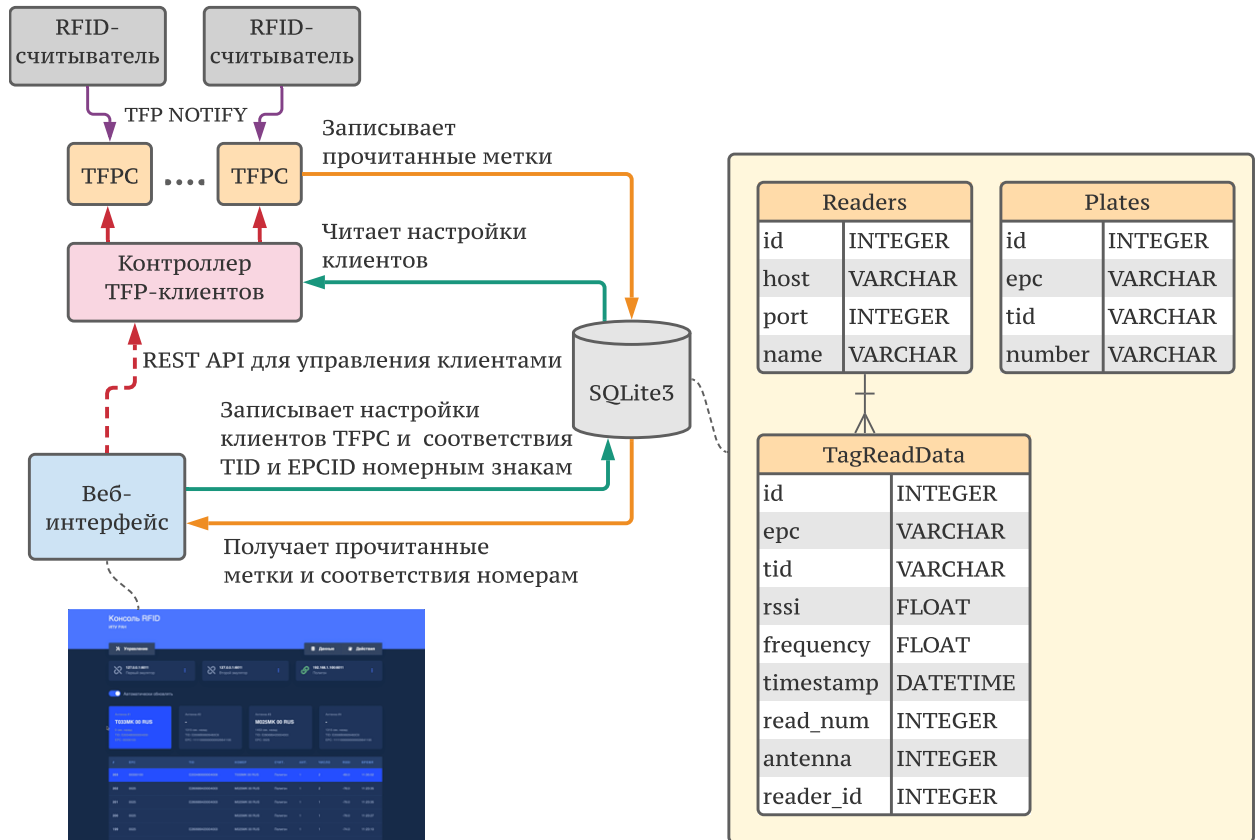


Рисунок 5.16 — Схема клиентского программного обеспечения для испытаний

Также для эксперимента потребовалось реализовать более функциональную клиентскую часть, позволяющую настраивать клиентов TFPC и получать статистику проездов через веб-интерфейс. Схема программного обеспечения и упрощенная модель данных показаны на рис. 5.16. Вся клиентская часть была построена на языке Python 3. Она включала клиентов TFPC, которые создавались динамически контроллером TFP-клиентов. Для взаимодействия между TFP-клиентами и контроллером использовался простой бинарный протокол. Параметры подключения контроллер получал из базы данных SQLite, в которую же записывали данные TFP-клиенты. Настройка считывателей и получение статистики проездов производилась через веб-интерфейс, реализованный на Flask. Для управления началом и остановкой чтения контроллер предоставлял программный интерфейс REST API, к которому подключался тот же веб-интерфейс.

### 5.5.3 Испытания на платном участке ЦКАД в 2021 году

Последние испытания начались в 2021 году на платном участке центральной кольцевой автодороги (ЦКАД) и проводились, как и в 2020 году, совместно с АО «Микрон». Целью экс-

перимента являлась проверка возможности использования UHF RFID для оплаты проезда по платной дороге и проверка точности позиционирования идентифицируемых автомобилей. В рамках эксперимента оборудована одна точка идентификации считывателем с четырьмя антеннами над двумя полосами одного направления. Метки производства АО «Микрон» были установлены на 9 автомобилей службы аварийных комиссаров, периодически проезжающих по этому участку ЦКАД. В отличие от предыдущих экспериментов антенны считывателя были установлены на большей высоте.

В эксперименте использовалось то же оборудование и программное обеспечение на считывателе, что и в эксперименте в Казани 2020 года. С учетом особенностей меток, считыватель был настроен на чтение 96 бит TID. Изменения коснулись только клиентской части — вместо реализации TFPC на Python 3 была использована более эффективная и надежная реализация на C++, модель данных была расширена для учета связей антенн с полосами и направлениями движения, и вместо СУБД SQLite использовалась более производительная и надежная PostgreSQL.

Стоит отметить, что если в 2014 году сторонние, не участвующие в эксперименте, метки составляли не более 10 % от всех прочитанных меток, то в 2021 году таких меток стало на несколько порядков больше. Так, за 26 дней эксперимента было прочитано более 63000 меток, и лишь 115 прочтений относились к меткам в номерах машин, используемых в эксперименте. Вероятно, остальные метки размещаются на упаковках товаров в кузовах фур, корпусах автомобилей или под лобовыми стеклами (например, для бесконтактного проезда на парковку или в дачные поселки).

## 5.6 Заключение

В главе были представлены следующие результаты.

1. Разработана распределенная система управления RFID-считывателями. Для управления и передачи данных в системе используются различные протоколы, позволяющие избежать перегрузки системы при увеличении числа RFID-считывателей. Система управления обладает высокой гибкостью, позволяет реализовывать на ее основе считыватели различной конфигурации.
2. Проведены три экспериментальных испытания системы: в 2014 и в 2020 годах в Казани, и в 2021 году в Московской области на ЦКАД. Успешные результаты испытаний показывают, что разработанную систему радиочастотной идентификации можно использовать для регистрации автомобилей, движущихся с большими скоростями и совершающих различные маневры на дороге.
3. Полученные результаты экспериментов согласуются с теоретическими результатами, представленными в диссертационной работе. Отмечено, что для эффективного использования системы желательно иметь выделенные каналы связи, для построения которых можно использовать многошаговые беспроводные сети.

## Заключение

Основные результаты работы заключаются в следующем.

1. Предложена и исследована стохастическая модель системы радиочастотной идентификации ТС, учитывающая скорость движения RFID-меток, расположенных на номерных знаках автомобилей, а также различные сценарии проведения циклического опроса и сбора данных с меток.
2. Разработан комплекс аналитических и имитационных моделей для анализа вероятности идентификации ТС, учитывающих особенности логического и физического уровней протокола стандарта EPC Class 1 Gen.2, и особенности распространения радиосигналов между RFID-меткой и считывателем.
3. Предложена методика моделирования многошаговых беспроводных сетей с помощью тандемных сетей массового обслуживания, учитывающая особенности трафика и интерференции в каналах связи.
4. Разработан метод вычисления оценок характеристик многофазных систем массового обслуживания большой размерности, моделирующих компьютерные сети, с коррелированными входными потоками и распределениями обслуживания фазового типа.
5. Разработана архитектура и реализована распределенная компьютерная система управления RFID-считывателями, предназначенная для организации сбора данных об идентифицированных транспортных средствах.
6. Произведены экспериментальные внедрения разработанной компьютерной системы радиочастотной идентификации транспортных средств на автодорогах в г. Казань и в Московской области.

## Список сокращений и условных обозначений

- ALE** Application Level Events, события уровня приложений
- API** Application Programming Interface, программный интерфейс приложения
- ASK** Amplitude Shift Keying, амплитудная манипуляция
- AWGN** Additive white Gaussian noise, аддитивный белый гауссовский шум
- BER** Bit Error Rate, вероятность битовой ошибки
- BLF** Backscatter Link Frequency, частота обратного рассеяния
- CRC** Cyclic Redundancy Check, циклический избыточный код
- CSMA/CA** Collision Sense Multiple Access with Collision Avoidance, множественный доступ с контролем несущей и избеганием коллизий
- CTS** Clear to Send, подтверждение доступности канала для передачи
- CW** в контексте RFID: Constant Wave, постоянный синусоидальный сигнал; в контексте CSMA/CA: Contention Window, размер конкурентного окна
- DCF** Distributed Coordination Function, распределенная функция координации
- DCI** Discovery, Configuration, and Initialization, поиск, настройка и инициализация
- DIFS** DCF Interframe Space, межкадровый интервал в DCF
- DR** Division Ratio, коэффициент деления
- DSRC** Dedicated short-range communications, выделенная связь ближнего действия
- EDCA** Enhanced Distributed Channel Access, усовершенствованный доступ к каналу
- EDCF** Enhanced DCF, усовершенствованный DCF
- EM** Expectation Maximization, максимизация ожидания
- EPC** Electronic Product Code, электронный код продукции
- EPCID** Electronic Product Code Identifier, идентификатор, хранящийся в банке памяти EPC в метке
- HCCA** HCF Controlled Channel Access, доступ к каналу с использованием HCF
- HCF** Hybrid Coordinator Function, гибридная функция координации
- HTTP** HyperText Transfer Protocol, протокол передачи гипертекста
- IEC** International Electrotechnical Commission, Международная электротехническая комиссия
- IEEE** Institute of Electrical and Electronics Engineers, Институт инженеров электротехники и электроники
- ISO** International Organization for Standardization, международная организация по стандартизации
- IoE** Internet of Everything, интернет всего
- IoT** Internet of Things, интернет вещей
- JSON** JavaScript Object Notation, обозначение объектов JavaScript

- LLRP** Low Level Reader Protocol, протокол считывателя низкого уровня
- MAP** Markovian Arrival Process, марковский входящий поток
- MDID** Mask-Designer Identifier, идентификатор производителя метки
- MIB** Management Information Base, база информации управления
- MSP** Markov Service Process, марковский процесс обслуживания
- PC** Protocol Control, поле управления протоколом
- PH** Phase-Type Distribution, распределение фазового типа
- PIE** Pulse Interval Encoding, кодирование длительностью импульса
- PSK** Phase-Shift Keying, фазовая манипуляция
- QoS** Quality of Service, качество обслуживания
- REST** Representational state transfer, передача репрезентативного состояния
- RFID** Radio Frequency Identification, радиочастотная идентификация
- RPC** Remote Procedure Call, удаленный вызов процедур
- RTcal** Reader-Tag calibration, калибровочный символ для канала от считывателя к метке
- RTS** Request to Send, запрос на захват канала
- SaaS** Software as a Service, программное обеспечение как сервис
- SIFS** Short Interframe Space, короткий межкадровый интервал
- SIP** Session Initiation Protocol, протокол установления сеанса
- SNMP** Simple Network Management Protocol, простой протокол сетевого управления
- SNR** Signal-to-Noise Ration, соотношение сигнал-шум
- TCP** Transmission Control Protocol, протокол управления передачей
- TID** Tag Identification, банк идентификации метки
- TLS** Transport Layer Security, протокол защиты транспортного уровня
- TRcal** Tag-Reader calibration, калибровочный символ для канала от метки к считывателю
- UDP** User Datagram Protocol, протокол пользовательских датаграмм
- UHF** Ultra High Frequency, ультравысокие частоты
- XaaS** Everything as a Service, что угодно как сервис
- XML** eXtensible Markup Language, расширяемый язык разметки
- ССК** Сильно связанная компонента
- ГИБДД** Государственная инспекция безопасности дорожного движения
- ЦКАД** Центральная кольцевая автодорога
- ЦОД** Центр обработки данных

## Словарь терминов

**backoff** : алгоритм двоичной экспоненциальной выдержки, процедура выбора случайного времени ожидания начала передачи при возникновении коллизий

**RFID** : технология радиочастотной идентификации, в которой активное устройство (считыватель) получает идентификатор пассивного устройства (метки) с помощью передачи радиосигналов

$\mathbb{P}\{A\}$  : вероятность события  $A$

$\mathbb{M}\xi$  : математическое ожидание случайной величины  $\xi$

$\oplus$  : операция Кронекеровой суммы матриц

$\{A\}_{ij}$  :  $(i,j)$ -й элемент матрицы  $A$

$I_W$  : единичная матрица размерности  $W$

$\mathbf{0}$  : нулевой вектор-строка

$\mathbf{1}$  : вектор-столбец, состоящий из единиц

## Список литературы

1. EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard. Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz. Release 2.1. — EPCGlobal, 2018. — P. 157.
2. ГОСТ Р 58701-2019 (ИСО/МЭК 18000-63:2015) Информационные технологии. Идентификация радиочастотная для управления предметами. Часть 63. Параметры радиointерфейса для диапазона частот 860 – 960 МГц (Тип С). — М.: Стандартинформ, 2019. — 557 с.
3. Abramson Norman. THE ALOHA SYSTEM // Proceedings of the November 17-19, 1970, fall joint computer conference on - AFIPS '70 (Fall). — New York, New York, USA: ACM Press, 1970. — P. 281.
4. Roberts Lawrence G. ALOHA packet system with and without slots and capture // ACM SIGCOMM Computer Communication Review. — 1975. — apr. — Vol. 5, no. 2. — Pp. 28–42.
5. Finkenzeller Klaus. RFID Handbook. — Chichester, UK: John Wiley & Sons, Ltd, 2010. — jun.
6. Dobkin Daniel M. The RF in RFID. — Elsevier, 2008. — P. 504.
7. Wu Hao-Ping. Intelligent Parking Management System Utilizing RFID // The ACM MobiSys 2019 on Rising Stars Forum - RisingStarsForum'19. — New York, New York, USA: ACM Press, 2019. — Pp. 37–41.
8. Cho Jae Hyung, Cho Myeong-Woo. Effective Position Tracking Using B-Spline Surface Equation Based on Wireless Sensor Networks and Passive UHF-RFID // IEEE Transactions on Instrumentation and Measurement. — 2013. — sep. — Vol. 62, no. 9. — Pp. 2456–2464.
9. Park Sangdo, Lee Hongchul. Self-Recognition of Vehicle Position Using UHF Passive RFID Tags // IEEE Transactions on Industrial Electronics. — 2013. — jan. — Vol. 60, no. 1. — Pp. 226–234.
10. Errington Angus F. C., Daku Brian L. F., Prugger Arnfinn F. Initial Position Estimation Using RFID Tags: A Least-Squares Approach // IEEE Transactions on Instrumentation and Measurement. — 2010. — nov. — Vol. 59, no. 11. — Pp. 2863–2869.
11. Zhai Yue, Guo Qiang, Min Hao. An Effective Velocity Detection Method for Moving UHF-RFID Tags // 2018 IEEE International Conference on RFID Technology & Application (RFID-TA). — IEEE, 2018. — sep. — Pp. 1–5.
12. Ubiquitous and Low Power Vehicles Speed Monitoring for Intelligent Transport Systems / Jose Luis Calderon Choy, Jing Wu, Chengnian Long, Yi-Bing Lin // IEEE Sensors Journal. — 2020. — jun. — Vol. 20, no. 11. — Pp. 5656–5665.

13. Speeding detection in RFID Systems on Roads / Tao Jing, Xingni Li, Wei Cheng, Yan Huo, Xiaoshuang Xing // 2013 International Conference on Connected Vehicles and Expo (IC-CVE). — IEEE, 2013. — dec. — Pp. 953–954.
14. Yang Chao, Wang Xuyu, Mao Shiwen. Unsupervised Drowsy Driving Detection With RFID // IEEE Transactions on Vehicular Technology. — 2020. — aug. — Vol. 69, no. 8. — Pp. 8151–8163.
15. Localizing Missing Entities Using Parked Vehicles: An RFID-Based System / Wynita M. Griggs, Rudi Verago, Joe Naoum-Sawaya, Rodrigo H. Ordonez-Hurtado, Robert Gilmore, Robert N. Shorten // IEEE Internet of Things Journal. — 2018. — oct. — Vol. 5, no. 5. — Pp. 4018–4030.
16. Electronic Vehicle Identification (EVI). White Paper. V.1. — RAIN RFID Alliance, 2018. — P. 10.
17. Landt J. The history of RFID // IEEE Potentials. — 2005. — oct. — Vol. 24, no. 4. — Pp. 8–11.
18. González Salvador Ricardo Meneses, Linares y Miranda Roberto. Passive UHF RFID Technology Applied to Automatic Vehicle Identification // Advanced RFID Systems, Security, and Applications. — IGI Global, 2013. — Pp. 188–220.
19. Blythe P. RFID for road tolling, road-use pricing and vehicle access control // IEEE Colloquium. RFID Technology. — Vol. 1999. — IEEE, 1999. — Pp. 8–16.
20. Khan Atif Ali, Yakzan Adnan I. Elberjaoui, Ali Maaruf. Radio Frequency Identification (RFID) Based Toll Collection System // 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks. — IEEE, 2011. — jul. — Pp. 103–107.
21. Yoon Won-Ju, Chung Sang-Hwa, Lee Seong-Joon. Implementation and performance evaluation of an active RFID system for fast tag collection // Computer Communications. — 2008. — nov. — Vol. 31, no. 17. — Pp. 4107–4116.
22. Al-Naima Fawzi M., Al-Any Hassan. Vehicle Location System Based on RFID // 2011 Developments in E-systems Engineering. — IEEE, 2011. — dec. — Pp. 473–478.
23. Tseng Jan-Dong, Wang Wen-De, Ko Rong-Jie. An UHF Band RFID Vehicle Management System // 2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID). — IEEE, 2007. — apr. — Pp. 390–393.
24. RFID-Based Navigation of Subway Trains / Alexander M. Kostrominov, Oleg N. Tyulyandin, Alexander B. Nikitin, Michael N. Vasilenko, Alexander T. Osminin // 2020 IEEE East-West Design & Test Symposium (EWDTS). — IEEE, 2020. — sep. — Pp. 1–6.



25. Unterhuber Alexander R., Iliev Stoyan, Biebl Erwin. Optimal Tilt Angle of Reader Antennas for Truck Tolling Based on UHF RFID // 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech). — IEEE, 2019. — jun. — Pp. 1–6.
26. Unterhuber Alexander R., Iliev Stoyan, Biebl Erwin M. Estimation Method for High-Speed Vehicle Identification With UHF RFID Systems // IEEE Journal of Radio Frequency Identification. — 2020. — dec. — Vol. 4, no. 4. — Pp. 343–352.
27. Unterhuber Alexander R., Iliev Stoyan, Biebl Erwin. Influence of the Vehicle Velocity on the Number of Reads in UHF RFID Scenarios // 2019 IEEE International Conference on RFID Technology and Applications (RFID-TA). — IEEE, 2019. — sep. — Pp. 421–426.
28. Dynamic RFID Identification in Urban Traffic Management Systems / Bartosz Pawłowicz, Bartosz Trybus, Mateusz Salach, Piotr Jankowski-Mihulowicz // Sensors. — 2020. — jul. — Vol. 20, no. 15. — P. 4225.
29. Mobile RFID Tag Detection Influence Factors and Prediction of Tag Detectability / Minho Jo, Hee Yong Youn, Si-Ho Cha, Hyunseung Choo // IEEE Sensors Journal. — 2009. — feb. — Vol. 9, no. 2. — Pp. 112–119.
30. COSMO: CO-Simulation with MATLAB and OMNeT++ for Indoor Wireless Networks / Zhi Zhang, Zhonghai Lu, Qiang Chen, Xiaolang Yan, Li-Rong Zheng // 2010 IEEE Global Telecommunications Conference GLOBECOM 2010. — IEEE, 2010. — dec. — Pp. 1–6.
31. Meneses Gonzalez Ricardo, Orosco Vega Roberto, Linares y Miranda Roberto. Some considerations about RFID system performance applied to the vehicular identification // 2011 IEEE International Conference on RFID-Technologies and Applications. — IEEE, 2011. — sep. — Pp. 123–127.
32. Lonkar Bhupesh B., Sayankar Manoj R., Charde Pallavi D. Design and Monitor Smart Automatic Challan Generation Based on RFID Using GPS and GSM // Proceedings of 3rd International Conference on Advances in Internet of Things & Connected Technologies (ICIoTCT) 2018. — SSRN, 2018.
33. Vehicle identification system through the interoperability of an ultra high frequency radio frequency identification system and its database / Jessie R. Balbin, Ramon G. Garcia, Flordeliza L. Valiente, Brian Christopher F. Aaron, Christopher John D. Celimen, Juan Carlos K. De Peralta, Joshua P. Despabiladeras // 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). — IEEE, 2017. — dec. — Pp. 1–5.
34. Bhavke Akshay, Pai Sadhana. Smart weight based toll collection & vehicle detection during collision using RFID // 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS). — IEEE, 2017. — aug. — Pp. 1–6.

35. Zhang Xiaoqiang, Tentzeris Manos. Applications of Fast-Moving RFID Tags in High-Speed Railway Systems // *International Journal of Engineering Business Management*. — 2011. — may. — Vol. 3. — P. 6.
36. Performance analysis of «fast-moving» RFID tags in state-of-the-art high-speed railway systems / Xiaoqiang Zhang, Vasileios Lakafosis, Anya Traille, Manos M. Tentzeris // *2010 IEEE International Conference on RFID-Technology and Applications*. — IEEE, 2010. — jun. — Pp. 281–285.
37. Pandit Anala Aniruddha, Mundra Ankit Kumar, Talreja Jyot. RFID Tracking System for Vehicles (RTSV) // *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*. — IEEE, 2009. — jul. — Pp. 160–165.
38. Sundar Rajeshwari, Hebbar Santhoshs, Golla Varaprasad. Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection // *IEEE Sensors Journal*. — 2015. — feb. — Vol. 15, no. 2. — Pp. 1109–1113.
39. Marais H., Grobler M. J., Holm J. E. W. Modelling of an RFID-based electronic vehicle identification system // *2013 Africon*. — IEEE, 2013. — sep. — Pp. 1–5.
40. Passive RFID-Based Inventory of Traffic Signs on Roads and Urban Environments / José García Oya, Rubén Martín Clemente, Eduardo Hidalgo Fort, Ramón González Carvajal, Fernando Muñoz Chavero // *Sensors*. — 2018. — jul. — Vol. 18, no. 7. — P. 2385.
41. An Efficient Scheme for Tag Information Update in RFID Systems on Roads / Tao Jing, Xing Wei, Wei Cheng, Mingyang Guan, Liran Ma, Yan Huo, Xiuzhen Cheng // *IEEE Transactions on Vehicular Technology*. — 2016. — apr. — Vol. 65, no. 4. — Pp. 2435–2444.
42. Wireless inventory of traffic signs based on passive RFID technology / E. Hidalgo, F. Munoz, A. Guerrero de Mier, R.G. Carvajal, Ruben Martin-Clemente // *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*. — IEEE, 2013. — nov. — Pp. 5467–5471.
43. On the Design and Deployment of RFID Assisted Navigation Systems for VANETs / Wei Cheng, Xiuzhen Cheng, Min Song, Biao Chen, Wendy W. Zhao // *IEEE Transactions on Parallel and Distributed Systems*. — 2012. — jul. — Vol. 23, no. 7. — Pp. 1267–1274.
44. An RFID-Based Intelligent Vehicle Speed Controller Using Active Traffic Signals / Joshué Pérez, Fernando Seco, Vicente Milanés, Antonio Jiménez, Julio C. Díaz, Teresa De Pedro // *Sensors*. — 2010. — jun. — Vol. 10, no. 6. — Pp. 5872–5887.
45. Porter J.D., Kim D.S. An RFID-Enabled Road Pricing System for Transportation // *IEEE Systems Journal*. — 2008. — jun. — Vol. 2, no. 2. — Pp. 248–257.
46. Understanding Citywide Resident Mobility Using Big Data of Electronic Registration Identification of Vehicles / Linjiang Zheng, Dong Xia, Li Chen, Dihua Sun // *IEEE Transactions on Intelligent Transportation Systems*. — 2020. — oct. — Vol. 21, no. 10. — Pp. 4363–4377.

47. HERO: Online Real-Time Vehicle Tracking / Hongzi Zhu, Minglu Li, Yanmin Zhu, L.M. Ni // IEEE Transactions on Parallel and Distributed Systems. — 2009. — may. — Vol. 20, no. 5. — Pp. 740–752.
48. Floerkemeier C., Sarma S. RFIDSim—A Physical and Logical Layer Simulation Engine for Passive RFID // IEEE Transactions on Automation Science and Engineering. — 2009. — jan. — Vol. 6, no. 1. — Pp. 33–43.
49. Wideband system-level simulator for passive UHF RFID / D. Arnitz, U. Muehlmann, T. Gigl, K. Witrissal // 2009 IEEE International Conference on RFID. — IEEE, 2009. — apr. — Pp. 28–33.
50. A Site-Specific Stochastic Propagation Model for Passive UHF RFID / Antonis G. Dimitriou, Stavroula Siachalou, Aggelos Bletsas, John N. Sahalos // IEEE Antennas and Wireless Propagation Letters. — 2014. — Vol. 13. — Pp. 623–626.
51. Azpilicueta Leyre, Vargas-Rosales Cesar, Falcone Francisco. Intelligent Vehicle Communication: Deterministic Propagation Prediction in Transportation Systems // IEEE Vehicular Technology Magazine. — 2016. — sep. — Vol. 11, no. 3. — Pp. 29–37.
52. Griffin J.D., Durgin G.D. Complete Link Budgets for Backscatter-Radio and RFID Systems // IEEE Antennas and Propagation Magazine. — 2009. — apr. — Vol. 51, no. 2. — Pp. 11–25.
53. Nikitin Pavel V., Rao K.V.S., Lam Sander. UHF RFID TAG CHARACTERIZATION: OVERVIEW AND STATE-OF-THE-ART // Proceedings of AMTA 2012. — 2012.
54. Nikitin P.V., Rao K. Effect of Gen2 protocol parameters on RFID tag performance // 2009 IEEE International Conference on RFID. — IEEE, 2009. — apr. — Pp. 117–122.
55. Nikitin Pavel V., Rao K. V. S. Antennas and Propagation in UHF RFID Systems // 2008 IEEE International Conference on RFID. — IEEE, 2008. — apr. — Pp. 277–288.
56. Nikitin Pavel, Rao K.V.S., Lazar Steve. An Overview of Near Field UHF RFID // 2007 IEEE International Conference on RFID. — IEEE, 2007. — Pp. 167–174.
57. Nikitin P.V., Rao K.V.S. Theory and measurement of backscattering from RFID tags // IEEE Antennas and Propagation Magazine. — 2006. — dec. — Vol. 48, no. 6. — Pp. 212–218.
58. Nikitin P.V., Rao K.V.S. Performance limitations of passive UHF RFID systems // 2006 IEEE Antennas and Propagation Society International Symposium. — IEEE, 2006. — Pp. 1011–1014.
59. Rao K.V.S., Nikitin P.V., Lam S.F. Antenna design for UHF RFID tags: a review and a practical application // IEEE Transactions on Antennas and Propagation. — 2005. — dec. — Vol. 53, no. 12. — Pp. 3870–3876.

60. Zanetti Davide, Danev Boris, Capkun Srdjan. Physical-layer identification of UHF RFID tags // Proceedings of the sixteenth annual international conference on Mobile computing and networking - MobiCom '10. — New York, New York, USA: ACM Press, 2010. — P. 353.
61. Lazaro Antonio, Girbau David, Villarino Ramon. EFFECTS OF INTERFERENCES IN UHF RFID SYSTEMS // Progress In Electromagnetics Research. — 2009. — Vol. 98. — Pp. 425–443.
62. Buettner Michael, Wetherall David. An empirical study of UHF RFID performance // Proceedings of the 14th ACM international conference on Mobile computing and networking - MobiCom '08. — New York, New York, USA: ACM Press, 2008. — P. 223.
63. Vogt Harald. Efficient Object Identification with Passive RFID Tags // Pervasive Computing. Pervasive 2002. Lecture Notes in Computer Science. — 2002. — Vol. 2414. — Pp. 98–113.
64. Performance analysis of RFID Generation-2 protocol / Chonggang Wang, Mahmoud Daneshmand, Kazem Sohraby, Bo Li // IEEE Transactions on Wireless Communications. — 2009. — may. — Vol. 8, no. 5. — Pp. 2592–2601.
65. Vahedi Ehsan, Ward Rabab K., Blake Ian F. Analytical modeling of RFID Generation-2 protocol using absorbing Markov chain theorem // 2012 IEEE Global Communications Conference (GLOBECOM). — IEEE, 2012. — dec. — Pp. 385–390.
66. Markovian model for computation of tag loss ratio in dynamic RFID systems / J. Vales-Alonso, M. V. Bueno-Delgado, E. Egea-Lopez, J. J. Alcaraz-Espin, J. Garcia-Haro // RFID Systech 2009 - Excellence in RFID Systems and Technologies, 5th European Workshop on RFID Systems and Technologies. — 2009.
67. On the Optimal Identification of Tag Sets in Time-Constrained RFID Configurations / Javier Vales-Alonso, María Victoria Bueno-Delgado, Esteban Egea-López, Juan José Alcaraz, Juan Manuel Pérez-Mañogil // Sensors. — 2011. — mar. — Vol. 11, no. 3. — Pp. 2946–2960.
68. Modeling the Anti-Collision Process of RFID System by Markov Chain / Qiaoling Tong, Xuecheng Zou, Dongsheng Liu, Yifei Dai // 2007 International Conference on Wireless Communications, Networking and Mobile Computing. — IEEE, 2007. — sep. — Pp. 2054–2057.
69. Analysis of the Power Outage Effects in RFID / Javier Vales-Alonso, Juan Pedro Munoz-Gea, Juan J. Alcaraz, F. J. Gonzalez-Castano // IEEE Communications Letters. — 2017. — feb. — Vol. 21, no. 2. — Pp. 306–309.
70. Di Marco Piergiuseppe, Santucci Fortunato, Fiscione Carlo. Modeling anti-collision protocols for RFID Systems with multiple access interference // 2014 IEEE International Conference on Communications, ICC 2014. — IEEE Computer Society, 2014. — Pp. 5938–5944.
71. A closed form solution for frame slotted ALOHA utilizing time and multiple collision recovery coefficients / Hazem A. Ahmed, Hamed Salah, Joerg Robert, Albert Heuberger // 2016 IEEE

- Topical Conference on Wireless Sensors and Sensor Networks (WiSNet). — IEEE, 2016. — jan. — Pp. 11–14.
72. Numeric Evaluation on the System Efficiency of the EPC Gen-2 UHF RFID Tag Collision Resolution Protocol in Error Prone Air Interface / Xin-Qing Yan, Yang Liu, Bin Li, Xue-Mei Liu // *International Journal of Distributed Sensor Networks*. — 2014. — mar. — Vol. 10, no. 3. — P. 716232.
73. Jeon Ki Yong, Cho Sung Ho. Performance of RFID EPC C1 Gen2 Anti-collision in Multi-path Fading Environments // *2009 Second International Conference on Communication Theory, Reliability, and Quality of Service*. — IEEE, 2009. — jul. — Pp. 125–128.
74. Kim Jeong Geun, Shin Woo Jin, Yoo Ji Ho. Performance Analysis of EPC Class-1 Generation-2 RFID Anti-collision Protocol // *Computational Science and Its Applications – ICCSA 2007*. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. — Pp. 1017–1026.
75. Vahedi Ehsan, Ward Rabab K., Blake Ian F. Performance Analysis of RFID Protocols: CDMA Versus the Standard EPC Gen-2 // *IEEE Transactions on Automation Science and Engineering*. — 2014. — oct. — Vol. 11, no. 4. — Pp. 1250–1261.
76. La Porta T F, Maselli G, Petrioli C. Anticollision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization // *IEEE Transactions on Mobile Computing*. — 2011. — feb. — Vol. 10, no. 2. — Pp. 267–279.
77. Bianchi Giuseppe. Performance analysis of the IEEE 802.11 distributed coordination function // *IEEE Journal on Selected Areas in Communications*. — 2000. — mar. — Vol. 18, no. 3. — Pp. 535–547.
78. Lauwens Ben, Scheers Bart, Van de Capelle Antoine. Performance analysis of unslotted CSMA/CA in wireless networks // *Telecommunication Systems*. — 2010. — jun. — Vol. 44, no. 1-2. — Pp. 109–123.
79. Chatzimisios P., Vitsas V., Boucouvalas A. C. Throughput and delay analysis of IEEE 802.11 protocol // *Proceedings 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications*. — 2002. — Oct. — Pp. 168–174.
80. Banchs Albert, Serrano Pablo, Azcorra Arturo. End-to-end delay analysis and admission control in 802.11 DCF WLANs // *Computer Communications*. — 2006. — apr. — Vol. 29, no. 7. — Pp. 842–854.
81. Sakurai Taka, Vu Hai L. MAC access delay of IEEE 802.11 DCF // *IEEE Transactions on Wireless Communications*. — 2007. — may. — Vol. 6, no. 5. — Pp. 1702–1710.
82. On the end-to-end delay analysis of the IEEE 802.11 distributed coordination function / J. S. Vardakas, I. Papapanagiotou, M. D. Logothetis, S. A. Kotsopoulos // *Second International Conference on Internet Monitoring and Protection, ICIMP 2007*. — 2007.

83. Packet delay analysis on IEEE 802.11 DCF under finite load traffic in multi-hop ad hoc networks / LinFang Dong, YanTai Shu, HaiMing Chen, MaoDe Ma // Science in China Series F: Information Sciences. — 2008. — apr. — Vol. 51, no. 4. — Pp. 408–416.
84. Hung Fu-Yi, Marsic Ivan. Access Delay Analysis of IEEE 802.11 DCF in the Presence of Hidden Stations // IEEE GLOBECOM 2007-2007 IEEE Global Telecommunications Conference. — IEEE, 2007. — nov. — Pp. 2541–2545.
85. Tickoo Omesh, Sikdar O. Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks // IEEE/ACM Transactions on Networking. — 2008. — Vol. 16, no. 4. — Pp. 878–891.
86. Felemban Emad, Ekici Eylem. Single hop IEEE 802.11 DCF analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases // IEEE Transactions on Wireless Communications. — 2011. — oct. — Vol. 10, no. 10. — Pp. 3256–3266.
87. Haghani Ehsan, Krishnan Michael N., Zakhor Avidah. A method for estimating access delay distribution in IEEE 802.11 networks // GLOBECOM - IEEE Global Telecommunications Conference. — 2011.
88. Dai Lin, Sun Xinghua. A unified analysis of IEEE 802.11 DCF networks: Stability, throughput, and delay // IEEE Transactions on Mobile Computing. — 2013. — Vol. 12, no. 8. — Pp. 1558–1572.
89. Exact distribution of access delay in IEEE 802.11 DCF MAC / T. Issariyakul, D. Niyato, E. Hossain, A. S. Alfa // GLOBECOM '05. IEEE Global Telecommunications Conference, 2005. — Vol. 5. — 2005. — Nov. — Pp. 5 pp.–2538.
90. Duffy Ken, Malone David, Leith Douglas J. Modeling the 802.11 Distributed Coordination Function in Non-Saturated Conditions // Ieee Communications Letters. — 2005. — Vol. 9, no. 8. — Pp. 715–717.
91. Admission control in IEEE 802.11e wireless LANs / Deyun Gao, Jianfei Cai, King Ngi Ngan, Hong Kong // IEEE Network. — 2005. — Vol. 19, no. 4. — Pp. 6–13.
92. Charfi Emna, Chaari Lamia, Kamoun Lotfi. Upcoming WLANs MAC access mechanisms: An overview // Proceedings of the 2012 8th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2012. — 2012.
93. Engelstad Paal E, Osterbo Olav N. Queueing delay analysis of IEEE 802.11e EDCA // Ifip Wons. — 2006.
94. Hazra PK, De A. Performance Analysis of IEEE 802.11 e EDCA with QoS Enhancements through TXOP based Frame-concatenation and Block-acknowledgement // International Journal of Advancements in Technology. — 2011. — Vol. 2, no. 4. — Pp. 542–560.

95. Performance analysis of IEEE 802.11e contention-based channel access / Zhen-ning Kong  
Zhen-ning Kong, D.H.K. Tsang, B. Bensaou, Deyun Gao // IEEE Journal on  
Selected Areas in Communications. — 2004. — Vol. 22, no. 10. — Pp. 2095–2106.
96. Liu Jing, Niu Zhisheng. Delay analysis of IEEE 802.11 e EDCA under unsaturated condi-  
tions // Networking Conference, 2007. WCNC 2007. — 2007.
97. Inan Inanc, Keceli Feyza, Ayanoglu Ender. Analysis of the 802.11e enhanced distributed  
channel access function // IEEE Transactions on Communications. — 2009. — Vol. 57,  
no. 6. — Pp. 1753–1764.
98. Misic Jelena, Rashwand Saeed, Misic Vojislav B. Analysis of impact of TXOP allocation  
on IEEE 802.11e EDCA under variable network load // IEEE Transactions on Parallel and  
Distributed Systems. — 2012. — Vol. 23, no. 5. — Pp. 785–799.
99. Yanfeng Zhu, Zhisheng Niu. A novel queueing model for finite load IEEE 802.11 WLANs //  
VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005. — Vol. 2. — IEEE,  
2006. — Pp. 1352–1356.
100. An Analytical Model for Capacity Evaluation of VoIP on HCCA and TCP File Transfers  
over EDCA in an IEEE 802.11e WLAN / Sri Harsha, S. V. R. Anand, Anurag Kumar,  
Vinod Sharma // International Conference on Distributed Computing and Networking /  
Ed. by Soma Chaudhuri, Samir R. Das, Himadri S. Paul, Srikanta Tirthapura. — Berlin,  
Heidelberg: Springer Berlin Heidelberg, 2006. — Vol. 4308 of *Lecture Notes in Computer  
Science*. — Pp. 245–256.
101. Ghazizadeh Reza. A Priority Queuing Model for HCF Controlled Channel Access (HCCA)  
in Wireless LANs // Int'l J. of Communications, Network and System Sciences. — 2009. —  
Vol. 02, no. 01. — Pp. 30–43.
102. Rashd Mohammad M., Hossain Ekram, Bhargava Vijay K. Queueing analysis of 802.11e  
HCCA with variable bit rate traffic // IEEE International Conference on Communications.  
— 2006. — Vol. 10, no. c. — Pp. 4792–4798.
103. Analysis of very high throughput (VHT) at MAC and PHY layers under MIMO channel in  
IEEE 802.11ac WLAN / Gul Zameen Khan, Ruben Gonzalez, Eun-chan Park, Xin-wen Wu //  
2017 19th International Conference on Advanced Communication Technology (ICACT). —  
IEEE, 2017. — Pp. 877–888.
104. IEEE 802.11ac: Enhancements for very high throughput WLANs / Eng Hwee Ong,  
Jarkko Knecht, Olli Alanen, Zheng Chang, Toni Huovinen, Timo Nihtila // IEEE Inter-  
national Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC. —  
2011. — Pp. 849–853.

105. Chandra Kishor, Prasad R. Venkatesha, Niemegeers Ignas. Performance Analysis of IEEE 802.11ad MAC Protocol // IEEE Communications Letters. — 2017. — 07. — Vol. 21, no. 7. — Pp. 1513–1516.
106. Heyman D.P., Lucantoni D. Modeling multiple ip traffic streams with rate limits // IEEE/ACM Transactions on Networking. — 2003. — dec. — Vol. 11, no. 6. — Pp. 948–958.
107. Klemm Alexander, Lindemann Christoph, Lohmann Marco. Modeling IP traffic using the batch Markovian arrival process // Performance Evaluation. — 2003. — oct. — Vol. 54, no. 2. — Pp. 149–173.
108. Vishnevskii V. M., Dudin A. N. Queueing systems with correlated arrival flows and their applications to modeling telecommunication networks // Automation and Remote Control. — 2017. — aug. — Vol. 78, no. 8. — Pp. 1361–1403.
109. Вишнеvский В.М., Дудин А.Н., Клименок В.И. Стохастические системы с коррелированными потоками. Теория и применение в телекоммуникационных сетях. — Москва: Техносфера, 2018. — С. 564.
110. Дудин А.Н., Клименок В.И. Системы массового обслуживания с коррелированными потоками. — Минск: БГУ, 2000. — С. 175.
111. Johnson Mary A., Taaffe Michael R. Matching moments to phase distributions: Mixtures of erlang distributions of common order // Communications in Statistics. Stochastic Models. — 1989. — Vol. 5, no. 4. — Pp. 711–743.
112. Buchholz Peter, Kriege Jan, Felko Iryna. Input Modeling with Phase-Type Distributions and Markov Models. SpringerBriefs in Mathematics. — Cham: Springer International Publishing, 2014.
113. Neuts Marcel F. A versatile Markovian point process // Journal of Applied Probability. — 1979. — dec. — Vol. 16, no. 4. — Pp. 764–779.
114. Scott Steven L., Smyth Padhraic. The Markov Modulated Poisson Process and Markov Poisson Cascade with Applications to Web Traffic Data // Bayesian Statistic. — 2003. — Vol. 7. — Pp. 671–680.
115. Lucantoni David M. The BMAP/G/1 queue: A tutorial // Performance Evaluation of Computer and Communication Systems. Performance 1993, SIGMETRICS 1993. Lecture Notes in Computer Science. — 1993. — Vol. 729. — Pp. 330–358.
116. HE Qi-Ming. The Versatility of MMAP[K] and the MMAP[K]/G[K]/1 Queue // Queueing Systems 2001 38:4. — 2001. — Vol. 38, no. 4. — Pp. 397–418.
117. Van Houdt Benny. Analysis of the adaptive MMAP[K]/PH[K]/1 queue: A multi-type queue with adaptive arrivals and general impatience // European Journal of Operational Research. — 2012. — aug. — Vol. 220, no. 3. — Pp. 695–704.



118. Buchholz Peter, Kemper Peter, Kriege Jan. Multi-class Markovian arrival processes and their parameter fitting // Performance Evaluation. — 2010. — nov. — Vol. 67, no. 11. — Pp. 1092–1106.
119. Klimenok Valentina, Dudin Alexander, Vishnevsky Vladimir. Priority Multi-Server Queueing System with Heterogeneous Customers // Mathematics. — 2020. — sep. — Vol. 8, no. 9. — P. 1501.
120. Analysis of queueing networks with blocking using a new aggregation technique / Johann Christoph Strelen, Berthold Bärk, Jürgen Becker, Volker Jonas // Annals of Operations Research. — 1998. — Vol. 79. — Pp. 121–142.
121. Henderson W., Taylor P. G. Product form in networks of queues with batch arrivals and batch services // Queueing Systems. — 1990. — 12. — Vol. 6, no. 1. — Pp. 71–87.
122. Strelen Johann Christoph. Approximate product form solutions for Markov chains // Performance Evaluation. — 1997. — 07. — Vol. 30, no. 1-2. — Pp. 87–110.
123. Strelen Johann Christoph. Approximate Disaggregation-Aggregation Solutions for General Queueing Networks // Soc. for Computer Simulation. — 1997. — Pp. 773–778.
124. Analysis of a semi-open queueing network with Markovian arrival process / Jiseung Kim, Alexander Dudin, Sergey Dudin, Chesoong Kim // Performance Evaluation. — 2018. — apr. — Vol. 120. — Pp. 1–19.
125. A Tandem BMAP/G/1  $\rightarrow$  M/N/0 Queue with Group Occupation of Servers at the Second Station / Chesoong Kim, Alexander Dudin, Valentina Klimenok, Olga Taramin // Mathematical Problems in Engineering. — 2012. — Vol. 2012. — Pp. 1–26.
126. The BMAP/G/1  $\rightarrow$  /PH/1/M tandem queue with feedback and losses / Che Soong Kim, Valentina Klimenok, Gennadiy Tsarenkov, Lothar Breuer, Alexander Dudin // Performance Evaluation. — 2007. — 08. — Vol. 64, no. 7-8. — Pp. 802–818.
127. Buchholz Peter. Bounding stationary results of Tandem networks with MAP input and PH service time distributions // ACM SIGMETRICS Performance Evaluation Review. — 2006. — jun. — Vol. 34, no. 1. — Pp. 191–202.
128. Bean Nigel, Green David, Taylor Peter. The Output Process of an MMPP/M/1 Queue // Journal of Applied Probability. — 1998. — Vol. 35, no. 4. — Pp. 998–1002.
129. Lian Zhaotong, Zhao Ning. Departure processes and busy periods of a tandem network // Operational Research. — 2011. — 11. — Vol. 11, no. 3. — Pp. 245–257.
130. Horváth Gábor, Okamura Hiroyuki. A Fast EM Algorithm for Fitting Marked Markovian Arrival Processes with a New Special Structure // Computer Performance Engineering. EPEW 2013. Lecture Notes in Computer Science. — 2013. — Vol. 8168. — Pp. 119–133.

131. Ephraim Yariv, Roberts William J.J. An EM algorithm for Markov modulated Markov processes // *IEEE Transactions on Signal Processing*. — 2009. — Vol. 57, no. 2. — Pp. 463–470.
132. Buchholz Peter. An EM-algorithm for MAP fitting from real traffic data // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. — 2003. — Vol. 2794. — Pp. 218–236.
133. Okamura Hiroyuki, Dohi Tadashi. Faster Maximum Likelihood Estimation Algorithms for Markovian Arrival Processes // *2009 Sixth International Conference on the Quantitative Evaluation of Systems*. — Budapest, Hungary: IEEE, 2009. — sep. — Pp. 73–82.
134. Asmussen Søren, Nerman Olle, Olsson Marita. Fitting Phase-type Distributions via the EM Algorithm // *Scandinavian Journal of Statistics*. — 1996. — Vol. 23, no. 4. — Pp. 419–441.
135. Bobbio Andrea, Cumani Aldo. ML estimation of the parameters of a PH distribution in triangular canonical form // *Computer performance evaluation*. — 1992. — no. 22. — Pp. 33–46.
136. Thummler A., Buchholz P., Telek M. A Novel Approach for Fitting Probability Distributions to Real Trace Data with the EM Algorithm // *2005 International Conference on Dependable Systems and Networks (DSN'05)*. — Yokohama, Japan, Japan: IEEE, 2005. — Pp. 712–721.
137. Okamura Hiroyuki, Dohi Tadashi, Trivedi Kishor S. Improvement of expectation-maximization algorithm for phase-type distributions with grouped and truncated data // *Applied Stochastic Models in Business and Industry*. — 2013. — mar. — Vol. 29, no. 2. — Pp. 141–156.
138. Okamura Hiroyuki, Dohi Tadashi, Trivedi Kishor S. A refined em algorithm for PH distributions // *Performance Evaluation*. — 2011. — oct. — Vol. 68, no. 10. — Pp. 938–954.
139. El Abdouni Khayari Rachid, Sadre Ramin, Baverkort Boudewijn R. Fitting world-wide web request traces with the EM-algorithm // *Performance Evaluation*. — 2003. — apr. — Vol. 52, no. 2-3. — Pp. 175–191.
140. Osogami Takayuki, Harchol-Balter Mor. Closed form solutions for mapping general distributions to quasi-minimal PH distributions // *Performance Evaluation*. — 2006. — jun. — Vol. 63, no. 6. — Pp. 524–552.
141. Bobbio A., Horváth A., Telek M. Matching Three Moments with Minimal Acyclic Phase Type Distributions // *Stochastic Models*. — 2005. — jan. — Vol. 21, no. 2-3. — Pp. 303–326.
142. Telek Miklós, Heindl Armin. Matching Moments For Acyclic Discrete And Continuous Phase-Type Distributions Of Second Order // *International Journal of Simulation Systems, Science and Technology*. — 2003. — Vol. 3.
143. Vanden Bosch Peter M., Dietz Dennis C., Pohl Edward A. Moment matching using a family of phase-type distributions // *Communications in Statistics. Stochastic Models*. — 2000. — jan. — Vol. 16, no. 3-4. — Pp. 391–398.

144. Horváth Gábor, Telek Miklós. A Canonical Representation of Order 3 Phase Type Distributions // Formal Methods and Stochastic Models for Performance Evaluation. EPEW 2007. Lecture Notes in Computer Science. — 2007. — Vol. 4748. — Pp. 48–62.
145. Schmickler Leonhard. Meda: Mixed erlang distributions as phase-type representations of empirical distribution functions // Communications in Statistics. Stochastic Models. — 1992. — Vol. 8, no. 1. — Pp. 131–156.
146. Telek M., Horváth G. A minimal representation of Markov arrival processes and a moments matching method // Performance Evaluation. — 2007. — oct. — Vol. 64, no. 9-12. — Pp. 1153–1168.
147. Canonical form based MAP(2) fitting / Levente Bodrog, Peter Buchholz, Jan Krieger, Miklós Telek // Proceedings - 7th International Conference on the Quantitative Evaluation of Systems, QEST 2010. — 2010. — Pp. 107–116.
148. Current results and open questions on PH and MAP characterization / L Bodrog, A Heindl, A. Horváth, Gábor Horváth, M Telek // Numerical Methods for Structured Markov Chains. Dagstuhl Seminar Proceedings. — Schloss Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), 2008. — Pp. 1–6.
149. Horvath G., Buchholz P., Telek M. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag correlation // Second International Conference on the Quantitative Evaluation of Systems (QEST'05). — IEEE, 2005. — Pp. 124–133.
150. Casale Giuliano, Zhang Eddy Z., Smirni Evgenia. Trace data characterization and fitting for Markov modeling // Performance Evaluation. — 2010. — Vol. 67. — Pp. 61–79.
151. Aldous D., Shepp L. The least variable phase type distribution is Erlang // Communications in Statistics. Stochastic Models. — 1987. — Vol. 3. — Pp. 467–473.
152. Horvath Andras, Horvath Gabor, Telek Miklos. A Joint Moments Based Analysis of Networks of MAP/MAP/1 Queues // Performance Evaluation. — 2010. — Vol. 67. — Pp. 759–778.
153. Al-Jaroodi Jameela, Aziz Junaid, Mohamed Nader. Middleware for RFID Systems: An Overview // 2009 33rd Annual IEEE International Computer Software and Applications Conference. — IEEE, 2009. — Pp. 154–159.
154. Middleware for Internet of Things: A Survey / Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, Siobhan Clarke // IEEE Internet of Things Journal. — 2016. — 02. — Vol. 3, no. 1. — Pp. 70–95.
155. Low Level Reader Protocol (LLRP). Version 1.1 Ratified Standard. — EPCGlobal, 2010. — P. 198.
156. Reader Management 1.0.1. — EPCGlobal, 2007. — P. 242.

157. Discovery, Configuration, and Initialization (DCI) for Reader Operations. Version 1.0 Ratified Standard. — EPCGlobal, 2009. — Pp. 1–26.
158. The Application Level Events (ALE) Specification, Version 1.1.1. Part I: Core Specification. — EPCGlobal, 2009. — P. 229.
159. The Application Level Events (ALE) Specification, Version 1.1.1. Part II: XML and SOAP Bindings. — EPCGlobal, 2009. — P. 119.
160. RAIN Reader Communication Interface Guideline. V.4.0. — RAIN RFID Alliance, 2020. — P. 82.
161. Hoag J.E., Thompson C.W. Architecting RFID Middleware // IEEE Internet Computing. — 2006. — 09. — Vol. 10, no. 5. — Pp. 88–92.
162. Yu Jian, Lai Shengli. Message Publish/Subscribe System of RFID Middleware // 2009 International Conference on New Trends in Information and Service Science. — IEEE, 2009. — 06. — Pp. 288–292.
163. Xiu Li, Zhan Xu, Lifang Wu. A design for RFID middleware system extended from ALE specification // 2012 IEEE International Conference on Computer Science and Automation Engineering. — IEEE, 2012. — 06. — Pp. 271–274.
164. Koutsoubelias Manos, Lalis Spyros. Design and Implementation of an Extensible Architecture for the Efficient Remote Access of Simple RFID-Readers // 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications. — IEEE, 2007. — Pp. 1–5.
165. Managing RFID Sensors Networks with a General Purpose RFID Middleware / Ismael Abad, Carlos Cerrada, Jose A. Cerrada, Rubén Heradio, Enrique Valero // Sensors 2012, Vol. 12, Pages 7719-7737. — 2012. — 6. — Vol. 12. — Pp. 7719–7737.
166. An intelligent high available RFID middleware / Chien-Chang Hsu, Hsin Mei, Chun-Sho Lee, Da-Gang Lee // 2008 International Conference on Machine Learning and Cybernetics. — IEEE, 2008. — 07. — Pp. 3166–3171.
167. Floerkemeier Christian, Roduner Christof, Lampe Matthias. RFID Application Development With the Accada Middleware Platform // IEEE Systems Journal. — 2007. — 12. — Vol. 1, no. 2. — Pp. 82–94.
168. dos Santos Yago Luiz, Canedo Edna Dias. On the Design and Implementation of an IoT based Architecture for Reading Ultra High Frequency Tags // Information 2019, Vol. 10, Page 41. — 2019. — 1. — Vol. 10. — P. 41.
169. Aazam Mohammad, Huh Eui-Nam. Fog Computing: The Cloud-IoT\IoE Middleware Paradigm // IEEE Potentials. — 2016. — 05. — Vol. 35, no. 3. — Pp. 40–44.

170. PerLa: A Language and Middleware Architecture for Data Management and Integration in Pervasive Information Systems / F. A. Schreiber, R. Camplani, M. Fortunato, M. Marelli, G. Rota // IEEE Transactions on Software Engineering. — 2012. — 03. — Vol. 38, no. 2. — Pp. 478–496.
171. SARIF: A novel framework for integrating wireless sensor and RFID networks / Jaekyu Cho, Yoonbo Shim, Taekyoung Kwon, Yanghee Choi, Sooyeon Kim // IEEE Wireless Communications. — 2007. — 12. — Vol. 14, no. 6. — Pp. 50–56.
172. Kamoun Faouzi. RFID system management: state-of-the art and open research issues // IEEE Transactions on Network and Service Management. — 2009. — 09. — Vol. 6, no. 3. — Pp. 190–205.
173. Jongyoung Lee, Naesoo Kim. Performance Test Tool for RFID Middleware: Parameters, Design, Implementation, and Features // 2006 8th International Conference Advanced Communication Technology. — IEEE, 2006. — Pp. 149–152.
174. Implementation of RFID Middleware Based on Client-Multiserver Architecture for Traceability of Autoparts / Daniel Henao-Jaramillo, Daniel Alejandro Montoya-Tamayo, Yolanda Alvarez-Rios, Victor Hugo Aristizabal-Tique // IEEE Latin America Transactions. — 2019. — 6. — Vol. 17. — Pp. 930–936.
175. An attribute-based access control using chaincode in RFID systems / Santiago Figueroa, Javier Anorga, Saioa Arrizabalaga, Inigo Irigoyen, Mario Monterde // 2019 10th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2019 - Proceedings and Workshop. — IEEE, 2019. — 6.
176. Zhang Jin Feng, Wen Chang Ji. The university library management system based on radio frequency identification // Proceedings - 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2017. — IEEE, 2018. — 2. — Pp. 1–6.
177. Li Na, Tan Jie, Zhu Zhiyuan. Monitor and control system with RFID technology in discrete manufacturing line // 2010 IEEE International Conference on RFID-Technology and Applications. — IEEE, 2010. — 06. — Pp. 71–76.
178. Shah Shraddha, Singh Bharti. RFID based school bus tracking and security system // 2016 International Conference on Communication and Signal Processing (ICCSP). — IEEE, 2016. — 04. — Pp. 1481–1485.
179. Context aware middleware for RFID based pharmaceutical supply chain / Marwa Chamekh, Sadok El Asmi, Mohamed Hamdi, Tai-Hoon Kim // 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC). — IEEE, 2017. — jun. — Pp. 1915–1920.

180. RFID Application to Airport Luggage Tracking as a Green Logistics Approach / Yasir Rouchdi, Achraf Haibi, Khalid El Yassini, Mohammed Boulmalf, Kenza Oufaska // Colloquium in Information Science and Technology, CIST. — IEEE, 2018. — 12. — Pp. 642–649.
181. Design considerations of RFID based baggage handling system, a literature review / Harjo Baskoro, Harjanto Prabowo, Agung Trisetiyarso, Meyliana, Achmad Nizar Hidayanto // Proceedings of 2017 International Conference on Information Management and Technology, ICIMTech 2017. — IEEE, 2018. — 1. — Pp. 210–214.
182. Thompson Craig W., Thompson Dale R. Identity Management // IEEE Internet Computing. — 2007. — 05. — Vol. 11, no. 3. — Pp. 82–85.
183. БЕЗОПАСНОСТЬ RFID-СИСТЕМ / В.С. Бельский, Е.С. Грибоедова, К.Д. Царегородцев, А.А. Чичаева // International Journal of Open Information Technologies. — 2021. — № 9.
184. Mobile RFID and its design security issues / Ali Bashir, Sajjad Chauhdary, Sayed Shah, Myong-soon Park // IEEE Potentials. — 2011. — 07. — Vol. 30, no. 4. — Pp. 34–38.
185. Fouladgar Sepideh, Afifi Hossam. A Simple Delegation Scheme for RFID Systems (SiDeS) // 2007 IEEE International Conference on RFID. — IEEE, 2007. — Pp. 1–6.
186. Larionov Andrey A., Ivanov Roman E., Vishnevsky Vladimir M. UHF RFID in Automatic Vehicle Identification: Analysis and Simulation // IEEE Journal of Radio Frequency Identification. — 2017. — March. — Vol. 1, no. 1. — Pp. 3–12.
187. Vishnevsky Vladimir, Larionov Andrey, Roman Ivanov. Applying UHF RFID for Vehicle Identification: Protocol and propagation simulation // 2017 IEEE International Conference on RFID (RFID). — Phoenix, AZ: IEEE, 2017. — May. — Pp. 73–80.
188. Vishnevskiy Vladimir, Larionov Andrey, Ivanov Roman. Analysis and Simulation of UHF RFID Vehicle Identification System // Communications in Computer and Information Science. — Vol. 678. — Springer, 2016. — Pp. 35–46.
189. Lavrukhin I. R., Larionov A. A., Yelizarov Andrey Albertovich. ANALYSIS AND MODELING OF THE PROTOCOL OF RADIO FREQUENCY IDENTIFICATION OF VEHICLES ON ROAD STATIONS // 2018 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO). — IEEE, 2018. — July. — Pp. 1–5.
190. Matz Gerald, Hlawatsch Franz. Fundamentals of Time-Varying Communication Channels // Wireless Communications Over Rapidly Time-Varying Channels. — Elsevier, 2011. — Pp. 1–63.
191. Balanis Constantine A. Antenna Theory: Analysis and Design. — 4th edition. — Wiley, 2016. — P. 1104.

192. Larionov Andrey, Ivanov Roman, Vishnevsky Vladimir. A stochastic model for the analysis of session and power switching effects on the performance of UHF RFID system with mobile tags // 2018 IEEE International Conference on RFID (RFID). — Orlando: IEEE, 2018. — Pp. 1–8.
193. Ларионов А.А. О математической модели системы радиочастотной идентификации мобильных объектов, оснащенных RFID-метками // Труды 13-го Всероссийского совещания по проблемам управления (ВСПУ XIII, Москва, 2019). — Москва: ИПУ РАН, 2019. — С. 2953–2957.
194. Review of methodology and design of broadband wireless networks with linear topology / Vladimir Vishnevsky, Achyutha Krishnamoorthy, Dmitry Kozyrev, Andrei Larionov // Indian Journal of Pure and Applied Mathematics. — 2016. — Vol. 47, no. 2. — Pp. 329–342.
195. Методы исследования и проектирования широкополосных беспроводных сетей вдоль протяженных транспортных магистралей / В.М. Вишневский, А. Кришнамурти, Д.В. Козырев, А.А. Ларионов, Р.Е. Иванов // Т-Comm: Телекоммуникации и транспорт. — 2015. — Т. 9, № 5. — С. 9–15.
196. Новое поколение систем безопасности на автодорогах и их применение в интеллектуальных транспортных системах / В.М. Вишневский, Р.Н. Минниханов, А.Н. Дудин, В.И. Клименок, А.А. Ларионов // Информационные технологии и вычислительные системы. — 2013. — Т. 4. — С. 80–89.
197. Вишневский В.М., Ларионов А.А., Семёнова О.В. Оценка производительности высокоскоростной беспроводной тандемной сети с использованием каналов сантиметрового и миллиметрового диапазона радиоволн в системах управления безопасностью дорожного движения // Проблемы управления. — 2013. — Т. 4. — С. 50–56.
198. Вишневский В.М., Ларионов А.А., Целикин Ю.В. Анализ и исследование методов проектирования автоматизированных систем безопасности на автодорогах с использованием новых широкополосных беспроводных средств и RFID-технологий // Т-Comm: Телекоммуникации и транспорт. — 2012. — Т. 7. — С. 48–54.
199. Performance Evaluation of the Priority Multi-Server System MMAP/PH/M/N Using Machine Learning Methods / Vladimir Vishnevsky, Valentina Klimenok, Alexander Sokolov, Andrey Larionov // Mathematics. — 2021. — Vol. 9, no. 24.
200. Vishnevsky Vladimir, Larionov Andrey. Stochastic Multiphase Models and Their Application for Analysis of End-to-End Delays in Wireless Multihop Networks // Applied Probability and Stochastic Processes. Infosys Science Foundation Series. — Springer, Singapore, 2020. — Pp. 457–471.
201. A Multiphase Queueing Model for Performance Analysis of a Multi-hop IEEE 802.11 Wireless Network with DCF Channel Access / Andrey Larionov, Vladimir Vishnevsky, Olga Semenova, Alexander Dudin // Information Technologies and Mathematical Modelling. Queueing

- Theory and Applications. ITMM 2019. Communications in Computer and Information Science. — Vol. 1109. — Springer, Cham, 2019. — Pp. 162–176.
202. State Reduction in Analysis of a Tandem Queueing System with Correlated Arrivals / Vladimir Vishnevsky, Andrey Larionov, Olga Semenova, Roman Ivanov // Communications in Computer and Information Science. — Vol. 800. — Springer, 2017. — Pp. 215–230.
203. Estimation of IEEE 802.11 DCF access performance in wireless networks with linear topology using PH service time approximations and MAP input / Andrey A. Larionov, Vladimir M. Vishnevsky, Roman E. Ivanov, Olga V. Semenova // Proceedings of the 11th IEEE International Conference on Application of Information and Communication Technologies (AICT2017, Moscow). — Vol. 2. — Moscow: IEEE, 2017. — Pp. 85–89.
204. Vishnevski Vladimir, Larionov Andrey, Ivanov Roman. An open queueing network with a correlated input arrival process for broadband wireless network performance evaluation // Communications in Computer and Information Science. — Vol. 638. — Springer, 2016. — Pp. 354–365.
205. Methods of performance evaluation of broadband wireless networks along the long transport routes / Vladimir Vishnevsky, Alexander Dudin, Dmitry Kozyrev, Andrey Larionov // Communications in Computer and Information Science. — Vol. 601. — Springer, 2016. — Pp. 72–85.
206. Дудин А.Н., Листопад Н.И., Царенков Г.В. Улучшенный алгоритм оптимизации работы узла сети интернет // Проблемы проектирования информационно-телекоммуникационных систем. — Pp. 28–43.
207. Horváth Gábor. Moment Matching-Based Distribution Fitting with Generalized Hyper-Erlang Distributions // Analytical and Stochastic Modeling Techniques and Applications. ASMTA 2013. Lecture Notes in Computer Science. — Vol. 7984. — Springer, Berlin, Heidelberg, 2013. — Pp. 232–246.
208. Tarjan Robert. Depth-First Search and Linear Graph Algorithms // SIAM Journal on Computing. — 1972. — Vol. 1, no. 2. — Pp. 146–160.
209. Vishnevsky Vladimir M., Larionov Andrey, Ivanov Roman. Architecture of application platform for RFID-enabled traffic law enforcement system // 2014 7th International Workshop on Communication Technologies for Vehicles, Nets4Cars-Fall 2014. — IEEE, 2014. — Pp. 45–49.
210. Vishnevsky Vladimir M., Larionov Andrey. Design concepts of an application platform for traffic law enforcement and vehicles registration comprising RFID technology // 2012 IEEE International Conference on RFID-Technologies and Applications, RFID-TA 2012. — IEEE, 2012. — Pp. 148–153.



## Список рисунков

1.1	Схема распределённой системы радиочастотной идентификации транспорта . . .	13
1.2	Логическая структура памяти метки . . . . .	20
1.3	Символы в схеме P1E и преамбулы команд RFID-считывателя . . . . .	21
1.4	Схема опроса меток и раунд инвентаризации . . . . .	23
1.5	Организация доступа к каналу при использовании простейшей версии механизма DCF . . . . .	30
1.6	Открытая сеть массового обслуживания с линейной топологией . . . . .	34
2.1	Структура системы радиочастотной идентификации автомобилей . . . . .	43
2.2	Разброс длительностей сообщений между считывателем и меткой. . . . .	45
2.3	Зависимость длительностей ответов EPC+PC+CRC от параметров протокола. . .	46
2.4	Зависимость максимальной длительности раунда от значения Q. . . . .	46
2.5	Зависимость максимального числа раундов, в которых принимает участие метка, от значения Q. . . . .	47
2.6	Схема системы радиочастотной идентификации автомобилей и ее геометрические параметры . . . . .	49
2.7	Расчет бюджета соединения в зависимости от расстояния между считывателем и меткой. . . . .	50
2.8	Зависимость мощности сигналов, принятых меткой и считывателем, от расстояния и времени . . . . .	52
2.9	Коэффициент отражения и затухание сигнала для параллельной, перпендикулярной и круговой поляризацій . . . . .	53
2.10	Зависимость коэффициента отражения от угла падения для параллельной, перпендикулярной и круговой поляризации при разных значениях относительной диэлектрической проницаемости $\epsilon$ и проводимости $\sigma$ . . . . .	53
2.11	Вероятность битовой ошибки (BER) для разных параметров Tag1 и M . . . . .	54
2.12	Изменение вероятности битовой ошибки (BER) из-за эффекта Доплера. . . . .	55
2.13	Число раундов, в которых участвует метка, в зависимости от длительности работы на одной антенне, для разных стратегий выбора флага сессии. . . . .	57
2.14	Вероятность успешного чтения меток в номерах при различных значениях Tag1 и кодировании ответов M = 4 . . . . .	59
2.15	Вероятность успешного чтения меток в номерах при различных значениях M и Tag1 = 12,5 мкс . . . . .	60
2.16	Влияние эффекта Доплера на вероятность успешного чтения метки при M = 4, Tag1 = 12,5 мкс . . . . .	61
2.17	Вероятность идентификации автомобиля с метками на переднем и заднем номере, идентифицируемого по любой из меток . . . . .	61
3.1	Пример раунда опроса при идентификации по EPCID и TID . . . . .	68
3.2	Валидация модели расчета средней длительности раундов . . . . .	72

3.3	Операции над системой . . . . .	74
3.4	Изменение числа активных меток при выполнении элементарных операций . . . . .	76
3.5	Пример представления раундов в виде операций, и декомпозиция этих операций до элементарных операций . . . . .	79
3.6	Алгоритм построения матрицы перехода $D$ . . . . .	80
3.7	Итерационный алгоритм расчета длительностей раундов . . . . .	83
3.8	Структура пространства состояний основного процесса $\{\gamma_r\}$ , когда в системе $N$ меток . . . . .	86
3.9	Переходные вероятности основного процесса $\{\gamma_r\}$ при выполнении операции инвентаризации $V_N^\nabla$ . . . . .	87
3.10	Переходные вероятности основного процесса $\{\gamma_r\}$ при выполнении элементарных операции . . . . .	88
3.11	Оценки характеристик раундов . . . . .	95
3.12	Сравнение оценок вероятности идентификации, полученных с помощью аналитической и имитационной модели . . . . .	97
3.13	Вероятности идентификации в различных сценариях . . . . .	98
3.14	Относительные ошибки оценки вероятности идентификации . . . . .	99
3.15	Абсолютные ошибки оценки вероятности идентификации . . . . .	99
4.1	Беспроводная сеть с линейной топологией и ее аналитическая модель . . . . .	102
4.2	Пример представления РН-распределения в виде МАР-потока . . . . .	105
4.3	Сети массового обслуживания с кросс-трафиком и без него . . . . .	108
4.4	РН-распределения для аппроксимации потоков по двум моментам . . . . .	118
4.5	Ациклическое РН-распределение АСРН(2) и гиперэрланговское распределение $ME_n(2)$ , используемые при аппроксимации потоков по трем моментам . . . . .	120
4.6	Области существования РН-распределений. Зеленым цветом показана область существования АСРН(2) [142], розовым – произвольных РН-распределений, в частности $ME_n(2)$ [111]. Ниже линии $\gamma = c - 1/c$ РН-распределений не существует . . . . .	120
4.7	Области видимости станций многошаговой беспроводной сети . . . . .	125
4.8	Схема моделирования каналов беспроводной сети . . . . .	126
4.9	Схема моделирования беспроводной сети с произвольным числом станций . . . . .	127
4.10	Примеры отсеиваемых МАР-потоков. На стрелках показана интенсивность переходов. Сплошные линии — переходы матрицы $D_1$ , пунктирные — $D_0$ . . . . .	128
4.11	Разбиение наборов входных данных на простые и сложные . . . . .	129
4.12	Диаграммы разброса для различных сочетаний параметров в наборах простых входных данных . . . . .	130
4.13	Разброс оценок, полученных аппроксимацией потоков экспоненциальным распределением . . . . .	131
4.14	Разброс оценок, полученных аппроксимацией потоков РН-распределениями по двум моментам . . . . .	132

4.15	Разброс оценок, полученных аппроксимацией потоков РН-распределениями по трем моментам . . . . .	133
4.16	Разброс оценок, полученных аппроксимацией МАР-потоков МАР-потоками по трем моментам и коэффициенту корреляции . . . . .	134
4.17	Разброс оценок, полученных методом Монте-Карло . . . . .	134
4.18	Точность оценок, полученных с помощью методов аппроксимации потоков и метода Монте-Карло . . . . .	135
4.19	Изменение коэффициентов корреляции и вариации в выходном потоке для системы МАР/РН/1/М . . . . .	136
4.20	Изменение коэффициентов корреляции и вариации в выходящих потоках на разных узлах сети . . . . .	136
4.21	Сравнение оценок, полученных методами Монте-Карло и аппроксимацией только выходящих потоков РН-распределениями по трем моментам . . . . .	137
4.22	Длительность расчета точного решения, а также расчетов с помощью различных методов аппроксимации выходящих потоков и метода Монте-Карло . . . . .	137
4.23	Сходимость метода Монте-Карло и скорость выполнения расчетов . . . . .	138
4.24	Распределение длительностей передач пакетов в каналах калибровочной сети для $\lambda = 500$ . . . . .	140
4.25	Средние длительности передач, коэффициенты вариации и асимметрии для каналов калибровочной сети . . . . .	141
4.26	Межконцевые задержки в калибровочной сети . . . . .	141
4.27	Межконцевые задержки в сети из 10 станций, рассчитанные с помощью оригинальной методики . . . . .	142
4.28	Характеристики распределений длительностей передачи пакетов в каналах беспроводной сети из 10 станций . . . . .	142
4.29	Улучшенная схема моделирования беспроводной сети с произвольным числом станций . . . . .	142
4.30	Межконцевые задержки в сети из 10 станций, рассчитанные с помощью улучшенной методики . . . . .	143
4.31	Ошибки оценок межконцевых задержек беспроводной сети с помощью моделей сетей массового обслуживания . . . . .	143
5.1	Компоненты системы управления . . . . .	146
5.2	Примеры размещения модулей системы . . . . .	148
5.3	Протоколы передачи данных и управления . . . . .	149
5.4	Работа супервайзера в режиме IMMP-прокси . . . . .	150
5.5	Пример IMMP-сессии . . . . .	151
5.6	Пример SUAP-сессии . . . . .	152
5.7	Пример соединения ИТОР между веб-интерфейсом и RFID-адаптером . . . . .	153
5.8	Пример создания подписок в соединении ИТОР . . . . .	154
5.9	Пример подключения клиента по протоколу TFP . . . . .	156

5.10	Работа TFP-сервера при создании подписок от двух клиентов . . . . .	157
5.11	Архитектура супервизора SVRD . . . . .	158
5.12	Архитектура RFID-адаптера G2RD . . . . .	159
5.13	Аппаратная структура RFID-считывателя . . . . .	163
5.14	Схема размещения считывателей и программных компонентов распределенной системы в Казани в 2014 году . . . . .	166
5.15	Схема размещения RFID-оборудования в Казани в 2020 году . . . . .	168
5.16	Схема клиентского программного обеспечения для испытаний . . . . .	169

## Список таблиц

1	Технологии, используемые в распределённой системе радиочастотной идентификации транспорта, разработанной в рамках диссертационной работы. . .	15
2	Параметры, использованные при расчете бюджета соединения . . . . .	50
3	Среднее число меток, участвующих в раунде, при движении автомобилей со скоростью 60 км/ч . . . . .	59
4	Параметры аналитической модели RFID . . . . .	93
5	Порядки выходящих MAP-потокa в зависимости от порядков PH-распределения ( $V$ ), входящих MAP-потокa ( $W$ ) и емкости очереди ( $M$ ) . . . . .	112
6	Граничные значения параметров для исследования методов аппроксимации потокa . . . . .	128
7	Параметры беспроводных каналов и трафика для эксперимента . . . . .	139
8	Параметры RFID-считывателей в Казани в 2014 году. . . . .	167

**Приложение А**

**Акты о внедрении**

**АКТ**  
**внедрения результатов диссертационной работы**  
**«Технология построения и методы исследования систем управления**  
**безопасностью дорожного движения на основе широкополосных**  
**беспроводных сетей и радиочастотной идентификации»**  
**Ларионова Андрея Алексеевича,**  
**представленной на соискание ученой степени**  
**кандидата технических наук**

Результаты диссертационной работы Ларионова А.А. были использованы при выполнении НИР «Знак-Метка», выполненного совместно с ФКУ НПО «СТиС» МВД России, ПАО «Микрон», ИПУ РАН и ОАО «Авангард». НИР была разработана в соответствии с дорожной картой, утвержденной заместителем Председателя Правительства Российской Федерации Ю.И. Борисовым от 4 июня 2019 г. №4966п-П7.

В рамках НИР были использованы разработанные в диссертации модели протокола UHF RFID для оценки эффективности работы RFID-считывателя, разработанного под руководством и при участии Ларионова А.А.

Разработанная в диссертации система управления отечественными RFID-считывателями и результаты исследования в части выбора оптимальных настроек RFID-считывателей эффективно использовались при проведении совместного эксперимента ИПУ РАН и ПАО «Микрон» на центральной кольцевой автомобильной дороге (ЦКАД) в период с сентября по декабрь 2021 года.

Директор  
по прикладным изделиям  
радиочастотной идентификации  
АО «Микрон»



А.Ю. Резник

ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ  
«БЕЗОПАСНОСТЬ  
ДОРОЖНОГО ДВИЖЕНИЯ»  
Оренбургский тракт, д. 5, г. Казань, 420059



«ЮЛ ХӘРӘКӘТЕ  
ИМИНЛЕГЕ»  
ДӘҮЛӘТ  
БЮДЖЕТ УЧРЕЖДЕНИЯСЕ

Оренбург тракты, 5 йорт, Казан шәһәре, 420059

Телефон: (843) 533-37-78, факс: (843) 533-37-98, e-mail: gubdd@tatar.ru, сайт: http://gubdd.tatarstan.ru

## АКТ

**внедрения результатов диссертационной работы  
«Технология построения и методы исследования систем управления  
безопасностью дорожного движения на основе широкополосных  
беспроводных сетей и радиочастотной идентификации»  
Ларионова Андрея Алексеевича,  
представленной на соискание ученой степени  
кандидата технических наук**

Результаты диссертационной работы Ларионова А.А. были использованы при экспериментальном внедрении крупномасштабной системы радиочастотной идентификации транспортных средств в городе Казань.

В рамках эксперимента использовались разработанные в диссертации аналитические и имитационные модели для оценки производительности и выбора оптимальных настроек отечественного RFID-считывателя, в создании которого Ларионов А.А. принимал активное участие.

Ларионов А.А. являлся также одним из основных участников эксперимента по внедрению системы радиочастотной идентификации транспортных средств; им разработано программное обеспечение для управления RFID-считывателями и сбора статистических данных о транспортных средствах, оборудованных номерными знаками с RFID-метками.

Директор  
д.т.н., проф., член-корреспондент  
АН Республики Татарстан



Р.Н. Минниханов