

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 24.М41-мм

Михайлов Михаил Дмитриевич

Обнаружение разладки в контроле перегрузки сети

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
доцент кафедры системного программирования, к. ф.-м. н. Гориховский В. И.

Санкт-Петербург
2026

Содержание

Введение	3
Постановка задачи	4
Обзор	5
Обзор используемых технологий	5
Обзор генераторов синтетических данных	5
1 Описание решения для генерации синтетических данных	8
1.1 Архитектура генератора синтетических данных	8
1.2 Процедура моделирования	9
1.3 Валидация генератора	10
2 Систематичный обзор методов обнаружения разладки в сетевых данных	10
2.1 Цель и методика обзора	10
2.2 Выявленные подходы	12
2.3 Выводы из обзора	13
Заключение	14

Введение

Современные транспортные протоколы используют алгоритмы контроля перегрузки (ССА) для эффективной передачи данных. Традиционные подходы, основанные на факте потери пакетов, плохо работают в беспроводных сетях, в частности из-за того что потери могут быть вызваны не только перегрузкой, но и радиопомехами. Современные алгоритмы, такие как BBR, строят модель сети, анализируя её метрики. Одним из подходов к анализу метрик является обнаружение разладки — момента скачкообразного изменения статистических свойств потока данных.

Существующие методы обнаружения разладки часто плохо применимы к сетевым данным из-за нереалистичных предположений о независимости и распределении наблюдаемых метрик. Кроме того, отсутствие специализированных размеченных данных затрудняет разработку и сравнение новых алгоритмов. Это создаёт потребность в новых решениях, адаптированных к специфике сетевого трафика.

Работа выполняется в рамках совместного проекта СПбГУ и Huawei¹, направленного на создание эффективного онлайн-алгоритма обнаружения разладки для сетевых данных. Разработка такого алгоритма требует комплексного подхода, включающего анализ существующих методов, создание инструментов для валидации и проведения экспериментального исследования.

Работа направлена на создание инструмента генерации данных и онлайн-алгоритма обнаружения разладки для трафика QUIC, в рамках технического задания выданного со стороны Huawei.

¹Проект №ТС20250703044

Постановка задачи

Целью работы является разработка онлайн-алгоритма обнаружения разладки в сетевых данных. Для её выполнения в рамках технического задания были поставлены следующие задачи:

1. разработать генератор синтетических данных, имеющих схожие статистические свойства с данными трафика протокола QUIC, где в качестве ССА используется BBR;
2. провести систематический обзор существующих алгоритмов обнаружения разладки в сетевых данных;
3. разработать и реализовать онлайн-алгоритмы обнаружения разладки, учитывающие специфику сетевого трафика;
4. провести экспериментальное исследование предложенной реализации для выбора наилучшего алгоритма.

Настоящий отчёт содержит результаты по первым двум задачам, поскольку отвечающие им этапы в рамках проекта завершились в декабре. Оставшиеся две задачи, будут представлены в следующем семестре — соответствующие этапы в рамках проекта еще не завершены.

Обзор

Обзор используемых технологий

Для реализации проекта выбран язык программирования Python, что соответствует техническому заданию. Научные вычисления и обработка данных построены на стеке NumPy, SciPy, Matplotlib и Seaborn. NumPy выбран для работы с массивами данных благодаря большому опыту команды в его использовании и необходимости интеграции с другими компонентами проекта. SciPy применяется для статистических расчётов и работы с распределениями, а Matplotlib и Seaborn — для визуализации результатов.

Обзор генераторов синтетических данных

Для разработки и тестирования алгоритмов обнаружения разладки необходимы размеченные данные, обладающие статистическим сходством с реальными метриками сети (RTT, OWD, потери пакетов) при работе QUIC с алгоритмом контроля перегрузки BBR. Поскольку сбор и ручная разметка реальных данных затруднены, в техническом задании поставлена задача разработки генератора синтетических данных. В рамках технического задания есть ряд требований к инструментам, которые можно использовать в качестве основы:

1. генератор должен уметь генерировать многомерные потоки метрик (в частности RTT, OWD, потери) согласованным образом, т.е. учитывая взаимосвязь метрик;
2. генератор должен запускаться локально, без тестового стенда;
3. генератор не может использовать ПО с копилефтной лицензией.

Математические модели и специализированные генераторы

Наиболее близкими к задаче являются специализированные генераторы, которые создают временные ряды с заранее заданными точками изменения. Примером является скрипт `rtt_gen`, разработанный в рамках работы Shao et al. (2017). Он генерирует искусственные ряды RTT, имитируя два типа изменений: скачкообразные смены уровня (из-за смены топологии) и длительные отклонения из-за перегрузки. Генератор помечает моменты разладки, что позволяет использовать его для оценки алгоритмов обнаружения. Однако он ограничен только одной

метрикой (RTT), что не позволяет его использовать для генерации сразу нескольких метрик, как этого требует техническое задание.

Инструменты для тестирования QUIC

Существуют платформы для тестирования производительности QUIC. Среди таких, инструмент `quic-test` единственный не требующий запуска тестового стенда. Он позволяет эмулировать различные сетевые условия (потери, задержки, пропускную способность) и собирать метрики в реальном времени и инструмент обеспечивает высокую правдоподобность данных, так как использует реальную реализацию QUIC. Однако его основная цель — нагрузочное тестирование, и генерация размеченных данных с контролируруемыми моментами разладки в этом инструменте невозможна.

Сетевые симуляторы (ns-3, Mininet)

Дискретно-событийные симуляторы, такие как ns-3, позволяют детально моделировать работу сетевых протоколов. Для ns-3 существуют реализации QUIC (модуль `quic-ns-3`). Для Mininet есть инструмент `mininet-quic`. Такие симуляторы обеспечивают требуемую гибкость и могут воспроизводить сложные сценарии разладки за счёт изменения параметров каналов и очередей. Однако

- `quic-ns-3` распространяется под GPLv2;
- `mininet-quic` является проприетарным.

Поэтому оба инструмента не могут быть задействованы в проекте.

Общие генераторы синтетических временных рядов

В литературе по анализу временных рядов представлены многочисленные методы генерации синтетических данных (например, на основе случайных блужданий, авторегрессионных моделей, GAN) Эти методы позволяют генерировать многомерные ряды со сложными корреляционными структурами. Однако они, как правило, не учитывают особенности сетевых метрик (например, наличие как непрерывных и дискретных метрик одновременно) и таким образом, вопрос о том корректно ли их применение требует отдельного изучения, выходящего за рамки проекта.

Выводы

Проведённый обзор показывает, что готового инструмента, который бы одновременно удовлетворял всем требованиям задачи, не существует. Было принято решение написать свой генератор.

1. Описание решения для генерации синтетических данных

1.1. Архитектура генератора синтетических данных

Для реализации генератора, удовлетворяющего требованиям проекта, была разработана архитектура на основе дискретно-событийного моделирования. Основным принципом стало разделение системы на автономные активные компоненты (акторы), взаимодействующие посредством сообщений-событий. Архитектура решения состоит из следующих основных компонентов (рис. 1):

- **Generator** — координирует выполнение симуляции, управляет конфигурациями сети и переключением между ними в моменты разладки.
- **Актеры (Actor)** — моделируют сущности сети (отправитель, буфер, роутер).
- **События (Event)** — передают изменения состояния пакетов между акторами.

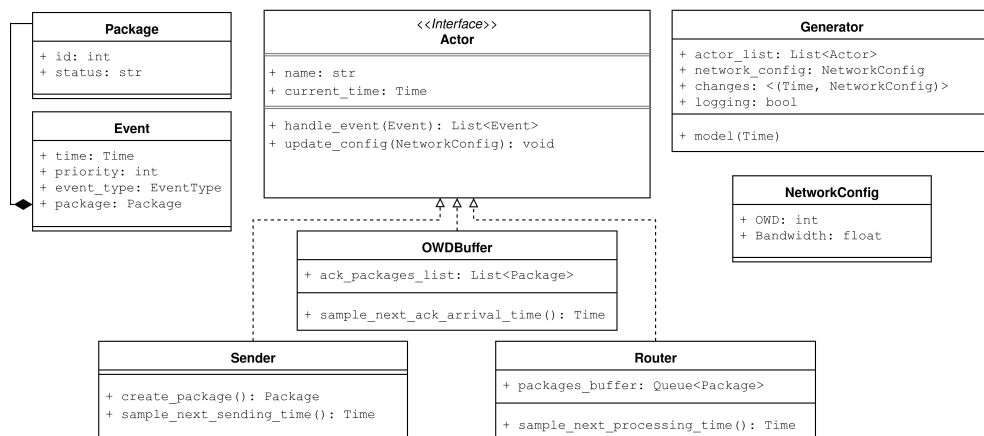


Рис. 1: Диаграмма классов архитектуры генератора

Для моделирования случайных времен обработки пакетов в узлах сети введена система фабрик распределений (через интерфейс `DistributionFactory`). Каждая фабрика отвечает созданию непрерывного вероятностного распределения, параметризованного интенсивностью событий. Использование фабрик распределений позволяет конфигурировать меняющиеся статистические свойства генерируемых данных, сохраняя воспроизводимость (за счёт фиксации `seed`), но расширяя полноту сценариев.

Все активные компоненты сети наследуются от базового класса **Actor**, который определяет интерфейс обработки событий и управления очередью событий. Каждый актор реализует таблицу обработчиков событий (`create_handlers_table()`) и методы для работы с событиями:

- Актор **Sender** моделирует источник трафика, реализуя логику генерации пакетов и обработки подтверждений. Он ведёт полную историю всех пакетов, что позволяет впоследствии рассчитывать метрики сети (RTT, BIF, loss rate). Также в этом акторе эмулируется работа алгоритма контроля перегрузки BBR в стационарном режиме (probe-фаза).
- Актор **OWDBuffer** моделирует одностороннюю задержку (One-Way Delay) и буферизацию на пути от отправителя к маршрутизатору и обратно. Он учитывает пропускную способность канала, размер буфера и время распространения сигнала.
- Актор **Router** представляет собой сетевой узел с ограниченной пропускной способностью и очередью пакетов. Он моделирует обработку пакетов с заданным распределением времени обслуживания, что позволяет имитировать различные сценарии нагрузки и перегрузки.

1.2. Процедура моделирования

На рис. 2 представлена диаграмма активностей классов **Generator**, **Sender**, **OWDBuffer**, **Router** при выполнении моделирования.

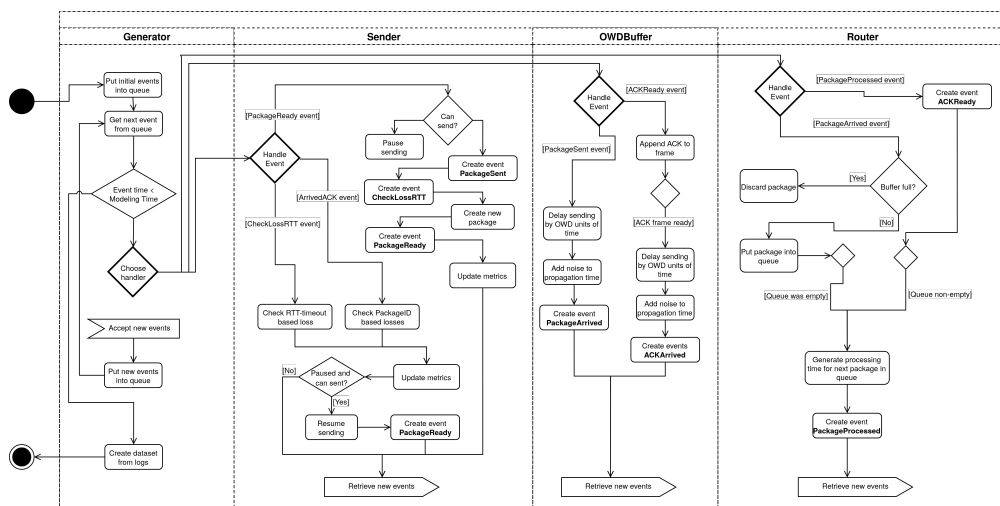


Рис. 2: Диаграмма активностей основных сущностей для генерации

Разладка в генерируемых данных моделируется двумя способами:

- Изменение параметров распределений в узлах сети. Например, увеличение среднего времени обработки пакета для имитации перегрузки.
- Изменение структурных параметров сети: пропускная способность сети, размер буфера, односторонней задержки.

Отметим что эти способы не являются взаимоисключающими: так, изменение времени обработки пакета приводит к изменению пропускной способности сети. Моменты разладки задаются в конфигурации генератора в виде временных меток, при наступлении которых активная конфигурация сети заменяется на новую. Это позволяет создавать сложные сценарии с множественными изменениями характеристик сети.

1.3. Валидация генератора

Пример использования генератора для создания ряда RTT с тремя точками разладки представлен на рис. 3. Полученные данные включают временные ряды метрик (RTT, ROWD, потери, скорость отправки и т.п.) и точную разметку моментов изменения. По итогам валидации, другие участники коллектива предложили методологию подборки параметров распределений в акторах, обеспечившей сходство (в рамках технического задания) генерируемых данных с образцовыми данными, полученных от Huawei.

2. Систематичный обзор методов обнаружения разладки в сетевых данных

Авторские права на полный текст обзора сделанного в рамках проекта принадлежат Huawei. В настоящем разделе кратко описана методика обзора и основные выводы.

2.1. Цель и методика обзора

Целью обзора является анализ подходов к обнаружению разладки, применяемых в сетевом трафике, и выделение методов, перспективных для онлайн-обнаружения в контексте контроля перегрузки QUIC/BBR.

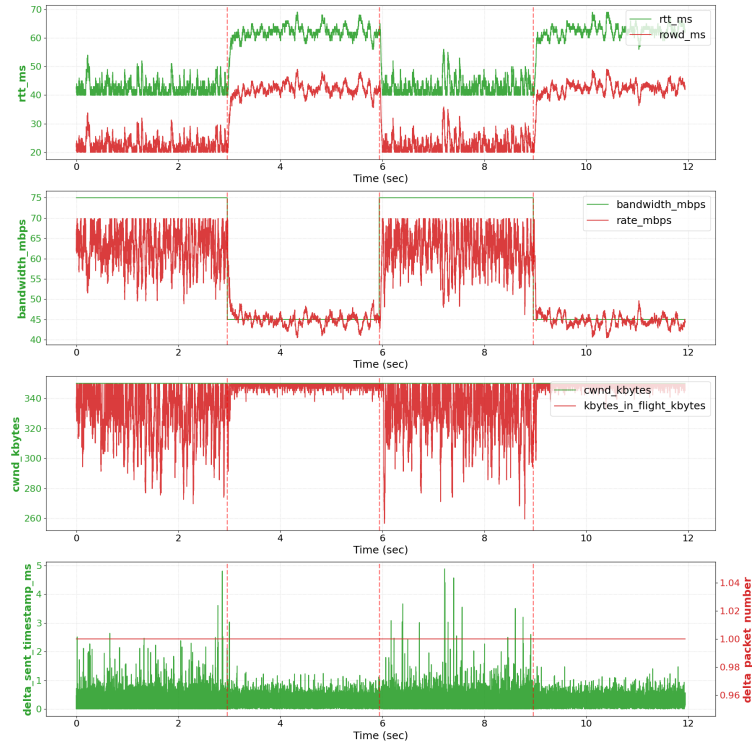


Рис. 3: Пример выходных данных генератора: временные ряды метрик *RTT* и *OWD* с отмеченными точками разладки

Поиск проводился в GOOGLE SCHOLAR, ARXIV и IEEE XPLORE по запросам, связанным с обнаружением разладки в сетевых данных. Рассматривалось три категории источников: статьи/книги, обзорные статьи/книги, экспериментальные исследования:

- Для каждой категории источников был проведен поиск в системах: GOOGLE SCHOLAR, GOOGLE BOOKS, ARXIV
- Для каждой пары “категория” – “поисковая система” были сформулированы поисковые запросы
- При первичном отборе просматривались аннотация и ключевые слова

Результаты приведены в таблице 1 Из более чем 130 первоначально

Вид источника	Отобрано	Просмотрено
Тематические статьи	16	>130
Обзорные статьи	9	11
Экспериментальные исследования	1	30

Таблица 1: Сводная статистика по отобранным источникам

найденных работ для детального анализа было отобрано 26.

2.2. Выявленные подходы

Рассматривались следующие алгоритмы:

- **Контрольная карта Шухарта (Shewhart Control Chart)** : отслеживает отклонение скользящего среднего последних m наблюдений от целевого значения μ_0 . Разладка регистрируется, если отклонение превышает контрольные пределы, определяемые стандартным отклонением процесса σ и коэффициентом-параметром L . Вычислительная сложность: $O(1)$ по времени на наблюдение, $O(m)$ по памяти.
- **Параметрический алгоритм CUSUM [2]**: основан на накоплении сумм логарифмов отношений правдоподобий между распределениями до (f_0) и после (f_1) разладки. Статистика S_n обновляется рекуррентно, разладка объявляется при превышении порога h . Параметры: f_0, f_1, h . Вычислительная сложность: $O(1)$ по времени и памяти.
- **Непараметрический алгоритм CUSUM [2]**: обнаруживает сдвиг среднего относительно целевого значения μ_0 , используя две рекуррентные статистики для положительного и отрицательного отклонений. Параметры: ожидаемый размер сдвига δ и порог h . Вычислительная сложность: $O(1)$ по времени и памяти.
- **Байесовский онлайн-алгоритм обнаружения разладки [1]**: вычисляет апостериорное распределение длины серии $P(r_n | x_{1:n})$ — количества наблюдений с последней разладки. Решение принимается на основе этого распределения, например, при превышении порога h . Параметры: априорное распределение разладки (обычно геометрическое с параметром p), порог h . Вычислительная сложность: $O(n^2)$ по времени и памяти (может быть оптимизирована до $O(m^2)$ при усечении).
- **Алгоритм SDAR (Sequentially Discounting Auto-Regression) [3]**: адаптивно обучает авторегрессионную модель $AR(m)$ с дисконтирующим фактором r , постепенно забывая старые данные. Разладка обнаруживается как аномалия в скользящем среднем от логарифмической правдоподобности новых наблюдений. Параметры: порядок авторегрессии m , фактор дисконтирования r , размер окна усреднения T . Вычислительная сложность: $O(m^3)$ по времени и $O(m^2)$ по памяти в наивной реализации.

- **Алгоритм на основе SSA:** использует метод SSA для построения сигнального подпространства по скользящему окну из m наблюдений. Статистика разладки вычисляется как сумма квадратов расстояний векторов тестовой выборки до этого подпространства, на которую накладывается CUSUM-статистика. Параметры: длина окна m , размер лага k , количество собственных векторов l , параметры тестовой выборки (p, q) , порог h . Вычислительная сложность: $O(mk \log k)$ по времени на наблюдение, $O(m^2)$ по памяти.

2.3. Выводы из обзора

Анализ показал, что существующие методы не полностью отвечают требованиям задачи онлайн-обнаружения разладки в сетевом трафике QUIC. Большинство алгоритмов либо имеют слишком сильные предположения о распределении данных, которые не выполняются для данных полученных от Huawei, либо обладают вычислительной сложностью, превышающую установленную в техническом задании. При этом в существующей литературе недостаточно освещены вопросы работы с многомерными коррелированными сетевыми метриками в режиме реального времени. На основании обзора были отобраны алгоритмы, которые будут включены в экспериментальное сравнение относительно разрабатываемого алгоритма.

Заключение

В ходе выполнения работы, соответствующей первому этапу кооперационного проекта СПбГУ и Huawei, достигнуты следующие результаты, соотносимые с поставленными задачами:

1. **Разработан генератор синтетических данных**, обладающий статистическим сходством с метриками трафика протокола QUIC при использовании алгоритма контроля перегрузки BBR. Генератор реализован на основе архитектуры дискретно-событийного моделирования. Инструмент позволяет воспроизводимо генерировать размеченные временные ряды (RTT, OWD, потери пакетов) с задаваемыми сценариями разладки в соответствии с техническим заданием.
2. **Проведён систематический обзор** существующих методов обнаружения разладки в сетевых данных. Обзор выполнен в научных базах (Google Scholar, arXiv, IEEE Xplore) в соответствии с техническим заданием. В результате обзора были проанализированы и классифицированы более 130 работ.
3. **Разработка и реализация онлайн-алгоритмов обнаружения разладки** является задачей действующего этапа проекта и будет представлена в отчёте за следующий семестр.
4. **Экспериментальное исследование и выбор наилучшего алгоритма** запланировано на следующий этап проекта и будет выполнено после реализации алгоритмов-кандидатов.

Исходный код разработанного генератора синтетических данных и полный текст систематического обзор проходит процедуру экспортного контроля и в последствии будет передан Huawei.

Список литературы

1. *Adams R. P., MacKay D. J. C.* Bayesian Online Changepoint Detection. — 2007. — arXiv: [0710.3742 \[stat.ML\]](https://arxiv.org/abs/0710.3742). — URL: <https://arxiv.org/abs/0710.3742>.
2. *Wang H., Xie Y.* Sequential change-point detection: Computation versus statistical performance // Wiley Interdisciplinary Reviews: Computational Statistics. — 2024. — Jan. 1. — Vol. 16, no. 1. — e1628. — ISSN 1939-0068. — DOI: [10.1002/wics.1628](https://doi.org/10.1002/wics.1628). — URL: <https://wires.onlinelibrary.wiley.com/doi/10.1002/wics.1628> (visited on 11/27/2025) ; Publisher: John Wiley & Sons, Ltd.
3. *Yamanishi K., Takeuchi J.-i.* A unifying framework for detecting outliers and change points from non-stationary time series data // Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. — ACM, 07.2002. — C. 676—681. — (KDD02). — DOI: [10.1145/775047.775148](https://doi.org/10.1145/775047.775148). — URL: <http://dx.doi.org/10.1145/775047.775148>.