

**Министерство образования и науки Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ”**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**АГРЕГАЦИЯ КАНАЛОВ СВЯЗИ ПОВЕРХ СЕТЕВОГО УРОВНЯ OSI-**  
**МОДЕЛИ С УЧЕТОМ ТРЕБОВАНИЙ QOS**

Автор Ванин Алексей Борисович \_\_\_\_\_  
(Фамилия, Имя, Отчество) (Подпись)

Направление подготовки (специальность) 09.04.01  
Информатика и вычислительная техника

Квалификация Магистр

Руководитель Соснин В.В., к.т.н. \_\_\_\_\_  
(Фамилия, И., О., ученое звание, степень) (Подпись)

**К защите допустить**

Зав. кафедрой \_\_\_\_\_  
(Фамилия, И., О., ученое звание, степень) (Подпись)

“ \_\_\_\_\_ ” 20 \_\_\_\_ г.

Санкт-Петербург, 2018 г.

# ОГЛАВЛЕНИЕ

Оглавление .....	5
Введение .....	6
Глава 1. Обзор литературы и постановка задачи .....	10
1.1. История развития механизмов агрегации каналов передачи данных .....	10
1.2. Агрегация на уровне протоколов транспортного уровня OSI-модели....	13
1.3. Агрегация каналов при передаче мультимедийного трафика .....	15
1.4. Критерий эффективности и постановка задачи исследования .....	18
1.5. Выводы по главе 1 .....	20
Глава 2. Исследование эффективности агрегации .....	21
2.1. Исследование агрегации трафика с использованием TCP .....	21
2.2. Упрощения и допущения .....	25
2.3. DCCP и балансировка трафика.....	26
2.4. Маркировка трафика .....	28
2.5. Выводы по главе 2.....	30
Глава 3. Анализ полученных результатов .....	31
3.1. Описание разработанного приложения.....	31
3.2. Проведение экспериментов и анализ результатов.....	33
3.3. Сценарии применения в реальных проектах .....	36
3.4. Выводы по главе 3 .....	37
Заключение .....	39
Список литературы .....	41

## ВВЕДЕНИЕ

**Актуальность области исследования.** С развитием телекоммуникационных и информационных технологий в современном мире продолжает расти доля передаваемого мультимедийного трафика [1]. Под этим понимается не только передача видео и аудио данных по требованию, осуществляемая, например, при просмотре фильмов или видеороликов в сети интернет.

Последние 5 лет стремительно растёт популярность потокового вещания (streaming), будь это трансляция в прямом эфире какого-то события или работа с удалёнными сервисами в режиме реального времени с последующей демонстрацией результата работы пользователю [2]. С развитием технологий, позволивших организовывать такие трансляции имея лишь телефон с выходом в интернет и приложением для потокового вещания, всё чаще возникают вопросы о способах повышения эффективности передачи таких данных. Одним из таких способов является использование механизмов агрегации каналов передачи данных.

**Объектом исследования** является совокупность каналов передачи данных в рамках одной вычислительной системы.

**Предметом** исследования является агрегация этих каналов при использовании соответствующих протоколов.

**Цель** исследования состоит в повышении эффективности передачи потоковых мультимедийных данных (звук, видео) при доступности на стороне отправителя нескольких каналов передачи.

Для достижения цели исследования были поставлены следующие **задачи**:

1. Провести обзор решений, связанных с механизмами агрегации каналов передачи данных на различных уровнях OSI-модели.
2. Оценить эффективность существующих протоколов агрегации при передаче мультимедийного трафика с учётом требований систем реального времени (потоковое вещание).

3. Исследовать причины, по которым существующие протоколы могут передавать потоковые мультимедийные данные неэффективно.
4. Разработать решение, позволяющее увеличить эффективность передачи, и провести анализ его эффективности.

**Теоретическая и методологическая основа исследования.** В качестве методов исследования используются методы математической статистики и теории планирования машинных экспериментов, теория массового обслуживания, имитационное моделирование. Средствами исследования являются: среда имитационного моделирования NS-3, модель MPTCP, реализация MPTCP, язык программирования C++.

**Актуальность темы исследования.** Современные вычислительные машины, объединённые в компьютерную сеть, зачастую обладают несколькими сетевыми интерфейсами: ноутбуки поставляются с Ethernet- и Wi-Fi адаптерами, мобильные телефоны обладают возможностью передаваться данные по Wi-Fi, 3G, LTE сетям. Одним из способов повысить качество передачи путём увеличения суммарной пропускной способности или увеличения надёжности передачи данных на таких устройствах является агрегация нескольких каналов передачи данных [16,17].

Принимая во внимание популярность приложений для потокового вещания на мобильных телефонах и в целом рост сервисов, работающих с потоковым вещанием, вполне естественно желание использовать данное решение для повышения качества трансляций. Однако специфика мультимедийного трафика накладывает определённые ограничения и существующие решения могут быть неоптимальными для этого класса задач. Исследованию этого вопроса и посвящена данная работа.

**Степень теоретической разработанности темы.** Теоретическую основу работы составляют научные труды, посвященные исследованию механизмов агрегации каналов передачи данных. Большой вклад в разработку этой темы были

внесены исследователями и учёными Ramin Khalili из европейского исследовательского центра Huawei, Martin Becke из Университета прикладных наук Гамбурга, профессором Olivier Bonaventure из Университетского колледжа Лондона.

**Информационная база исследования.** В основе исследования лежат статьи «Multipath TCP: Analysis, Design and Implementation» [4] и «Multipath Transport on the Internet» [5], описывающие особенности работы протоколов агрегации каналов передачи данных.

**Научная новизна исследования.**

1. Определены границы эффективного использования существующих протоколов агрегации каналов передачи данных на сетевом уровне OSI-модели при передаче мультимедийного трафика.
2. Разработан программный комплекс, позволяющий эффективно передавать потоковый трафик по объединённому каналу.
3. Сформированы рекомендации по построению эффективных систем с использованием агрегации каналов передачи данных для передачи потокового трафика.

**Практическая значимость исследования.** Разработанные рекомендации могут быть использованы при построении сетевой инфраструктуры с проксирующими серверами, обеспечивающими оперативный доступ к запрашиваемым ресурсам за счёт агрегации медленных каналов передачи данных у клиентов.

**Апробация результатов исследования.** Основные положения работы были доложены и обсуждены на VI Всероссийском конгрессе молодых учёных [6]. Работа была отмечена дипломом за лучший научно-исследовательский доклад в рамках III сессии научной школы «ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ». Кроме того, вопросы инкапсуляции мультимедийного трафика были изложены на 20-ой Международной конференции «DCCN-2017 Распределенные компьютерные и телекоммуникационные сети:

управление, вычисление, связь». Основные положения опубликованы в сборнике тезисов конференции [7].

**Объем и структура работы.** Диссертационная работа содержит 43 страницы машинописного текста, 18 рисунков, 7 таблиц и список литературы, включающий 19 источников. Структурно работа состоит из введения, трёх частей и заключения. Во введении обоснована актуальность и новизна, определена цель и задачи исследования. Первая часть содержит обзор механизмов агрегации, там же находится постановка задачи. Во второй части исследуется эффективность существующих протоколов агрегации и вырабатываются подходы для повышения эффективности передачи данных. Третья часть посвящена анализу полученных результатов, которые приведены в заключении.

# ГЛАВА 1. ОБЗОР ЛИТЕРАТУРЫ И ПОСТАНОВКА ЗАДАЧИ

## 1.1. История развития механизмов агрегации каналов передачи данных

Вопрос о способах повышения качества работы вычислительных сетей стоял и стоит перед разработчиками и исследователями с момента их развития и массового применения во всем мире. В то время как одни специалисты занимались усовершенствованием физических компонент, будь то линии передачи данных или обслуживающее сетевое оборудование, другие разрабатывали подходы на основе уже существующих решений. Идеи об использовании нескольких параллельных каналов передачи данных для увеличения суммарной пропускной способности, надёжности или для иных целей сами по себе не новы. К середине 90-х годов прошлого века большая часть коммутаторов поставлялась с возможностью объединения линий передач данных [3]. Однако первые попытки стандартизировать подходы и решения в этой области были произведены лишь в 1998 году, когда был представлен проект стандарта IEEE 802.3ad, описывающий то, что дальше будет названо Link Aggregation Control Protocol. Стандарт будет принят позднее в 2000 году [8].

Использование механизмов агрегации каналов передачи данных позволяет достичь:

- Увеличения суммарной пропускной способности и уменьшение задержек
- Увеличения надёжности передачи данных и отказоустойчивости сети
- Осуществления балансировки загрузки каналов

Говоря о пропускной способности, в простейшем случае, обладая двумя каналами передачи данных со сходными показателями задержек, максимальной пропускной способности, джиттера, мы можем отправлять по ним параллельно отдельные фрагменты или пакеты, как представлено на рисунке 1.

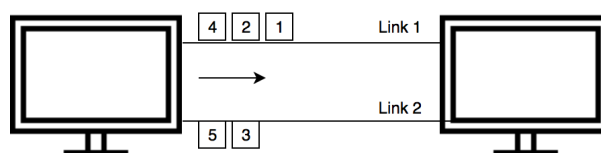


Рисунок 1. Способ передачи пакетов по параллельным каналам для увеличения пропускной способности

Для увеличения надёжности каналы передачи данных могут функционировать в режиме «активный-пассивный» и при появлении отказов при передаче данных по активному каналу, происходит переключение на один из пассивных каналов. Другой подход заключается в параллельной передаче одних и тех же данных по всем доступным каналам, снижая вероятность того, что определенный пакет потеряется или исказится во всех каналах. Эти подходы продемонстрированы на рисунке 2.

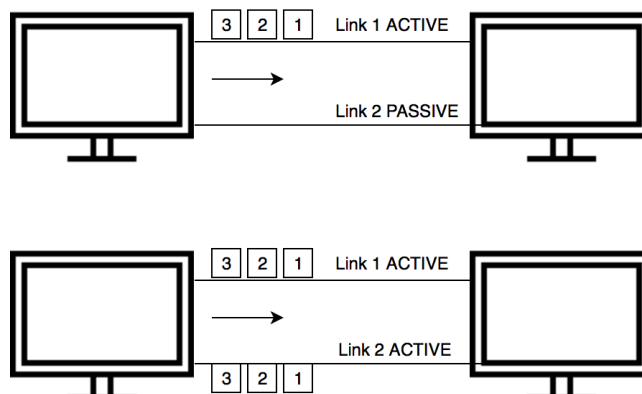


Рисунок 2. Способы передачи пакетов по параллельным каналам для увеличения надёжности

При конфигурации сети также может потребоваться передавать трафик разного типа по разным каналам из соображений безопасности, показателей загрузки или иным другим. Агрегация каналов может взять на себя бремя балансировки.

Для достижения указанных выше целей разработчики протоколов решают ряд технических проблем, связанных с распараллеливанием передаваемых данных. В первую очередь это проблема упорядоченности передаваемых данных.



Принимающая сторона должна воссоздать исходный поток, части которого могут приходить непоследовательно и с произвольной задержкой, особенно если каналы передачи данных не обладают смежными характеристиками. Другой проблемой является внесение избыточности. Дополнительная информация о параллельном потоке, например, порядковые номера пакетов, могут добавляться протоколами агрегации, тем самым уменьшая передаваемый объем полезных данных. Это может вызвать дополнительную фрагментацию, негативно влияющую на загрузку сетевого оборудования. Третьей важной проблемой является гетерогенность и изменчивость характеристик независимых параллельных каналов передачи данных [14]. В процессе функционирования и передачи данных показатели загрузки канала, задержек внутри и другие могут неоднократно измениться. Протоколы агрегации должны следить за этим и корректировать интенсивность передаваемого потока данных в каждый доступный канал.

Как было отмечено выше, уже в середине 90-х годов существовали разные проприетарные протоколы агрегации, которые поставлялись вместе с сетевым оборудованием. Несмотря на принятие общего стандарта, в мире по-прежнему существует огромное разнообразие механизмов агрегации каналов, как закрытых проприетарных решений, так и свободных с исходными текстами. Яркими представителями являются Link Aggregation Control Protocol, механизмы Port Truncing, NIC Teaming, LAG (Link Aggregation). В операционных системах семейства Linux агрегирующий режим работы с сетевыми устройствами называется Bonding, который реализован в виде драйвера в ядре операционной системы. Всё это многообразие протоколов зачастую является несовместимым друг с другом.

Все эти решения так или иначе подразумевают использование не на клиентских системах, а на передающих узлах, формирующих структуру сети. Это разумно, поскольку различное сетевое оборудование, будь то маршрутизаторы, коммутаторы или иное другое, обладает совокупностью портов, а значит и возможностью их параллельного использования для повышения эффективности работы всей сети. Однако современные тенденции таковы, что всё больше и

больше клиентских станций – стационарных компьютеров, ноутбуков, мобильных телефонов, имеют несколько независимых друг от друга каналов передачи данных. Это может быть как несколько сетевых карт в компьютере, так и возможность работы с Wi-Fi и 3G сетями одновременно в мобильном телефоне. Вектор разработки механизмов агрегации сместился в сторону высокоуровневых протоколов, использование которых предполагается только на клиентских системах.

### 1.2. Агрегация на уровне протоколов транспортного уровня OSI-модели

Основная идея, лежащая в основе высокоуровневых протоколов агрегации заключается в том, что соединение происходит не между двумя соседними узлами сети (рисунок 3.а) , а между клиентской и серверной компонентой, которые могут быть расположены удалённо и взаимодействие между которыми происходит поверх проложенной вычислительной сети между ними (рисунок 3.б). Это позволяет увеличивать эффективность передачи данных между приложениями, не изменяя структуры всей сети, поверх которой идёт обмен данными.

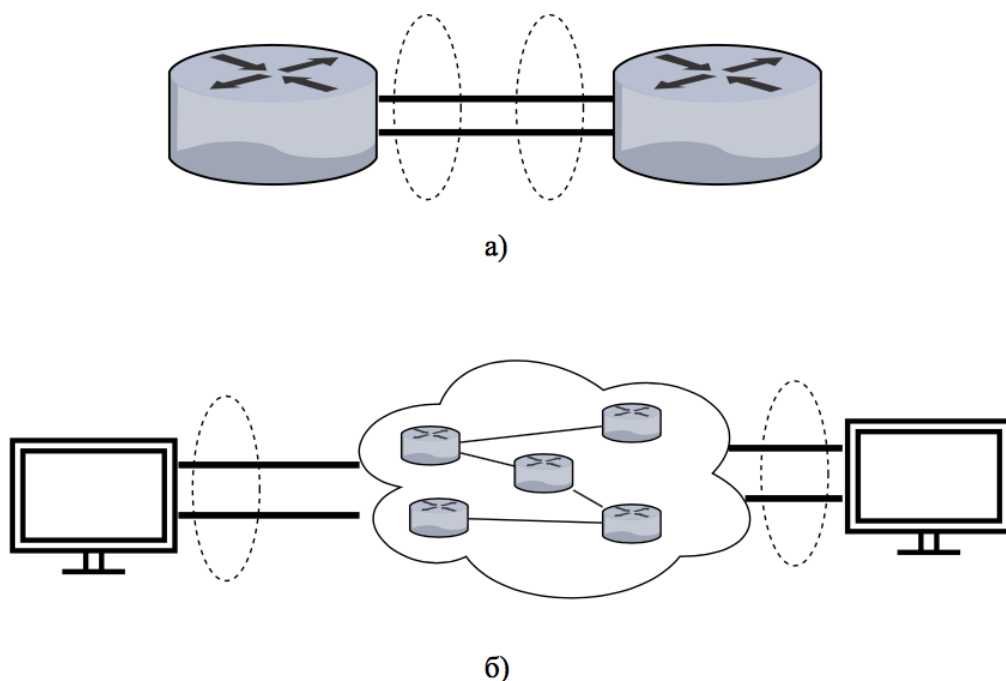


Рисунок 3. Отличие агрегации на канальном и транспортном уровнях OSI-модели.

Между клиентом и сервером предварительно устанавливается несколько соединений с помощью протоколов стека TCP/IP. После процедуры инициализации передаваемые данные определенным образом фрагментируются и инкапсулируются в дейтаграммы транспортного протокола и передаются по заранее установленным соединениям. Приёмная сторона получает данные, распаковывает их и передает дальше уже в упорядоченном виде. Существуют два дополнения к протоколам TCP и SCTP для их функционирования в режиме агрегации каналов: это Multipath TCP (MPTCP) и Concurrent Multipath Transfer SCTP (CMT-SCTP) соответственно.

MPTCP производит предварительную установку нескольких TCP-сессий между клиентской и серверной компонентой (TCP-Flow). При поступлении данных на MPTCP сокет, они инкапсулируются в TCP-дейтаграмму, которая распространяется по одному из выбранных TCP-Flow. Данная схема представлена на рисунке 4.

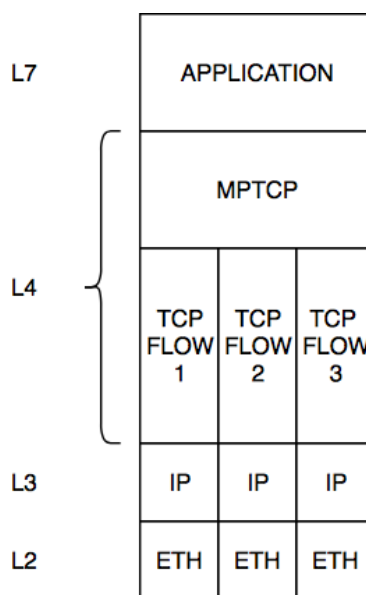


Рисунок 4. Соотношение уровней OSI-модели при передаче данных с MPTCP

Алгоритм функционирования CMT-SCTP в плане установления соединения и передачи данных почти идентичен MPTCP [9]. Первое отличие заключается в том, что на транспортном уровне используется протокол SCTP вместо

ТСР. Кроме этого, на этапе предварительного соединения CMT-SCTP устанавливает количество соединений равных минимальному количеству доступных портов на клиенте или сервере. MPTCP в свою очередь устанавливает сессию между каждой парой портов у сервера и клиента, создавая полносвязную топологию [10] (рисунок 5)

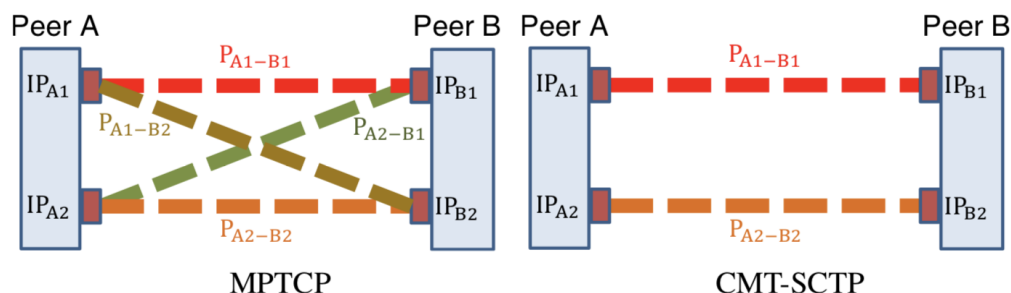


Рисунок 5. Топология соединений протоколов агрегации

### 1.3. Агрегация каналов при передаче мультимедийного трафика

Описанные выше протоколы поддерживают принципы fairness (честность), которые несколько ограничивают их максимально возможную пропускную способность. Кроме того, и тот и другой протокол использует гарантированную доставку данных, что влечёт за собой повторную отправку потерянных или искажённых пакетов. Для передачи мультимедийных данных, будь то голос или видео, использование таких протоколов не всегда позволяет достичь желаемого качества обслуживания.

В общем случае мультимедийные данные можно разделить на две условные категории:

- Мультимедийные данные по запросу (on demand)
- Поточковые мультимедийные данные (streaming)

Основные отличия заключаются в требованиях, которые выдвигаются к ним в процессе передачи. К мультимедийным данным по-запросу в первую очередь выдвигаются требования по минимизации потерь и искажений при передаче, что обеспечивает максимальное соответствие принимаемого видео и аудио ряда с исходным материалом. Допустима предварительная буферизация с

ожиданием. Примерами сервисов, оперирующих с мультимедийными данными по запросу, являются различные видео хранилища, такие как Youtube, Vimeo и другие, или онлайн кинотеатры, например Амедиатека, Netflix.

Говоря о потоковых мультимедийных данных, к ним предъявляются требования систем реального времени – а именно максимально оперативный отклик. Для увеличения оперативности потерянные кадры не переотправляются, а пользователь получает видео и аудио ряд с некоторыми искажениями. Примерами сервисов, работающих с потоковыми мультимедийными данными, являются: Twitch, Periscope.

В связи с тем, что существующие протоколы агрегации основаны на гарантированной передаче данных, что может повлиять на эффективность вещания потокового видео, было решено исследовать влияние потерь и задержек в каналах на работу данных протоколов. Тестовый стенд (рисунок 6) состоял из двух компьютеров с двумя сетевыми картами, соединенных через коммутатор двумя независимыми каналами связи, в которых варьировались величины задержек и потерь. Пропускная способность каналов 100 Мбит/с. Компьютеры работали под управлением ОС Ubuntu 14.10, процессор Intel Core i5-2467M с тактовой частотой 1.6 Гигагерц, одноядерный режим работы, объем оперативной памяти 1 Гигабайт. Для потокового вещания был подготовлен видеоряд длительностью 2 минуты 32 секунды, постоянный битрейт 1 Мбит/с. Подготовленное видео транслировалось при помощи утилиты ffmpeg с первого компьютера на второй, где выводилось пользователю на экран при помощи утилиты VLC. Вероятность потерь варьировалась в интервале от 0 до 0,1 с шагом 0,005. Задержка выставлялась в интервале от 0 до 300 миллисекунд.

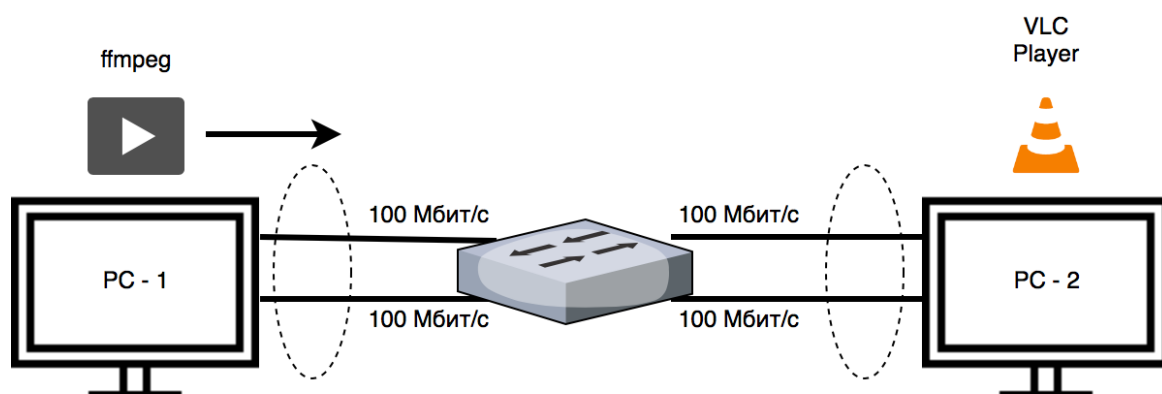


Рисунок 6. Схема тестового стенда

Проведенный эксперимент продемонстрировал, что при некоторых значениях потерь и задержек можно добиться полной остановки вещания: изображение и звук начинают приходить с задержками, дёргаться, пусть и не теряя в качестве исходного кадра [7]. Интенсивность отправки утилитой ffmpeg снижается и происходит полная остановка. Более подробно описание результатов представлено в главе 2.

Другой важный вывод – архитектурные решения в CMT-SCTP негативно влияют на функционирование данного протокола при сильно асимметричных каналах. Из-за того, что в MPTCP устанавливается полносвязная топология, то при наличии больших потерь в одном канале и отсутствии их в другом в нашем эксперименте все данные идут через канал без потерь, тем самым обеспечивая лучшее качество картинки для пользователя, по сравнению с CMT-SCTP. В таблице 1 представлена конечная задержка принимаемого видео с общей задержкой в канале 300 мс. Поэтому в дальнейшем мы будем рассматривать только MPTCP как протокол агрегации каналов.

Таблица 1. Сравнение работы протоколов MPTCP и CMT-SCTP

Протокол	Задержка видео на приемной стороне при вероятности потерь указанных через ; в двух симметричных каналах		
	0% ; 1%	0% ; 4,5%	0% ; 9%
MPTCP	0,301 с.	0,301 с.	0,301 с.
CMT-SCTP	1,83 ± 0,12 с.	10,24 ± 2,14 с.	∞

#### 1.4. Критерий эффективности и постановка задачи исследования

Оценка качества видео – нетривиальная задача, на результат которой могут оказать влияние параметры исходного видеоматериала, способы кодирования и человеческое восприятие. Хотя и существуют определенные рекомендации, позволяющие гарантировать некоторое качество обслуживания [19], в том числе и для потокового видео, всё равно конечный результат принято оценивать другой метрикой – качеством восприятия (Quality of Experience). Обычно эта оценка формируется статистическими методами путём опроса группы респондентов и выставления ими некоторой субъективной оценки прослушанному аудио или просмотренному видео ряду [11]. На рисунке 7 представлены некоторые рекомендованные значения задержек и потерь в канале для корректной работы разных классов трафика.

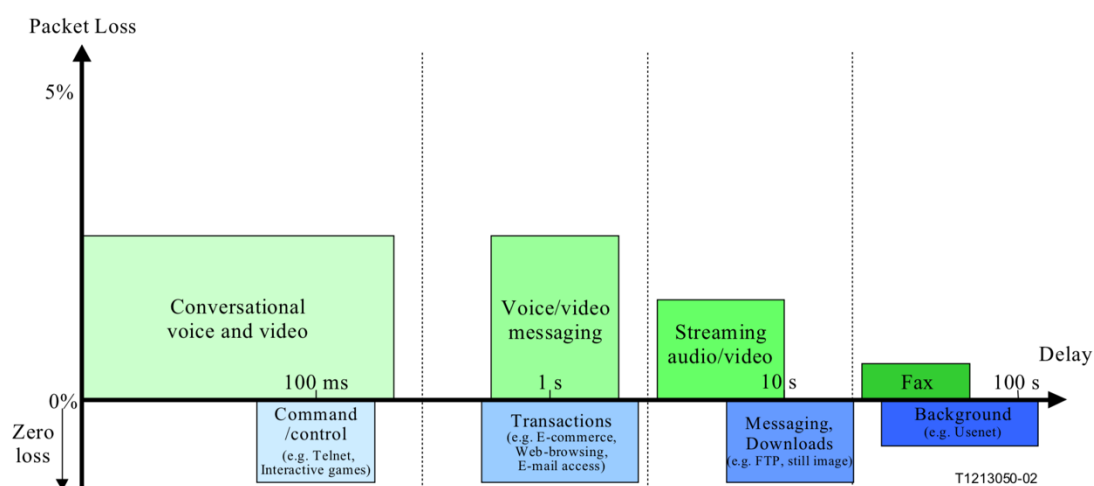


Рисунок 7. Характеристики канала передачи данных для разных типов трафика в рекомендации ITU-T Rec. G.1010

Попробуем оценить наш результат некоторым количественным способом, а именно количеством принятых плеером VLC блоков, которые он смог декодировать, обозначим величину *Decoded*. Декодированные блоки содержат видео и аудио фрагменты и демонстрируются пользователю. Соответственно, чем больше блоков будет продемонстрировано пользователю, тем полнее он получит представление о передаваемом потоке. Не будем забывать, что в потоковом видео предъявляются требования систем реального времени: все блоки,

представленные с задержкой выше, чем рекомендации QoS не будут учитываться. Нормируем эту величину поделив количество принятых декодированных блоков на количество блоков, сгенерированных отправителем, обозначим за *Generated*. Эту величину необходимо максимизировать и это будет являться критерием эффективности.

$$\frac{Decoded}{Generated} \rightarrow \max$$

Таким образом можно представить результаты, полученные в ходе исследования протокола МРТСП на симметричных каналах (рисунок 8). Как видно из графика, существует ряд условий, где данные передаются довольно неэффективно.



Рисунок 8. Эффективность передачи потокового видео при использовании МРТСП

Поставим цель – разработать систему, позволяющую передавать потоковые мультимедийные данные при тех же значениях потерь и задержек в симметричном и асимметричном агрегированном канале передачи данных с большей эффективностью, чем МРТСП.



### 1.5. Выводы по главе 1

1. Использование механизмов агрегации каналов передачи данных для увеличения суммарной пропускной способности, надёжности или балансировки является распространённой практикой уже с середины 90-х годов. Однако популярные на сегодня решения подразумевают объединение каналов на канальном уровне OSI-модели. Сейчас происходит рост популярности решений для агрегации каналов на пользовательских системах при помощи более высокоуровневых протоколов вроде MPTCP и CMT-SCTP.
2. Данные протоколы в своей основе используют надёжную передачу данных с подтверждениями и пересылками потерь, что в свою очередь может негативно сказаться на качестве обслуживания при передаче потокового мультимедийного трафика.
3. Протокол CMT-SCTP имеет архитектурные особенности, которые не позволяют передавать эффективно потоковый мультимедийный трафик при асимметричных характеристиках каналов передачи данных, что ограничивает область использования данного протокола.
4. Протокол MPTCP не позволяет эффективно передавать потоковый мультимедийный трафик при высоких процентах потерь в каналах (зависит от общей задержки в канале), а значит необходимо исследовать возможности повышения эффективности передачи такого трафика используя агрегацию каналов передачи.

## ГЛАВА 2. ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ АГРЕГАЦИИ

### 2.1. Исследование агрегации трафика с использованием ТСП

Рассмотрим детальнее ход эксперимента с агрегацией потокового мультимедийного трафика при помощи МРТСП. Исследование показало, что протокол адаптивно использует предоставленные каналы передачи данных благодаря установке полносвязной топологии между клиентской и серверной стороной. Это позволяет корректировать интенсивность потока, что означает при асимметричной загрузке каналов иметь минимальную пропускную способность равную пропускной способности самого ненагруженного канала.

$$\hat{C}_{min} = \max (C_1, C_2, \dots, C_n)$$

где  $\hat{C}_{min}$  – минимальная пропускная способность агрегированного канала,

$C_1, C_2, \dots, C_n$  – совокупность пропускных способностей каждого из каналов.

Рассмотрим детальнее случай, когда оба канала имеют симметричные показатели задержек и потерь. Задержки в канале варьировались в интервале от 0 до 300 мс, а вероятность потерь от 0 до 0,1. Важно понимать, что ТСП фактически перестаёт функционировать при проценте потерь в канале  $\sim 11\%$ , а значит мы подошли почти к технологическому ограничению протокола.

При проведении экспериментов была выявлена зависимость, что при достижении определенных значений задержек и потерь полностью останавливалась передача данных – для пользователя за вторым компьютером происходит замирание видеокадра на длительное время, после чего экран заполняют произвольные видео артефакты, не позволяющие оценить, что происходит на экране (рисунок 9).

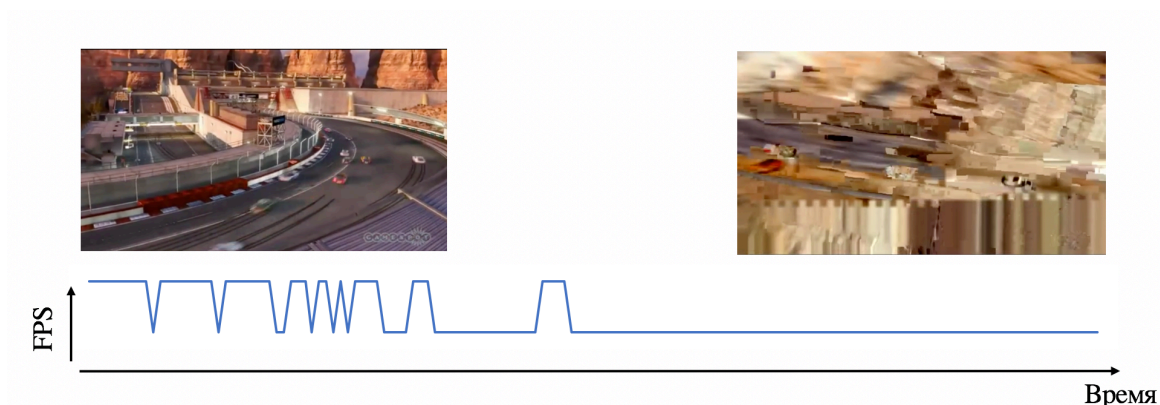


Рисунок 9. Паузы в транслировании мультимедийных данных при высоких значениях задержек и потерь

Такая остановка трансляции и последующие совершенно искажённые кадры могут быть объяснены переполнением и сбросом буферов в приложении, которое управляет вещанием. В буфере могут храниться все неотправленные пакеты по причине задержек на повторную отправку, которые происходят при потерях. Это очень важный момент, особенно для мобильных приложений, или приложений во встроенных системах, где могут существовать повышенные требования к расходу памяти. Переполнение буфера может привести к его сбросу в клиентском приложении, отчего все последующие пакеты будут нести информацию из будущих кадров, когда у принимающей стороны есть лишь части прошлых. Этим объясняется тот факт, что вещание после длительной паузы на принимающей стороне возобновляется, однако оно очень фрагментировано.

В ходе исследования были предприняты попытки определить те самые граничные значения, при которых начинается переполнение буферов на клиентской стороне. Результаты представлены на рисунке 10. В процессе эксперимента происходило варьирование вероятности потерь, а не величины задержки в каналах.

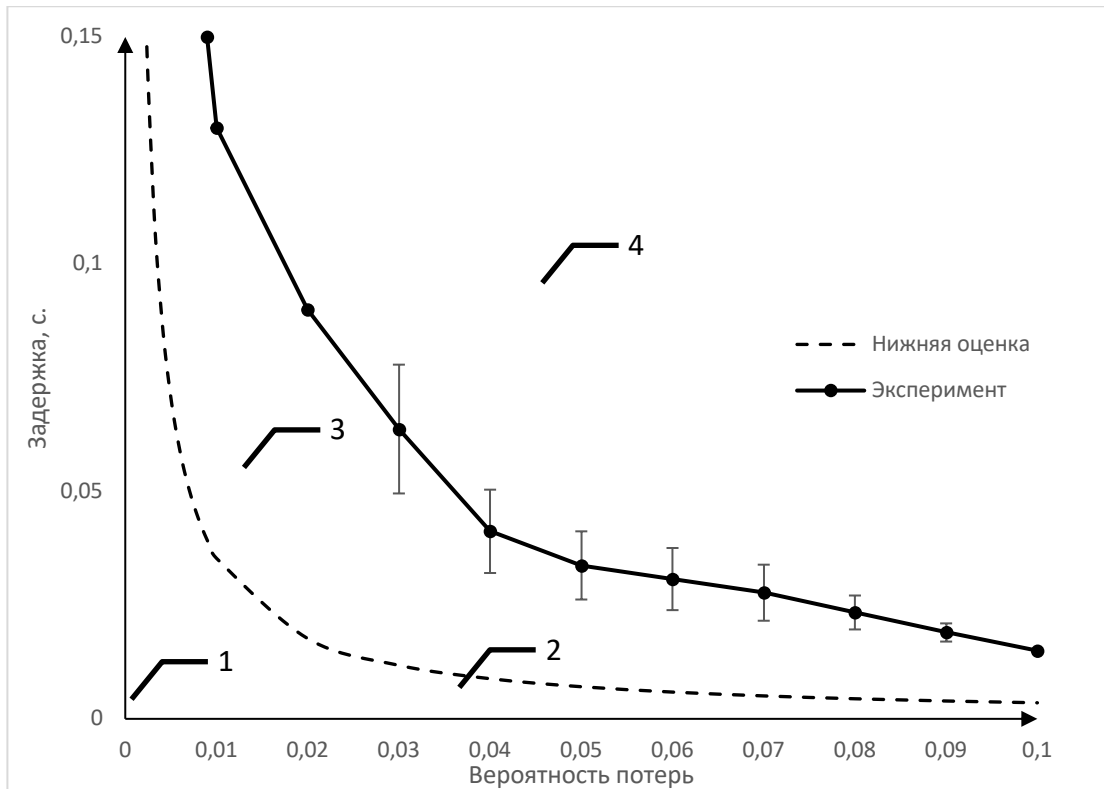


Рисунок 10. Оценка границы сброса буфера и прекращения вещания

Дополним полученные результаты некоторой простой аналитической оценкой. Обозначим за  $Buf$  – размер буфера в битах,

$BitR$  – постоянный битрейт видео в бит/с,

$D$  – задержка на передачу пакета в секундах,

$N$  – количество переданных пакетов,

$L$  – вероятность потери в симметричных каналах

Пусть видео имеет постоянный битрейт и передаётся в качестве детерминированного потока с отправкой пакетов через равные интервалы времени. Также примем упрощение, что потеря пакетов происходит не случайным образом (рисунок 11.а). Их количество определено заранее и все потери происходят последовательно друг за другом (рисунок 11.б).

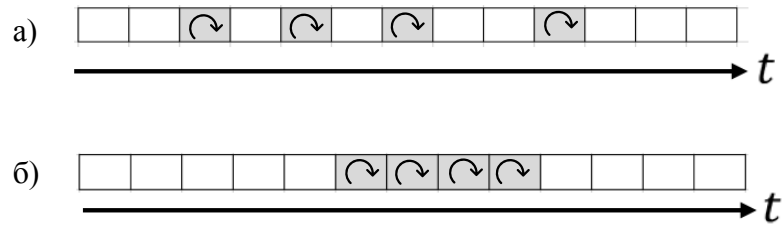


Рисунок 11. Модель потерь пакетов в канале связи

Исходя из этого можно предположить, что размер буфера должен вмещать в себя все повторно отправленные пакеты:

$$Buf \geq N \cdot L \cdot BitR \cdot D$$

Отсюда можно выразить и максимальную вероятность потери в канале, при которой даже с выбранным размером буфера не будет происходить его перезагрузка и сброс:

$$L \leq \frac{Buf}{BitR \cdot N \cdot D}$$

Таким образом мы можем изобразить на рисунке 10 нижнюю границу – некоторый упрощенный вариант, который хоть и не будет встречаться в жизни, однако он позволяет выделить несколько зон на графике, которые можно однозначно интерпретировать, что поможет корректно настроить своё приложение для работы с существующими протоколами агрегации каналов связи. В таблице 2 представлены показатели этих зон.

Таблица 2. Показатели приёмного видеопотока при потерях и задержках в рамках описанных зон

Номер зоны	Артефакты	Задержки
1	нет	нет
2	нет	незначительные
3	нет	значительные
4	да	значительные

Зона 1 соответствует минимальным значениям задержек и потерь в канале. Этот вариант считается идеальным и использование MPTCP или CMT-SCTP для

передачи даже потокового мультимедийного трафика здесь можно считать оптимальным.

Зона 2 соответствует возросшим задержкам при таких же минимальных потерях. В данной зоне по-прежнему применимы существующие протоколы агрегации, поскольку единственное требование к мультимедийному трафику со стороны пользователя в данном случае – это оперативность предоставления данных. Задержка на формирование кадров на клиентской и серверной стороне достаточно мало относительно возросших задержек в канале. Зона 1 и 2 находятся ниже определенной нами аналитической границы. Эксперимент показывает, что все значения задержек и потерь, лежащие ниже этой границы, все ещё позволяют эффективно применять существующие протоколы агрегации каналов передачи данных даже для потокового мультимедийного трафика.

Характеристики канала лежащие внутри зоны 3 уже вызывают заметные пользователю задержки в трансляции. На компьютере 2 на тестовом стенде были заметны паузы во время вещания, которые происходили с разной интенсивностью и с разной длительностью, из-за чего падает эффективность передачи согласно выбранному критерию эффективности. Однако никаких артефактов в видеопотоке не наблюдается.

В зоне 4 происходит сброс буферов и полное искажение видео и аудио потока. Эффективность передачи данных именно в этой зоне крайне мала из-за постоянных повторных отправок потерянных пакетов. Повысить эффективность передачи именно в зоне 4 и частично в зоне 3 можно с помощью выбора транспортного протокола, основанного на ненадёжной передаче данных.

## **2.2. Упрощения и допущения**

Исследования по агрегированию каналов передачи данных с использованием протоколов ненадёжной передачи данных транспортного уровня будем производить в среде имитационного моделирования NS-3. Данная среда позволит нам использовать как заранее подготовленные компоненты, построенные для моделирования реальных сетевых подсистем, так и обладает возможностью

дополнить модель собственными компонентами, разработанными под эту среду моделирования.

Вариантов построения модели для имитирования тех или иных сценариев агрегирования каналов передачи данных может быть множество. Между клиентским и серверным приложением по разным каналам могут существовать независимые пути, пути с общими отрезками, полностью общие пути [18]. К тому же современные компьютерные сети часто используют динамическое построение маршрутов и автоматическую конфигурацию.

В данной работе мы будем рассматривать внутреннее строение сетей между клиентом и сервером в виде черного ящика (рисунок 12), где формируются потери и задержки по причинам, которые могут случаться из-за наличия общих высоконагруженных отрезков маршрута, изменчивости сети или иных других причинах.

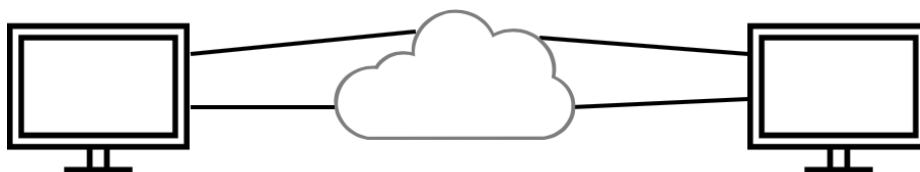


Рисунок 12. Соккрытие внутренней структуры сети между клиентом и сервером в процессе исследования

В самой среде моделирования NS-3 будет использовано расширение Direct Code Execution (DCE), которое позволяет использовать в моделировании не только подготовленные модели, но и реальные имплементации протоколов. Это вынужденная мера, связанная из-за отсутствия моделей некоторых протоколов, в частности DCCP

### **2.3. DCCP и балансировка трафика**

Причина, лежащая в основе низкой эффективности передачи потоковых мультимедийных данных, заключается в контроле доставки данных. И протокол SCTP и TCP производят контроль того, что данные доставлены в нужном порядке без потерь и искажений, в случае возникновения которых происходит

переотправка. А значит наш подход в увеличении эффективности заключается в использовании на транспортном уровне протокола без повторной отправки. В литературе уже выдвигались предложения использовать для этого протокол DCCP, а не UDP [5,15]. Основные сравнительные характеристики представлены в таблице 3.

Таблица 3. Сравнительная характеристика протоколов транспортного уровня

Показатель	Протоколы			
	TCP	SCTP	UDP	DCCP
Надёжность доставки	Да	Да	Нет	Нет
Установление соединений	Да	Да	Нет	Да
Управление потоком	Да	Да	Нет	Да

Хоть мы и отказываемся от надёжной доставки данных, мы всё ещё хотим использовать полезные механизмы управления потоком, которые присутствуют в протоколах TCP и SCTP. Управление потоком – важная часть работы протоколов агрегации, поскольку среда передачи имеет свойство изменять свои характеристики и для эффективной передачи следует оперативно адаптироваться под новое состояние системы. Существуют разные механизмы оценки характеристик канала между клиентским и серверным приложением. Одним из популярных решений является технология IP SLA, осуществляющая мониторинг канала и отправляющая через заданные интервалы времени небольшие наборы заранее подготовленных данных. По результатам оценки можно производить корректировку интенсивности передаваемого потока данных.

Далеко не всегда данный подход позволяет оперативно управлять потоком, поэтому механизмы управления потоком зачастую встраиваются и в сами протоколы передачи данных. В протоколе DCCP реализован CCID 2 (TCP-like Congestion Control — контроль насыщения в стиле TCP) являющийся



механизмом контроля насыщения с аддитивным увеличением и мультипликативным уменьшением интенсивности потока, в соответствии с моделью, используемой для протокола TCP, включая окно насыщения, замедленный старт [12]. Механизм CCID 2 обеспечивает максимальное использование полосы в течение продолжительного периода, совместимое со сквозным контролем насыщения, но уменьшает вдвое окно насыщения в ответ на каждый факт возникновения перегрузки. Это ведет к внезапным изменениям скорости, характерным для TCP [13].

На рисунке 13 представлено как CCID 2 позволяет управлять интенсивностью передачи в среде с переменными показателями доступной пропускной способности.

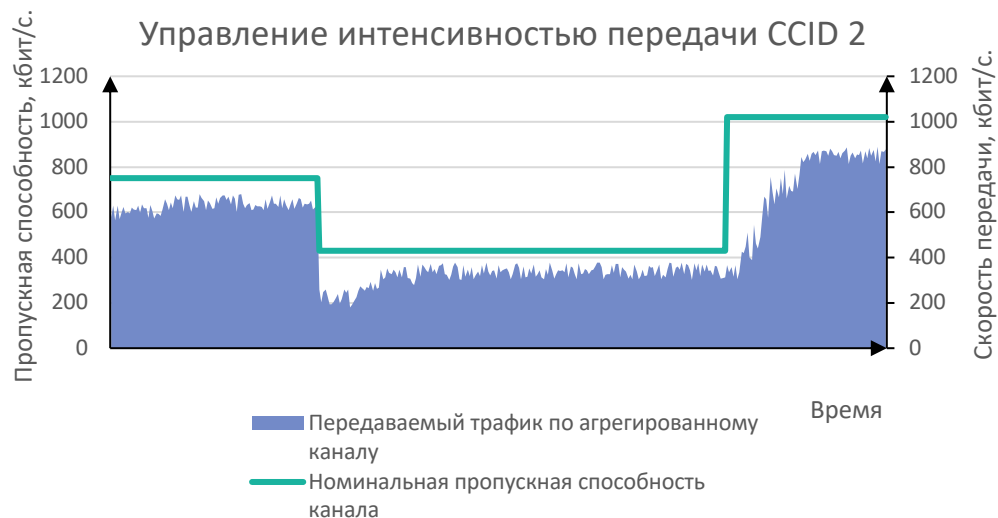


Рисунок 13. Управление интенсивностью передачи в канале при помощи CCID 2

## 2.4. Маркировка трафика

Другой важной деталью является очередность прихода трафика и передача его вышестоящим запрашивающим приложениям. DCCP обладает возможностями контроля порядка принимаемых дейтаграмм, однако, когда речь идёт о нескольких каналах передачи данных, то здесь необходимо использовать иные подходы. Одним из применяемых решений является промежуточное хранение пакетов в буфере.

На стороне отправителя каждая дейтаграмма маркируется по возможности уникальным идентификационным порядковым номером. Отправляясь по параллельным каналам, некоторые из этих дейтаграмм попадают в принимающее приложение на серверной стороне. Схема алгоритма приёма сообщений представлена на рисунке 14.

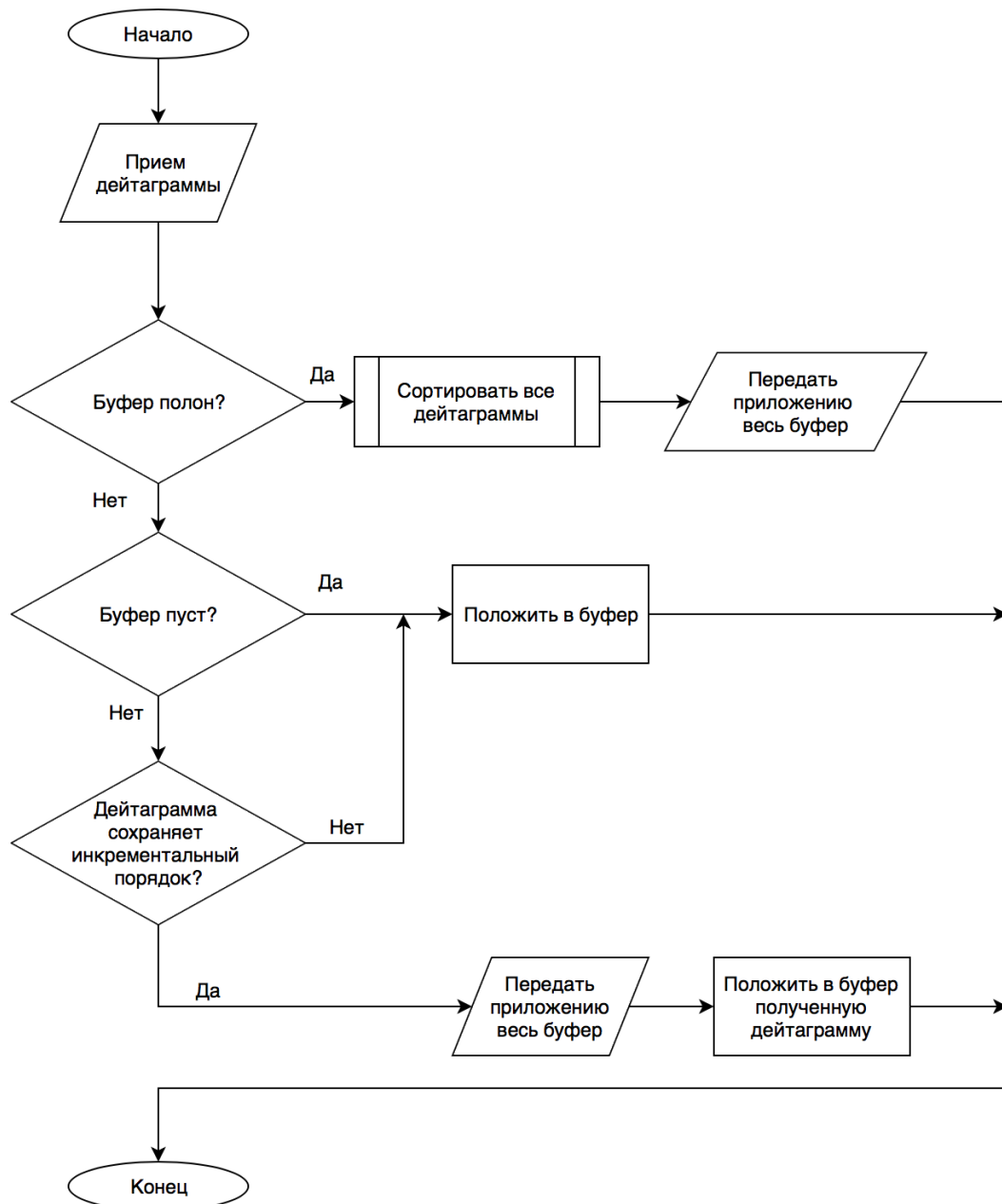


Рисунок 14. Схема алгоритма предварительной буферизации принимаемого потока

## **2.5. Выводы по главе 2**

1. Существующие протоколы агрегации каналов передачи данных при определенных значениях потерь и задержек в канале вызывают полную остановку вещания, что связано с переполнением буферов на отправляемой стороне. Нижнюю допустимую границу, при которой использование МРТСП не вызывает полной остановки трансляции, можно оценить аналитически.
2. Для повышения эффективности передачи потокового трафика необходимо использовать протоколы без повторной отправки потерянных или искаженных пакетов. В качестве такого протокола был выбран DCCP.
3. Протоколы агрегации каналов передачи данных должны активно использовать механизмы управления потоком для повышения суммарной пропускной способности. Использование средств интервального тестирования состояния канала может быть не оптимальным.

## ГЛАВА 3. АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

### 3.1. Описание разработанного приложения

Для проведения натурного эксперимента было решено построить реальную модель приложения, которое работает по принципам протокола агрегации каналов передачи данных на транспортном уровне. Используя данное приложение можно оценить каким образом изменится критерий эффективности при использовании решений, предложенных в главе 2 по сравнению с существующими протоколами агрегации.

Разработанное ПО под названием SUFix обладает клиент-серверной архитектурой, где клиентская часть инициализирует соединение при помощи DCCP-сокетов с серверной стороной. После установки соединения пользователю будет доступен интерфейс для отправки своих данных от клиента к серверу (рисунок 16). Направленные туда потоковые мультимедийные данные будут инкапсулироваться в дейтаграммы протокола DCCP с добавленной маркировкой для последовательного сбора внутри клиентской стороны (рисунок 15).

ETH	IP	DCCP	SUFix Mark	Payload
-----	----	------	---------------	---------

Рисунок 15. Структура кадра при передаче SUFix-дейтаграммы

Серверная сторона в свою очередь обладает буфером, который имеет параметризуемый размер. В данном буфере происходит ожидание пакетов в случае, если нарушается последовательность маркеров. Как только данный буфер переполняется, все накопленные пакеты отправляются дальше к пользовательскому приложению (в нашем случае это VLC плеер) с возрастающей очередностью.

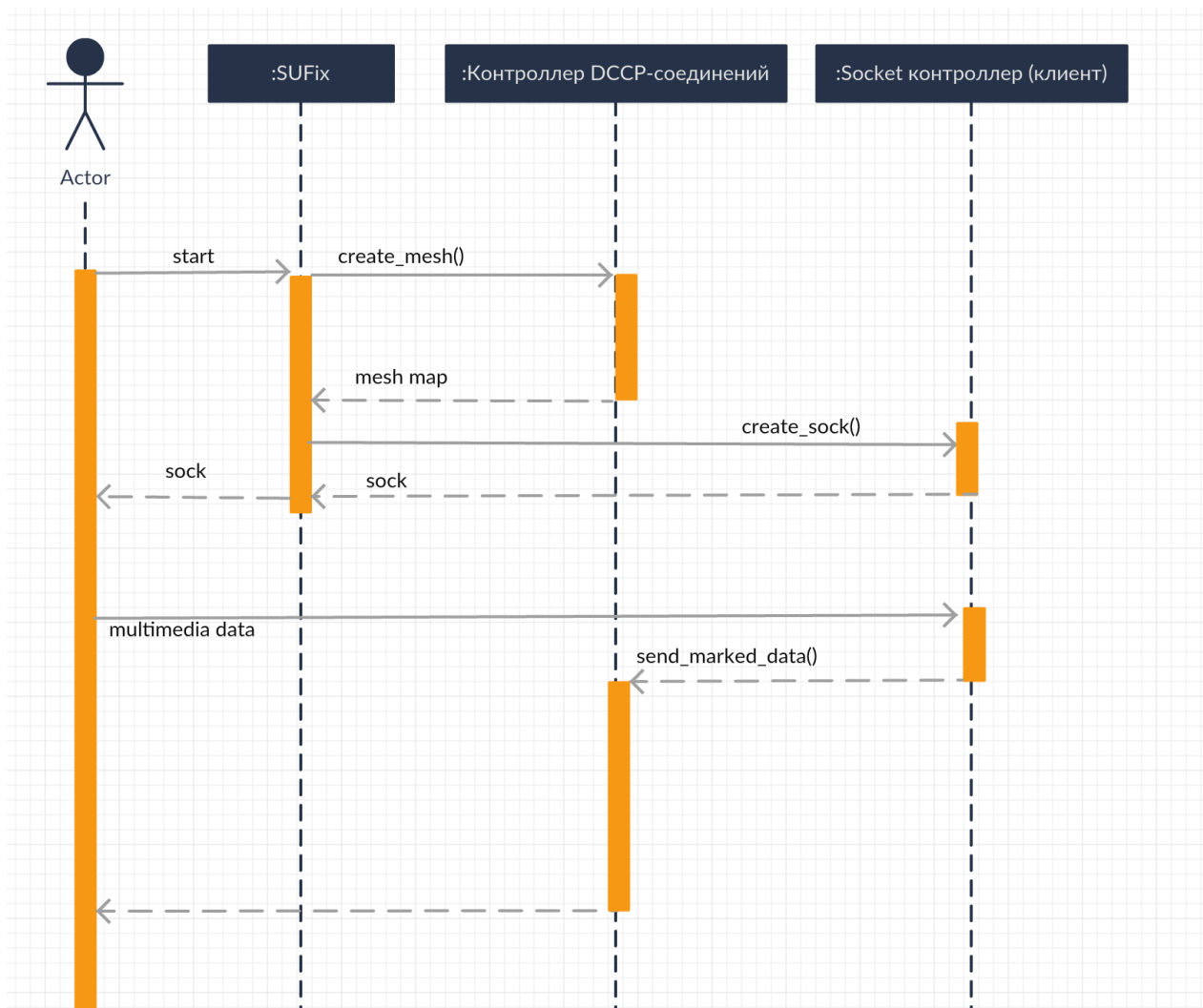


Рисунок 16. UML-диаграмма последовательности исполнения клиентской стороны

При разработке были использованы различные упрощения модели, связанные как с условиями тестового окружения, так и с технологичностью разработки программного обеспечения.

- Изменчивость среды передачи и балансировка передаваемого трафика.

Один из критериев выбора протокола DCCP заключался в способности контроля перегрузкой и интенсивностью передаваемого трафика. Однако в условиях нашего тестового стенда состояние каналов не меняется со временем, из чего следует одна из причин использования данного упрощения. Тем самым происходит имитация работы

идеального диспетчера, управляющего интенсивностью передачи данных в каждом отдельном канале.

- Имплементация на уровне пользовательского приложения, а не модуля ядра операционной системы.

Для доступа к метрикам протокола DCCP приложению необходимо работать в режиме ядра и функционировать как её модуль. Для ускорения разработки было решено отказаться от данного подхода.

- Одинаковое количество доступных портов при передаче данных на клиентской и серверной стороне.

Также для ускорения разработки и упрощения технических решений внутри программного обеспечения было решено пренебречь вариантом, когда создаётся комплексная топология соединений между клиентом и сервером, связанная с разным количеством доступных портов для соединения. Таким образом соединения будут устанавливаться парами между доступными портами при условии сохранения симметричных характеристик каналов.

### **3.2. Проведение экспериментов и анализ результатов**

Эксперимент проводился на основе того же тестового стенда, описанного в главе 1 (рисунок 6). Проведём также варьирование задержки в каналах передачи данных в интервале от 0 до 200 мс, а вероятность потерь в интервале от 0 до 0,1. Таким образом мы охватим 4 зоны результатов, описанных в главе 2 и рассмотрим, как повлиял протокол DCCP на эффективность передачи потокового мультимедийного трафика.

Результаты для зоны 1 представлены в таблице 4, где отображена разница в эффективности передачи MPTCP и разработанного решения: из нормированного количества принятых декодированных блоков при использовании SUFix вычитается нормированное количество принятых декодированных блоков при использовании MPTCP. Как видно из результатов, в данных условиях

использование протокола ненадёжной передачи данных незначительно повлияло на эффективность передачи данных в худшую сторону (значения в таблице являются отрицательными). Без потерь суммарное время задержки формируется в большей степени задержкой распространения сигнала в канале, нежели задержками форматирования и инкапсулирования пакетов. Поэтому в данной зоне следует придерживаться рекомендаций QoS для установления удовлетворительной задержки для пользователя.

Таблица 4. Разница между нормированными значениями декодированных блоков в зоне 1

Вероятность потерь	Разница в эффективности передачи между SUFix и MPTCP при задержках в канале				
	0 с.	0,01 с.	0,025 с.	0,05 с.	0,075 с.
0	0	0	0	0	0
0,01	-0,08±0,01	-0,07±0,02	-0,06±0,02	-0,04±0,02	-0,04±0,03

Результаты из зоны 2 представлены в таблице 5. Здесь заметно, что MPTCP эффективнее осуществляет передачу потокового мультимедийного трафика. Это связано с тем, что время повторной передачи настолько мало и незначительно, что протокол с надёжной передачей данных почти не теряет оперативности при отправке новых кадров. Не наблюдая значительной разницы в оперативности поставки видеопотока, пользователи всё же видят артефакты, которые возникают при потерянных пакетах в разработанном решении.

Результаты из зоны 3 представлены в таблице 6. В данной зоне уже начинаются заметные проблемы MPTCP, поскольку переотправки начинают занимать длительное время и многие корректно отправленные кадры приходят со значительным опозданием. Использование DCCP позволяет повысить эффективность передачи потоковых мультимедийных данных.

Таблица 5. Разница между нормированными значениями декодированных блоков в зоне 2

Вероятность потерь	Разница в эффективности передачи между SUFiX и MPTCP при задержках в канале	
	0 с.	0,01 с.
0	0	0
0,01	-0,08±0,01	-0,07±0,02
0,02	-0,08±0,01	-0,06±0,01
0,03	-0,09±0,03	-0,08±0,02
0,04	-0,11±0,02	-0,10±0,03
0,05	-0,16±0,03	-0,13±0,02
0,06	-0,20±0,03	-0,18±0,04
0,07	-0,21±0,02	-0,19±0,02
0,08	-0,24±0,03	-0,20±0,03
0,09	-0,27±0,04	-0,21±0,01

Таблица 6. Разница между нормированными значениями декодированных блоков в зоне 3

Вероятность потерь	Разница в эффективности передачи между SUFiX и MPTCP при задержках в канале		
	0,025 с.	0,05 с.	0,075 с.
0,01	-0,06±0,02	-0,04±0,02	-0,04±0,01
0,02	-0,05±0,02	0,05±0,02	0,21±0,03
0,03	-0,05±0,03	0,02±0,01	0,24±0,05
0,04	-0,02±0,01	0,11±0,03	0,31±0,04
0,05	0,01±0,02	0,21±0,06	0,37±0,03
0,06	0,05±0,03	0,25±0,03	0,44±0,02
0,07	0,06±0,01	0,29±0,03	0,48±0,08
0,08	0,07±0,02	0,32±0,08	0,50±0,02
0,09	0,07±0,02	0,34±0,07	0,53±0,04

Результаты из зоны 4 представлены в таблице 7. В данной зоне фактически бессмысленно использовать протоколы с надёжной передачей данных, поскольку происходит сброс буферов и остановка вещания. На общем рисунке 17, демонстрируется совокупная картина из всех проведенных экспериментов с



разработанным приложением. Зеленый цвет сигнализирует об повышении эффективности передачи с использованием продемонстрированного подхода, красный – о понижении эффективности по сравнению с МРТСП, серый цвет – результаты сопоставимы.

Таблица 7. Разница между нормированными значениями декодированных блоков в зоне 4

Вероятность потерь	Разница в эффективности передачи между SUFix и МРТСП при задержках в канале		
	0,1 с.	0,0125 с.	0,015 с.
0,01	0,16±0,03	0,23±0,06	0,36±0,02
0,02	0,35±0,04	0,43±0,02	0,52±0,06
0,03	0,39±0,02	0,51±0,02	0,53±0,09
0,04	0,42±0,04	0,54±0,08	0,56±0,02
0,05	0,45±0,05	0,53±0,02	0,61±0,04
0,06	0,50±0,03	0,53±0,04	0,60±0,08
0,07	0,52±0,06	0,54±0,04	0,58±0,03
0,08	0,52±0,08	0,52±0,12	0,57±0,06
0,09	0,53±0,08	0,54±0,07	0,54±0,04

Задержки в канале, с.	Разница в эффективности передачи между SUFix и МРТСП при вероятности потерь в канале:									
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
0,15	0,00	0,36	0,52	0,53	0,56	0,61	0,60	0,58	0,57	0,54
0,125	0,00	0,23	0,43	0,51	0,54	0,53	0,53	0,54	0,52	0,54
0,1	0,00	0,16	0,35	0,39	0,42	0,45	0,51	0,52	0,52	0,53
0,075	0,00	-0,04	0,21	0,24	0,31	0,37	0,44	0,48	0,50	0,53
0,05	0,00	-0,04	0,05	0,24	0,11	0,21	0,25	0,29	0,32	0,34
0,025	0,00	-0,06	-0,05	-0,05	-0,02	0,01	0,05	0,06	0,07	0,07
0,01	0,00	-0,07	-0,06	-0,08	-0,10	-0,13	-0,18	-0,19	-0,20	-0,21
0	0,00	-0,08	-0,08	-0,09	-0,11	-0,16	-0,20	-0,21	-0,24	-0,27

Рисунок 17. Сборные результаты нормированных показателей эффективности.

### 3.3. Сценарии применения в реальных проектах

Для корректной работы с высокоуровневыми протоколами агрегации каналов передачи данных необходима поддержка как со стороны

пользовательского приложения, так и со стороны сервера, с которым пользователь взаимодействует. Несмотря на то, что такой подход значительно проще и дешевле в отличие от изменения самой структуры сети с повсеместным внедрением агрегации на канальном уровне, всё же он требует определенных настроек для отдельных владельцев сервисов, будь то указанные выше онлайн кинотеатры или площадки для организации прямых трансляций.

Иной подход заключается в организации проксирующих сервисов на основе прокси-серверов, обладающих каналами с высокой пропускной способностью. Пользователи данных сервисов используя свои медленные каналы передачи данных, с помощью агрегирующих протоколов запрашивать или передавать данные к прокси-серверу (рисунок 18).

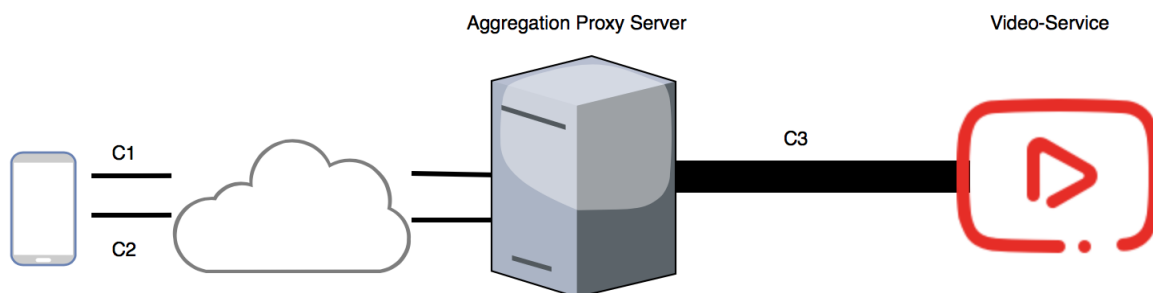


Рисунок 18. Организация агрегирующего прокси-сервера для ускорения доступа к ресурсам в глобальной сети

На текущий момент уже существуют сервисы, предоставляющие услуги по ускорению доступа к ресурсам сети интернет при помощи агрегирования каналов передачи данных и прокси-серверов. Однако в большинстве своём эти решения являются коммерческими и исследований на тему эффективности их работы не производилось.

### 3.4. Выводы по главе 3

1. Для тестирования эффективности подходов и решений, описанных в главе 2, была произведена разработка клиент-серверного приложения, которое позволяет провести эксперимент на развёрнутом ранее тестовом стенде. В процессе разработки некоторые описанные решения были упрощены, например

динамическое управление интенсивностью передаваемого потока, ввиду трудоёмкости разработки подобных решений.

2. По итогам проведенных экспериментов выявлено, что в областях 3 и 4, описанных в главе 2, разработанное решение позволяет эффективнее передавать потоковые мультимедийные данные. Однако в зонах 1 и 2 наблюдается снижение эффективности передачи.
3. Внедрение протоколов агрегации каналов передачи данных можно осуществлять в виде организации проксирующих сервисов, которые будут обладать каналами с широкой пропускной способностью до запрашиваемых пользователем сервисов.

## ЗАКЛЮЧЕНИЕ

В результате проделанной работы были выполнены следующие задачи:

1. Рассмотрены существующие решения в области агрегации каналов передачи данных для увеличения качества обслуживания с помощью увеличения надёжности, суммарной пропускной способности или другими способами.
2. При оценке эффективности был построен тестовый стенд, на котором производилось измерение эффективности передачи такого типа трафика с использованием протоколов агрегации каналов передачи данных MPTCP и CMT-SCTP.
3. Были определены границы задержек и потерь в каналах передачи данных, при которых использование существующих протоколов не вызывает полной остановки передачи трафика и сформированы предложения по увеличению эффективности передачи потокового мультимедийного трафика за счёт организации передачи на основе протокола без повторной передачи.
4. Разработано приложение, позволяющее эффективнее по сравнению с MPTCP передавать потоковый мультимедийный трафик при высоких значениях потерь и задержек в агрегируемых каналах связи.

На основе полученных результатов можно сделать следующие выводы:

1. Существующие протоколы агрегации каналов в своем большинстве являются протоколами канального уровня, однако современные тенденции приводят к постепенному внедрению высокоуровневых протоколов.
2. В основе низкой эффективности передачи потокового мультимедийного трафика находятся архитектурные ограничения транспортного протокола, который производит повторную отправку потерянных или искажённых пакетов. Использование протокола без повторной передачи позволяет значительно повысить эффективность даже там, где существующие протоколы агрегации просто неработоспособны.

3. Для построения некоторого оптимального решения для агрегации каналов передачи данных необходимо вести разработку в области гибридных решений, которые позволят при разных характеристиках канала передавать определенный тип трафика либо без контроля его корректной доставки, либо с ним.

Дальнейшее развитие проведенного исследования возможно по следующим направлениям:

1. Исследование эффективности передачи потокового мультимедийного трафика в гетерогенных каналах с изменяющимися во времени показателями задержек и потерь.
2. Разработка полноценного протокола агрегации каналов передачи данных, функционирующего в виде модуля ядра операционной системы и входящий в её сетевой стек.

## СПИСОК ЛИТЕРАТУРЫ

1. Cisco Visual Networking Index: Forecast and Methodology, 2016-2021– [Электронный ресурс]. – URL: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html> (дата обращения 16 апреля 2018 г.).
2. Gandolfi, Enrico, To watch or to play, it is in the game: The game culture on Twitch.tv among performers, plays and audiences // Journal of Gaming & Virtual Worlds – 2016. – Т. 8 – № 1. – С. 63–82.
3. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы. 5-е издание. – СПб.: Питер, 2017 – 992 с.
4. Peng Q., Walid A., Hwang J., Low S. H. Multipath TCP: Analysis, Design, and Implementation // IEEE/ACM Transactions on Networking – 2016. – Т. 24 – № 1. – С. 596–609.
5. Siikavirta S., Moktan G., Jaakkola A. Multipath Transport on the Internet // Proceedings of 6th Seminar of Finnish-Russian University Cooperation in Telecommunications – 2009. – С. 122–129.
6. Ванин А. Влияние потерь и задержек на эффективность протокола многопутевой маршрутизации MPTCP // Сборник тезисов VI всероссийского конгресса молодых учёных – СПб.: Университет ИТМО, 2017.
7. Vanin A., Sosnin V. Streaming video transmission via OpenVPN tunnels // Proceedings of the 20th International Science Conference Distributed Computer and Communication Networks – 2017. – С. 291–293.
8. IEEE P802.3ad Link Aggregation Task Force – [Электронный ресурс]. – URL: <http://www.ieee802.org/3/ad/> (дата обращения 28 ноября 2017 г.).
9. Ford A., Raiciu C. RFC 6824 TCP Extensions for Multipath Operation with Multiple Addresses – [Электронный ресурс]. – URL: <https://tools.ietf.org/html/rfc6824> (дата обращения 5 февраля 2017 г.).

10. Becke M., Adhari H., Rathgeb E., Fu Fa, Xiong Yang, Xing Zhou Comparison of Multipath TCP and CMT-SCTP based on Intercontinental Measurements // Proceedings of IEEE Global Communications Conference (GLOBECOM) – 2013 – С. 1360–1366.
11. Деарт В., Кожухов И. Исследование параметров качества обслуживания (QoS) определяющих качество восприятия (QoE) потокового видео при передаче через интернет // Т-Comm – Телекоммуникация и транспорт – 2013. – Т. 8 – С. 44–47.
12. Floyd S., Kohler E. RFC 4341 Profile for Datagram Congestion Control Protocol. Congestion Control ID 2: TCP-like Congestion Control – [Электронный ресурс]. – URL: <https://tools.ietf.org/html/rfc4341> (дата обращения 24 января 2018 г.).
13. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е издание. – СПб.: Питер, 2016 – 960 с.
14. Попов С.Г., Тучков А.С., Курочкин Л.М., Шарагин М.П. Алгоритм распределения потоков данных между беспроводными глобальными и локальными сетями связи // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – 2017. – Т. 10. – № 3. С. 7–17. DOI: 10.18721/JCSTCS.10301.
15. Wheeb A. Performance Evaluation of UDP, DCCP, SCTP and TFRC for Different Traffic Flow in Wired Networks // International Journal of Electrical and Computer Engineering (IJECE) – 2017. – Т. 7. – № 6. – С. 3552–3557. – DOI: 10.17586/2226-1494-2015-15-6-976-983.
16. Vondrous O., Macejko P., Kocur Z. Multipath TCP in LTE Networks // Information and Communication Technologies and Services – 2014. – Т. 4 – С. 294–300.
17. Shahrudin A., Raaid A., Wisam A. Simulated performance of TCP, SCTP, DCCP and UDP protocols over 4G network // Proceedings of 8th International Conference on Advances in Information Technology – 2014.

- 18.Дмитриев А. Методы балансировки UDP-дейтаграмм между сотовыми сетями // Научные работы участников конкурса «Молодые ученые Университета ИТМО» 2015 года. – СПб.: Университет ИТМО, 2016. С. 82 – 85.
- 19.ITU-T recommendation Y.1541. – Geneva, 2012 [Электронный ресурс]. – URL: <https://www.itu.int/rec/T-REC-Y.1541-201112-I/en> (дата обращения 6 мая 2017 г.).