



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ7 «Программное обеспечение ЭВМ и информационные технологии»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ
НА ТЕМУ:**

*«Метод реализации сетевой архитектуры кластера
машинного обучения с использованием механизмов
контроля перегрузок»*

Студент группы ИУ7-73Б

(Подпись, дата)

Паламарчук А.Н.

(Фамилия И.О.)

Руководитель

(Подпись, дата)

Клочков М.Н.

(Фамилия И.О.)

2025 г.

РЕФЕРАТ

Расчетно-пояснительная записка 25 с., 2 табл., 17 ист., 1 прил.

Ключевые слова: методы контроля перегрузок, методы балансировки нагрузки, моделирование сетей кластера машинного обучения.

Цель данной работы — провести анализ существующих методов и программных средств для реализации сетей кластера машинного обучения с использованием механизмов контроля перегрузок и балансировки нагрузки.

В данной научно-исследовательской работе будут рассмотрены различные существующие методы контроля перегрузок, балансировки нагрузки и моделирования сетей кластера машинного обучения, а также будет проведен сравнительный анализ по различным критериям.

СОДЕРЖАНИЕ

РЕФЕРАТ	3
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	6
ВВЕДЕНИЕ	7
1. Анализ предметной области	8
1.1. Специфика трафика для обучения нейронных сетей	8
1.2. Проблема перегрузок в сетях ЦОД	9
1.3. Требования к сетевой инфраструктуре ML-кластеров	9
1.4. Особенности коммутаторов ЦОД	10
2. Анализ методов контроля перегрузок	11
2.1. Методы управления потоком	11
2.2. Реактивные методы на основе ECN	12
2.3. Методы на основе измерения задержки RTT	12
2.4. Методы на основе внутрисетевой телеметрии INT	13
2.5. Комбинированные и предсказательные методы	13
2.6. Сравнение методов контроля перегрузок	14
3. Анализ методов балансировки нагрузки	16
3.1. Статическая балансировка и проблема коллизий ЕСМР	16
3.2. Динамическая балансировка на уровне пакетов	16
3.3. Динамическая балансировка на уровне групп пакетов	17
3.4. Сравнение методов балансировки нагрузки	18
4. Анализ методов и средств моделирования сетей	19
4.1. Подходы к моделированию сетей ЦОД	19
4.2. Анализ применимости существующих симуляторов	19
4.3. Особенности моделирования перегрузок и балансировки	20

4.4. Требования к процессу моделирования	21
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	23
ПРИЛОЖЕНИЕ А	25

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

GPU — graphics processing unit.

RDMA — remote direct memory access.

RoCEv2 — RDMA over Converged Ethernet version 2.

INT — in-band network telemetry.

HPCC — high precision congestion control.

ECN — explicit congestion notification.

RTT — round trip time.

PFC — priority-based flow control.

P4 — programming protocol-independent packet processors.

EFC — ethernet flow control.

DCQCN — data center quantized congestion notification.

ACK — acknowledgment packet.

FNCC — fast notification congestion control.

RoCC — robust congestion control.

SLO — service level objective.

ECMP — equal cost multi path.

ВВЕДЕНИЕ

В данной научно-исследовательской работе будут рассмотрены различные существующие методы контроля перегрузок, балансировки нагрузки и моделирования сетей кластера машинного обучения, а также будет проведен сравнительный анализ по различным критериям.

Цель данной работы — провести анализ существующих методов и программных средств для реализации сетей кластера машинного обучения с использованием механизмов контроля перегрузок и балансировки нагрузки.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать предметную область, выявить специфику трафика распределенного обучения, определить требования к сети;
- провести сравнительный анализ методов контроля перегрузок, среди которых выделяются реактивные и проактивные;
- провести сравнительный анализ методов балансировки нагрузки, среди которых выделяются статические и динамические;
- провести сравнительный анализ методов и средств моделирования сетей.

1. Анализ предметной области

1.1. Специфика трафика для обучения нейронных сетей

Современные модели машинного обучения, такие как большие языковые модели и рекомендательные системы на основе глубокого обучения, содержат миллиарды и даже триллионы параметров. Быстрое обучение таких моделей требует распределенных вычислений на кластерах, состоящих из тысяч графических процессоров (GPU) [1]. Эффективность этого процесса зависит от сети, объединяющей GPU, поскольку в крупных кластерах коммуникационные расходы могут доминировать над временем вычислений [1].

В отличие от традиционных центров обработки данных, которые обслуживают множество разнородных запросов, кластеры распределенного обучения имеют специфические характеристики. Обучение обычно происходит итеративно, где этапы вычислений чередуются с этапами коммуникации. Коммуникация осуществляется посредством коллективных операций, таких как All-Reduce и All-to-All [2].

Трафик, генерируемый такими операциями, обладает следующими отличительными чертами:

- синхронность — все узлы, участвующие в коллективной операции, начинают передачу данных в одно и то же время. Это создает интенсивные всплески трафика, которые могут перегрузить буферы коммутаторов [2];
- низкая энтропия — в отличие от веб-трафика, состоящего из множества мелких потоков, трафик обучения часто характеризуется небольшим количеством крупных и долгоживущих потоков, которые занимают значительную долю полосы пропускания [1];
- чувствительность к задержкам всех участников — производительность обучения определяется самым медленным участником. Если хотя бы один поток в коллективной операции задерживается из-за перегрузки, все остальные вычислительные узлы вынуждены простаивать в ожидании завершения синхронизации [3].

1.2. Проблема перегрузок в сетях ЦОД

Сетевая перегрузка возникает, когда объем трафика, поступающего на узел сети, превышает его пропускную способность или объем доступных буферов [4]. В контексте кластеров машинного обучения наиболее острой проблемой является трафик «многие-к-одному», возникающий при операциях агрегации данных [4]. Это приводит к лавинообразному заполнению выходных очередей на коммутаторах. Поскольку современные коммутаторы имеют ограниченные размеры буферов, которые не успевают расти пропорционально увеличению скоростей соединений (до 400–800 Гбит/с), буфера быстро переполняются [4]. Это влечет за собой следующие негативные последствия:

- рост задержек — пакеты проводят больше времени в очередях, что увеличивает время обработки;
- потеря пакетов — при отсутствии механизмов предотвращения потерь переполнение буфера ведет к отбрасыванию пакетов. Это критично для производительности, так как требует дорогостоящей по времени повторной передачи [3];
- снижение пропускной способности происходит из-за задержек и потерь, что напрямую увеличивает время итерации обучения модели [2].

1.3. Требования к сетевой инфраструктуре ML-кластеров

Требования к сетевой инфраструктуре определяются не только утилизацией ресурсов, пропускной способностью, заполнением буферов и задержками, но также необходимо обращать внимание и на минимизацию времени выполнения коллективных операций [6].

Для обеспечения высокой пропускной способности и низкой задержки современные кластеры переходят от использования стека TCP/IP к технологиям удаленного прямого доступа к памяти (RDMA). RDMA позволяет переносить данные напрямую между памятью приложений на разных серверах, минуя ядро операционной системы и центральный процессор [1]. Наиболее распространенным протоколом для реализации RDMA в сетях Ethernet является RoCEv2 [1].

Однако протокол RDMA был изначально разработан для сетей без потерь. Он крайне чувствителен к потере пакетов, потеря даже одного пакета может потребовать повторной передачи всей последовательности, что приводит к резкому падению производительности [2, 5]. Следовательно, дополнительным требованием к сетевой инфраструктуре ML-кластера на базе Ethernet является обеспечение гарантий доставки без потерь.

1.4. Особенности коммутаторов ЦОД

Объем буферной памяти — одно из главных ограничений. Размер буферной памяти коммутаторов растет значительно медленнее, чем пропускная способность их портов. При скоростях 400 Гбит/с и выше даже кратковременный всплеск трафика, характерный для ML-нагрузок, может исчерпать доступный буфер. Это делает сеть чувствительной к любым задержкам в реакции алгоритмов контроля перегрузок [4].

Поддержка внутрисетевой телеметрии (INT). Технология INT позволяет встраивать в заголовок проходящего пакета метаданные, такие как текущая длина очереди, время нахождения пакета в коммутаторе и загрузка выходного порта. Алгоритмы, использующие INT (например, НРСС), получают информацию о состоянии сети, что позволяет более эффективно управлять скоростью передачи данных по сравнению с методами, полагающимися на косвенные сигналы (ECN или RTT) [2].

Программируемость конвейера обработки пакетов позволяет выносить часть логики управления трафиком на сетевые устройства. Это открывает возможность для реализации более сложных, управляемых коммутатором алгоритмов контроля перегрузок, таких как RoCC или OFC. Такие подходы позволяют принимать решения на основе локальной информации об очередях, что значительно сокращает время реакции на перегрузку.

Управление потоком на основе приоритетов (PFC) позволяет организовать доставку без потерь. Поддерживается всеми современными коммутаторами ЦОД, однако его реализация на аппаратном уровне останавливает весь трафик

определенного класса приоритета, что неизбежно ведет к блокировке очереди и распространению перегрузки [2, 5].

Таким образом, особенности коммутаторов определяют круг применимых методов: стандартные решения (ECN/PFC) работают на любом оборудовании, но не всегда эффективно, в то время как более современные методы, требующие INT или P4-based, обеспечивают высокую производительность, но требуют наличия более дорогого оборудования.

2. Анализ методов контроля перегрузок

2.1. Методы управления потоком

EFC — метод управления потоком в сетях Ethernet, стандартизирован IEEE 802.3x. Основная идея данного механизма заключается в предотвращении потери пакетов из-за переполнения входного буфера на принимающей стороне путем временной приостановки передачи данных соседним узлом. Когда заполнение принимающего буфера достигает определенного порогового значения, коммутатор отправляет источнику кадр PAUSE. Получив этот кадр, отправитель прекращает передачу всех данных на указанное время. Возобновление передачи происходит либо по истечении таймера, либо после получения кадра с нулевым временем паузы, что сигнализирует об освобождении буфера [15]. Основной проблемой EFC является остановка всего трафика, что приводит к простою вычислительных ресурсов и блокировке потоков, не имеющих отношения к перегрузке.

PFC был разработан для устранения недостатков EFC, стандартизирован IEEE 802.1Qbb. PFC позволяет управлять потоками независимо для восьми различных классов трафика (приоритетов). При возникновении задержки в обработке пакетов определенного класса на стороне приемника, отправителю посыпается сигнал паузы только для трафика с конкретным приоритетом [16]. Это позволяет гарантировать доставку без потерь для трафика обучения, сохраняя возможность сброса пакетов для менее приоритетного трафика. Однако, PFC имеет существенные недостатки:

- блокировка — пакеты, направляющиеся в незагруженные порты, могут быть заблокированы в общей очереди из-за того, что один поток в этой очереди ожидает освобождения своего целевого порта [12];
- распространение перегрузки — паузы могут распространяться от коммутатора к коммутатору, формируя «дерево перегрузки» (congestion tree), что может привести к полной остановке значительной части сети [12];
- взаимные блокировки — в циклических топологиях встречные потоки, заблокированные командами PFC, могут создать ситуацию взаимной блокировки, требующую полной перезагрузки [12].

2.2. Реактивные методы на основе ECN

DCQCN комбинирует элементы ECN и PFC. Коммутаторы маркируют пакеты, если длина очереди превышает заданный порог. Получатель, обнаружив маркировку, отправляет уведомление обратно отправителю, который снижает скорость передачи. Однако, DCQCN имеет существенные недостатки:

- задержка реакции — поскольку уведомление проходит полный цикл (RTT) от коммутатора к получателю и обратно к отправителю, реакция часто запаздывает, что приводит к переполнению буферов и срабатыванию PFC [4];
- сложность настройки — DCQCN требует настройки множества параметров, которые статичны и не адаптируются к динамике трафика [8];
- проблема «потоков жертв» (Victim Flows) — в сценариях со смешанным трафиком DCQCN не различает потоки, создающие перегрузку, и потоки, страдающие от нее. Это приводит к некорректному ограничению не перегруженных потоков, попавших в общую очередь [10].

2.3. Методы на основе измерения задержки RTT

Альтернативный подход заключается в использовании измерения задержки (RTT), как основного сигнала о перегрузке. Примерами таких алгоритмов являются TIMELY и Swift [1, 2]. Основная идея состоит в том, что

рост RTT (по сравнению с базовым значением) напрямую коррелирует с ростом длины очереди на коммутаторах.

Преимущество методов на основе RTT заключается в том, что они не требуют поддержки со стороны коммутаторов (в отличие от ECN). Однако алгоритмы, реагирующие на RTT (например, TIMELY), склонны к нестабильности, что может приводить к дрейфу очередей [3].

Протокол Swift улучшает эту схему, используя разложение задержки на сетевую и хостовую составляющие, однако этот метод по-прежнему полагается на реакцию в рамках RTT, что может быть недостаточно быстро для устранения микро-выбросов трафика, характерных для ML-нагрузок [7].

2.4. Методы на основе внутрисетевой телеметрии INT

Для преодоления неточности сигналов ECN и RTT был разработан класс алгоритмов, использующих внутрисетевую телеметрию (INT). Основным представителем является HPCC [4, 5].

HPCC использует информацию о временной метке, длине очередей и другом, которую коммутаторы добавляют в заголовки пакетов. Это позволяет отправителю рассчитывать свободную полосу пропускания и мгновенно корректировать скорость отправки [2]. HPCC имеет ограничения:

- задержка уведомления: Информация INT доставляется только с пакетами подтверждения (ACK), что занимает около одного RTT. В условиях сетей (400 Гбит/с) за время одного RTT очередь может успеть переполниться [4];
- накладные расходы — добавление метаданных INT в каждый пакет увеличивает размер заголовков, снижая полезную пропускную способность [2].

2.5. Комбинированные и предсказательные методы

PowerTCP вводит концепцию «сетевой мощности» (Network Power), объединяя реакцию на длину очереди (как в ECN) и на изменения очереди (как в TIMELY), что позволяет достичь стабильности при пустых очередях и быстрой реакции на всплески трафика без использования телеметрии INT [3].

FNCC решает проблему задержки уведомления в НРСС. Метод использует пакеты ACK на обратном пути для доставки телеметрии с прямого пути, обеспечивая суб-RTT уведомление. Кроме того, FNCC уведомляет отправителя о количестве конкурирующих потоков, что позволяет быстрее корректировать скорость [4].

RoCC предлагает перейти от управления на хосте к управлению на коммутаторе. Коммутаторы самостоятельно вычисляют скорость для потоков с помощью PI-контроллера и сообщают ее отправителям. Это обеспечивает более высокую стабильность и скорость по сравнению с DCQCN [5].

FastFlow комбинирует сигналы ECN и RTT с механизмом урезания пакетов. Он использует алгоритм QuickAdapt для корректировки при обнаружении потери или урезания пакета, что минимизирует задержки [7].

Методы на основе машинного обучения. GraphCC [8] и PET [9] используют мультиагентное обучение с подкреплением для динамической настройки порогов ECN на коммутаторах в зависимости от текущего характера трафика. Polyphony [11] развивает данную идею, используя предсказательные модели для оптимизации конфигурации сети с целью соблюдения разрешенного уровня задержек (SLO).

2.6. Сравнение методов контроля перегрузок

Сравнение рассмотренных методов приведено в Таблице 2.1.

Таблица 2.1 — Сравнение методов контроля перегрузок

Метод	Критерий	Место принятия решения	Механизм реакции	Требования к оборудованию
DCQCN	ECN	Хост	Снижение скорости передачи	Поддержка ECN
TIMELY/Swift	RTT	Хост	Снижение скорости передачи	Высокая точность таймеров NIC
HPCC	INT	Хост	Управление окном	Поддержка INT
RoCC	Длина очереди	Коммутатор	Явная передача скорости	Программируемые коммутаторы (P4/FPGA)
PowerTCP	Сетевая мощность	Хост	Управление окном	Опционально INT
FNCC	Ускоренный INT	Хост	Управление окном	Поддержка INT, модификация ACK
FastFlow	ECN, RTT, Packet Trimming	Хост	Управление окном	Поддержка Packet Trimming
GraphCC/PET	Статистика очередей (ML)	Коммутатор	Управление окном	Вычислительные ресурсы на коммутаторе

Сравнительная таблица показывает, что выбор метода контроля перегрузок представляет собой компромисс между корректностью получаемого сигнала, скоростью реакции и сложностью аппаратной реализации. Стандартные методы (DCQCN, Swift) уступают комбинированным (FastFlow) и

телеметрическим (HPCC, RoCC) подходам. Однако внедрение последних опять же требует большей производительности и сложности аппаратной реализации.

3. Анализ методов балансировки нагрузки

3.1. Статическая балансировка и проблема коллизий ECMP

Стандартным методом балансировки в сетях ЦОД является Equal-Cost Multi-Path (ECMP). При использовании ECMP коммутатор вычисляет хэш заголовка пакета и на основе этого хэша выбирает один из равнозначных выходных портов. Все пакеты одного потока направляются по одному и тому же пути, что гарантирует сохранение порядка их доставки [1]. Однако при нагрузках машинного обучения ECMP имеет следующие недостатки:

- хэш-коллизии — трафик обучения состоит из небольшого числа крупных потоков. Возникают ситуации, когда несколько таких потоков перегружают соединение, в то время как другие простаивают [12];
- нечувствительность к загрузке — ECMP является статическим методом и не учитывает текущее состояние загрузки соединений. В условиях перегрузки, ECMP продолжает направлять трафик на перегруженные участки [1];
- влияние на хвостовую задержку — из-за коллизий время выполнения коллективной операции (All-to-All или All-Reduce) определяется самым медленным потоком, попавшим в коллизию, что может увеличивать время выполнения задач более чем в 5 раз [12].

3.2. Динамическая балансировка на уровне пакетов

Альтернативой статической привязке потока к пути является Packet Spraying. В этом подходе каждый отдельный пакет потока отправляется по разным доступным путям [6, 13].

Packet Spraying позволяет использовать всю доступную пропускную способность сети, равномерно загружая все соединения. Кроме того, имеет устойчивость к сбоям — при выходе из строя одного из путей теряется только часть пакетов, а не весь поток целиком [12].

Однако Packet Spraying создает серьезную проблему переупорядочивания пакетов (Packet Reordering). Поскольку пути имеют разную длину и задержку, пакеты прибывают к получателю не по порядку. Для традиционных транспортных протоколов (TCP, RoCEv2) это является сигналом о потере пакетов, что вызывает ложные срабатывания механизма повторной передачи и резкое снижение скорости [1, 6].

Решение проблемы переупорядочивания требует либо аппаратной поддержки на приемнике (большие буферы переупорядочивания), либо использования специализированных протоколов. Uno [6] предлагает использовать коды избыточности (erasure coding) совместно с балансировкой на уровне подпотоков (subflow-level load balancing), чтобы минимизировать влияние потерь и переупорядочивания. Протокол EDM [13] развивает данную идею, реализуя планировщик трафика на физическом уровне, что позволяет создавать виртуальные каналы без очередей и переупорядочивания.

3.3. Динамическая балансировка на уровне групп пакетов

Компромиссным решением между ECMP и Packet Spraying является коммутация на основе групп пакетов. Flowlet — это последовательность пакетов одного потока, разделенных паузой, превышающей максимальную разницу задержек между путями в сети [12].

Идея состоит в том, чтобы менять путь только в моменты пауз между группами пакетов. Это позволяет избежать переупорядочивания внутри Flowlet, но при этом динамически перераспределять нагрузку. Однако применимость Flowlet к ML-нагрузкам ограничена:

- отсутствие пауз — трафик RDMA при обучении моделей часто представляет собой непрерывный поток данных, где паузы, необходимые для формирования Flowlet-ов, отсутствуют [1];
- сложность определения порога тайм-аута для выявления flowlet-а является задачей, которая зависит от топологии сети. Слишком маленький тайм-аут ведет к переупорядочиванию, иначе — к вырождению в ECMP [12].

Алгоритм Hopper [1] предлагает усовершенствованный подход, используя виртуальные flowlets, границы которых определяются не паузами, а количеством переданных байт или временем RTT, что позволяет балансировать даже непрерывный трафик RDMA.

3.4. Сравнение методов балансировки нагрузки

Сравнение рассмотренных методов приведено в Таблице 3.1.

Таблица 3.1 — Сравнение методов балансировки нагрузки

Метод	Критерий	Место принятия решения	Гранулярность	Переупорядочивание
ECMP	Хэш заголовков	Коммутатор	Поток	Отсутствует
Packet Spraying	Round-Robin/Random	Хост или коммутатор	Пакет	Возникает
Flowlet	Длина очереди	Хост или коммутатор	Группа пакетов	Минимизировано
Hopper	RTT	Хост	RTT-эпоха (группа пакетов)	Контролируется

Сравнительная таблица показывает, что выбор метода контроля перегрузок представляет собой компромисс между эффективностью утилизации полосы пропускания и проблемой переупорядочивания пакетов. ECMP гарантирует доставку пакетов по порядку, но неэффективно использует ресурсы сети при ML-нагрузках. Packet Spraying обеспечивает более качественную балансировку, но требует борьбы с переупорядочиванием. Flowlet и его вариации (например, Hopper) представляют собой гибридные решения, стремящиеся объединить преимущества обоих подходов, но их эффективность сильно зависит от характера трафика и корректности настроек.

4. Анализ методов и средств моделирования сетей

4.1. Подходы к моделированию сетей ЦОД

Существует три основных подхода к моделированию сетей.

Натурное исследование — использование реальных коммутаторов и серверов. Этот метод ограничен размером доступного кластера.

Эмуляция — запуск сетевого стека и ПО коммутаторов в виртуальной среде. Этот подход не позволяет моделировать поведение сети в крупных масштабах из-за ограничений CPU.

Симуляция — создание программной модели поведения сети. Симуляторы делятся на два класса:

Пакетные симуляторы моделируют движение каждого отдельного пакета. Обеспечивают высокую точность, но требуют значительных вычислительных ресурсов.

Симуляторы уровня потоков стали альтернативой пакетным симуляторам для моделирования сетей ЦОД. Современные data-центры насчитывают сотни тысяч серверов, каналов связи и коммутаторов. Поскольку скорости передачи данных продолжают расти, моделирование таких масштабных сетей на уровне отдельных пакетов стало практически невозможным [14].

4.2. Анализ применимости существующих симуляторов

В моделировании сетей ЦОД наиболее часто используются следующие инструменты:

NS-3 — это дискретно-событийный симулятор, который позволяет моделировать широкий набор протоколов с высокой детализацией. В работах [1, 2, 3] NS-3 использовался для оценки алгоритмов Hopper, DCQCN, TIMELY и PowerTCP. Главным преимуществом NS-3 является его расширяемость и наличие моделей для RDMA/RoCEv2, что критически важно для ML-задач.

Однако, как указывается в [14], масштабируемость NS-3 ограничена — симуляция крупных топологий требует значительного времени.

OMNeT++ — среда моделирования, чаще всего используемая для моделирования беспроводных и оптических сетей. В работе [4] OMNeT++ использовался для оценки алгоритма FNCC. Предоставляет средства визуализации и анализа, но база готовых моделей для специфических протоколов ЦОД (таких как RoCEv2 или PFC) менее обширная, чем у NS-3.

ASTRA-SIM — симулятор для распределенного обучения. Специализируется на моделировании поведения рабочих нагрузок ML (коллективных операций), но использует упрощенные сетевые модели или интегрируется с NS-3 для повышения точности [2].

m4 (Learned Flow-level Simulator) — симулятор, который использует машинное обучение для предсказания поведения потоков. Он пытается объединить точность пакетных симуляторов со скоростью потоковых, обучаясь на данных из NS-3 [14].

4.3. Особенности моделирования перегрузок и балансировки

Моделирование сетей ЦОД для кластеров машинного обучения требует воспроизведения не только протоколов, но и их взаимодействия.

Моделирование стека RoCEv2 — основе лежит модель сети без потерь. Симулятор должен корректно обрабатывать логику работы механизма PFC, генерацию и обработку кадров PAUSE. Неточное моделирование PFC может скрыть такие проблемы, как PFC-штормы или взаимные блокировки.

Моделирование методов контроля перегрузок. Для DCQCN это включает ECN-маркировку на коммутаторе, генерацию пакетов на получателе и алгоритм реакции на отправителе. Для алгоритмов, основанных на телеметрии (HPCC, FNCC), модель коммутатора должна поддерживать добавление метаданных в заголовки пакетов, а модель хоста — их извлечение и обработку.

Моделирование методов балансировки нагрузки. ЕСМР требует наличия хэш-функции и таблицы маршрутизации с несколькими равнозначными путями. Для Packet Spraying модель должна поддерживать распределение пакетов одного потока по разным путям, что неизбежно приведет к их переупорядочиванию на приемнике.

Для корректной оценки производительности трафик, генерируемый в модели, должен по характеру соответствовать трафику, наблюдаемому в реальных ML-кластерах (коллективные операции, трафик «многие-к-одному»).

4.4. Требования к процессу моделирования

Исходя из перечисленных особенностей, к программному обеспечению для моделирования предъявляется следующий ряд требований:

- высокая гранулярность — система должна поддерживать дискретно-событийное моделирование с точностью до отдельного пакета для воспроизведения динамики очередей и микро-выбросов трафика;
- наличие моделей протоколов, в симуляторе должны быть реализованы модели протоколов RoCEv2, DCQCN и механизма PFC, составляющие основу современных сетей ЦОД;
- возможность реализации собственных алгоритмов — программное обеспечение должно предоставлять интерфейсы для разработки и интеграции пользовательских алгоритмов балансировки нагрузки и контроля перегрузок;
- возможность сбора телеметрии — инструмент должен позволять собирать ключевые метрики производительности для последующего анализа.

На основе проведенного анализа, наиболее подходящим инструментом является симулятор NS-3 с расширениями для RDMA, так как он предоставляет весь необходимый функционал и отвечает всем требованиям, а также доказал свою применимость для подобных задач в работах [1, 2, 3].

ЗАКЛЮЧЕНИЕ

В ходе выполнения научно-исследовательской работы были рассмотрены различные существующие методы контроля перегрузок, балансировки нагрузки и моделирования сетей кластера машинного обучения, а также был проведен сравнительный анализ по различным критериям.

По результатам выявлена специфика трафика обучения нейронных сетей, которая характеризуется коллективными коммуникациями, создающими синхронные всплески нагрузки и обладающими низкой энтропией потоков.

Что касается методов контроля перегрузок, то сравнительный анализ показал, что широко используемый алгоритм DCQCN, основанный на ECN, обладает высокой инерционностью и требует сложной настройки параметров, что делает его недостаточно эффективным для динамичных ML-нагрузок. Современные методы, использующие внутрисетевую телеметрию (HPCC) или управление со стороны коммутатора (RoCC), обеспечивают лучшую производительность, но предъявляют высокие требования к аппаратному обеспечению.

Выявлено, что статическая балансировка (ECMP) неэффективна для тяжелых потоков ML-трафика из-за хэш-коллизий. Динамическая балансировка на уровне пакетов (Packet Spraying) позволяет достичь максимальной пропускной способности сети, но требует решения проблемы переупорядочивания пакетов (Packet Reordering).

Для дальнейшей практической разработки метода обоснован выбор дискретно-событийного симулятора NS-3. Симулятор обладает необходимыми моделями протоколов, позволяет реализовывать пользовательскую логику на уровне очередей коммутаторов и конечных хостов, а также обеспечивает достаточную точность.

Таким образом, все задачи научно-исследовательской работы выполнены в полном объеме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Nosrati E., Ghaderi M. Predictive Load Balancing for RDMA Traffic. — 2025. — 11 c.
2. Khan T. и др. Impact of RoCE Congestion Control Policies on Distributed Training of DNNs. — 2022. — 10 c.
3. Addanki V., Michel O., Schmid S. PowerTCP: Pushing the Performance Limits of Datacenter Networks. — 2021. — 19 c.
4. Xu J. и др. FNCC: Fast Notification Congestion Control in Data Center Networks. — 2024. — 12 c.
5. Menikkumbura D. и др. Congestion Control for Datacenter Networks: A Control-Theoretic Approach // IEEE Access. — 2023. — Т. 34. — С. 1682-1696.
6. Bonato T. и др. Uno: A One-Stop Solution for Inter- and Intra-Datacenter Congestion Control and Reliable Connectivity. — 2025. — 13 c.
7. Bonato T. и др. FASTFLOW: Flexible Adaptive Congestion Control for High-Performance Datacenters. — 2024. — 13 c.
8. Bernárdez G. и др. GraphCC: A Practical Graph Learning-based Approach to Congestion Control in Datacenters. — 2023. — 11 c.
9. Cheng K. и др. PET: Multi-agent Independent PPO-based Automatic ECN Tuning for High-Speed Data Center Networks. — 2024. — 13 c.
10. Zhao K. и др. Prediction-Guided Control in Data Center Networks. — 2026. — 16 c.
11. McClure S., Ratnasamy S., Shenker S. Load Balancing for AI Training Workloads. — 2025. — 16 c.
12. Xiao W. и др. OFC: An Original Congestion-Based Fine-grained Priority Flow Control // Proc. 21st Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). — 2024. — С. 28-36.
13. Li C. и др. m4: A Learned Flow-level Network Simulator. — 2025. — 15 c.
14. Wan H., Fang L., Li X. Grid Operational Benefit Analysis of Data Center Spatial Flexibility. — 2025. — 5 c.

15. IEEE 802.3x-1997. IEEE Standards for Local and Metropolitan Area Networks: Specification for 802.3 Full Duplex Operation. — New York: IEEE, 1997.
16. IEEE 802.1Qbb-2011. IEEE Standard for Local and metropolitan area networks—Virtual Bridged Local Area Networks—Amendment: Priority-based Flow Control. — New York: IEEE, 2011.

ПРИЛОЖЕНИЕ А

Презентация к научно-исследовательской работе состоит из 4-х слайдов.