

# Разработка и исследование алгоритма централизованного управления видеопотоками MPEG-DASH в сетях Wi-Fi\*

А.Н. Красилов, М.В. Любогощев, Е.М. Хоров

ИППИ РАН

E-mails: {krasilov, liubogoshchev, khorov}@iitp.ru

**Аннотация** Одним из ключевых протоколов для организации сервисов видео по запросу (англ. Video on Demand) является протокол адаптивной передачи видеопотоков MPEG-DASH. Благодаря данному протоколу пользователи могут адаптивно выбирать качество загружаемого видеопотока. При принятии решения большинство существующих алгоритмов используют исключительно информацию, доступную на стороне пользователя (например, оценку пропускной способности соединения с сервером, объем данных в буфере и др.). В ряде работ показано, что такой подход может приводить к значительному ухудшению качества восприятия видеоизображения пользователями (англ. Quality of User Experience, QoE), особенно, в случае коротких видеопотоков. Для решения данной проблемы группой MPEG был разработан дополнительный протокол SAND, который позволяет пользователю и серверу обмениваться служебной информацией с промежуточными сетевыми узлами (например, точкой доступа сети Wi-Fi). В данной работе предложен алгоритм централизованного выбора битрейта видеопотоков и управления канальными ресурсами на точке доступа Wi-Fi, использующий новые функциональные возможности протокола SAND. С помощью имитационного моделирования показано, что предложенный алгоритм позволяет значительно улучшить QoE, особенно в случае передачи коротких видеопотоков.

## 1 Введение

На сегодняшний день видеопотоки составляют до 50% от всего мобильного трафика. Ожидается, что доля видеопотоков будет только увеличиваться и достигнет 75% в 2021 году [1]. Статистика таких популярных сервисов, как Youtube и Facebook показывает, что, во-первых, продолжительность значительной части видеороликов не превышает двух минут [2], а, во-вторых, для более длинных видеороликов доля пользователей, продолжающих просмотр, значительно сокращается после первых двух минут просмотра [3].

---

\* Исследование выполнено в ИППИ РАН за счет гранта Российского научного фонда (проект № 16-19-10687)

Кроме того, в последние годы наблюдается значительное увеличение объемов видеорекламы. При этом средняя продолжительность рекламного ролика составляет 60 секунд [4]. В связи с этим, важной задачей как для владельцев видеосервисов, так и для операторов, предоставляющих услуги связи, является обеспечение высокого качества восприятия видеоизображения пользователем (англ. Quality of user Experience, QoE), начиная с первых секунд видеопотока.

На сегодняшний день большинство пользователей просматривают видеоролики с устройств, подключенных к сети Интернет при помощи беспроводных технологий связи, таких как Wi-Fi или LTE. Для того чтобы адаптировать передачу видеопотоков к возможным флуктуациям качества соединения клиента с сервером, был разработан стандарт MPEG-DASH (Dynamic Adaptive Streaming over HTTP) [5]. С помощью данного стандарта клиенты могут адаптивно выбирать качество загружаемого видеопотока в зависимости от качества соединения. В целом, передача видеопотока по протоколу MPEG-DASH осуществляется следующим образом. На сервере видеофайлы делятся на фрагменты небольшой длительности, называемые сегментами. Каждый сегмент закодирован в нескольких уровнях качества. Клиент запрашивает сегменты по-очереди и для каждого сегмента выбирает определенное качество, основываясь на алгоритме, называемом алгоритмом выбора битрейта, который не специфицирован в стандарте. В нашей предыдущей работе [6] показано, что даже наиболее совершенные алгоритмы адаптации, использующие только локальную информацию, имеют ряд недостатков, которые значительно снижают QoE, особенно, в случае коротких видеопотоков.

Для улучшения качества обслуживания видеопотоков в 2016 году группа MPEG опубликовала дополнение к протоколу MPEG-DASH — протокол Server and Network assisted DASH (SAND) [7]. С помощью данного протокола клиент и сервер могут обмениваться служебной информацией с промежуточными сетевыми узлами (например, точкой доступа сети Wi-Fi). С помощью служебных сообщений протокола SAND клиенты могут сообщать промежуточным узлам информацию о собственном статусе (например, о текущем уровне буфера), а промежуточные узлы, в свою очередь, могут информировать клиентов об ожидаемой в ближайшее время пропускной способности соединения, рекомендованном клиентам качестве видеоизображения и т.д. Таким образом, с помощью протокола SAND становится возможной организация централизованного выбора качества для нескольких видеопотоков, а также разработка алгоритмов управления ресурсами сети. Однако стандарт [7] лишь описывает форматы сообщений и не предлагает конкретного алгоритма, когда и как использовать эти сообщения.

В данной работе предложен новый алгоритм централизованного управления качеством видеопотоков и распределения канальных ресурсов, который может быть реализован на точке доступа сети Wi-Fi. С помощью имитационного моделирования показано, что предложенный алгоритм поз-

воляет значительно повысить QoE, особенно, в случае передачи коротких видеопотоков.

Дальнейшее изложение работы построено следующим образом. В Разделе 2 рассмотрены существующие алгоритмы выбора битрейта, использующие локальную информацию, доступную на клиенте. В Разделе 3 дана математическая постановка задачи централизованного выбора битрейта и распределения канальных ресурсов, а также предложено решение данной задачи. В Разделе 4 описана реализация предложенного решения в сетях Wi-Fi. Результаты сравнения алгоритма централизованного управления с существующими алгоритмами представлены в Разделе 5. Наконец, в Разделе 6 резюмируются основные результаты работы.

## 2 Обзор существующих работ

С момента публикации стандарта [5] исследователями было предложено множество алгоритмов управления передачей видеопотоков MPEG-DASH. Один из ключевых вопросов, который возникает при разработке любого алгоритма, заключается в том, какие объективные метрики QoE измерять для настройки параметров алгоритмов и оценки их эффективности.

Самым важным фактором, влияющим на QoE, является качество просматриваемого видеоизображения (определяемое разрешением, частотой кадров, средним числом бит на один пиксель изображения и т.д.). Для его оценки предложено множество метрик (см., например, обзор [8]). Однако во многих работах используется средний битрейт, вычисляемый, как размер сегмента в битах, деленный на продолжительность его воспроизведения в секундах. В целом, чем выше битрейт, тем выше разрешение и число бит на пиксель, т.е. качество видеоизображения. Другим важным фактором является плавность воспроизведения видеоизображения. Она включает в себя такие метрики, как задержка начала воспроизведения первого кадра и доля пауз в общей продолжительности воспроизведения видеоклипа.

Задача алгоритмов выбора битрейта заключается в нахождении баланса между различными метриками, влияющими на QoE, т.к. данные метрики являются зависимыми друг от друга. Например, увеличение битрейта загружаемых сегментов требует большей пропускной способности сети, а следовательно увеличивает время загрузки сегмента и вероятность опустошения буфера. В данной работе рассмотрены два алгоритма, работающие локально на клиенте и использующие различную информацию при выборе битрейта следующего сегмента.

Первый алгоритм, называемый Instant, предложен в работе [9]. Данный алгоритм измеряет пропускную способность соединения с сервером в некотором временном окне и выбирает максимально возможный битрейт, не превышающий измеренной пропускной способности. Для первого сегмента алгоритм выбирает минимальный битрейт. Такой выбор позволяет минимизировать задержку начала воспроизведения.

Второй алгоритм, рассматриваемый в данной работе, называется Buffer Occupancy based Lyapunov Algorithm (BOLA). Он был предложен в работе [10] и в дальнейшем реализован в видеоплеере DASH-JS [11]. Данный алгоритм максимизирует функцию полезности, зависящую от среднего битрейта и ожидаемой доли пауз воспроизведения. В работе [10] показано, что для видеопотоков бесконечной длительности алгоритму достаточно только информации о текущем уровне буфера на клиенте. Однако для видеопотоков конечной длительности выбор битрейта осуществляется с учетом пропускной способности соединения. Отметим, что BOLA, аналогично алгоритму Instant, выбирает минимальный битрейт для первого сегмента.

Как показано в нашей предыдущей работе [6], использование алгоритмов выбора битрейта, использующих только информацию, доступную на клиенте, приводит к следующим недостаткам. Во-первых, средний битрейт просматриваемого пользователем видеоизображения оказывается значительно меньше доступной пользователю пропускной способности. Во-вторых, средний битрейт существенно снижается с уменьшением длительности видеофрагмента. В-третьих, использование более агрессивной политики выбора битрейта приводит к повышению вероятности опустошения буфера и высокой вероятности возникновения пауз воспроизведения. Данные проблемы вызваны недостатком информации о состоянии канала на клиенте. В данной работе, используя возможности протокола SAND, мы разрабатываем алгоритм централизованного управления битрейтом видеопотоков и распределения канальных ресурсов, который позволяет решить указанные выше проблемы и повысить QoE при обслуживании видеопотоков.

### 3 Алгоритм централизованного управления видеопотоками

#### 3.1 Постановка задачи

Рассмотрим сеть Wi-Fi, состоящую из одной точки доступа и  $N$  станций, загружающих видеопотоки MPEG-DASH. С помощью протокола SAND станции информируют точку доступа о текущем уровне буфера  $Q_i$  в секундах. Также, мы предполагаем, что точке доступа известно множество битрейтов  $\mathcal{B}_i = \{b_i^{(j)}\}_{j=1}^{M_i}$ , доступное для видеопотока, передаваемого станции  $i$ . Продолжительность сегментов у всех видеопотоков одинакова и равна  $\tau$ . Точка доступа измеряет пропускную способность  $C_i$  беспроводного канала со станцией  $i$ , как число бит, которое точка доступа может передать станции за единицу времени. Если станции  $i$  выделена доля канальных ресурсов  $\eta_i$ , то средняя пропускная способность соединения с данной станцией будет равна  $C_i\eta_i$ .

Задача централизованного выбора битрейтов видеопотоков и назначения канальных ресурсов может быть сформулирована следующим образом. Каждой станции требуется назначить битрейт  $b_i \in \mathcal{B}_i$  и выделить долю канальных ресурсов  $\eta_i$  с целью максимизации функции полезности:

$U(b_1, b_2, \dots, b_N)$ . При этом необходимо выполнить ограничения на суммарную долю канальных ресурсов  $\eta$ , выделенную для обслуживания всех видеопотоков ( $\sum \eta_i \leq \eta$ ). В данной работе рассматривается следующая функция полезности  $U(b_1, b_2, \dots, b_N) = \sum_{i=1}^N \log b_i$ . Данная функция обеспечивает баланс между средним качеством видеопотоков, загружаемых каждой станцией, и равномерностью распределения канальных ресурсов между станциями [12].

Определим связь доли канальных ресурсов  $\eta_i$ , выделенных для передач станции  $i$ , и динамику объема буфера  $Q_i$  станции  $i$ . Пусть битрейт текущего сегмента, назначенный станции  $i$  равен  $b_i$ . Тогда, если доля канальных ресурсов, выделенных станции  $i$  равна  $\eta_i = \frac{b_i}{C_i} F(Q_i)$ , то сегмент будет загружен за время  $\frac{b_i \tau}{\eta_i C_i} = \frac{\tau}{F(Q_i)}$ . Буфер в конце загрузки рассматриваемого сегмента будет равен  $Q_i = \max(Q_i - \tau/F(Q_i), 0) + \tau$ . Здесь мы ввели функцию  $F(Q_i)$ , называемую функцией запаса и определяющую скорость изменения буфера на станции  $i$ . Например, чтобы обеспечить задержку начала воспроизведения первого сегмента не больше чем  $I_0$  секунд, требуется установить значение  $F(0) = \frac{\tau}{I_0}$ . Отметим, что выполнение ограничения на задержку воспроизведения первого сегмента возможно только в случае, если  $I_0 \geq \frac{\eta C_i}{\min b_i}$ . Иными словами, требование может быть выполнено только при условии, что клиенту назначен минимально возможный битрейт и выделены все доступные канальные ресурсы.

Для того, чтобы избежать прерываний воспроизведения, на каждой станции требуется поддерживать некоторый объем буфера  $Q_{opt}$ . В случае, когда  $Q_i < Q_{opt}$ , требуется увеличивать его размер, т.е. установить  $F(Q_i) > 1$ . Заметим, что чем больше значения  $F(Q_i)$ , тем быстрее можно увеличить размер буфера. Однако большее значение  $F(Q_i)$  увеличивает расход канальных ресурсов. При  $Q_i \geq Q_{opt}$  имеет смысл сократить расход канальных ресурсов, установив  $F(Q_i) = 1$ . Следовательно, выбор функции запаса позволяет управлять размером буфера. В данной работе используется следующая кусочно-линейная функция запаса:

$$F(Q_i) = \begin{cases} \frac{\tau}{I_0}, & Q_i = 0 \\ A - \frac{A-1}{Q_{opt}-\tau}(Q_i - \tau), & 0 < Q_i \leq Q_{opt} \\ 1, & Q_i > Q_{opt}, \end{cases} \quad (1)$$

где  $I_0$ ,  $A$ ,  $Q_{opt}$  — параметры, позволяющие управлять размером буфера на различных станциях.

Используя данное определение функции запаса, сформулируем задачу выбора битрейтов следующим образом. Обозначим  $x_{ij} = \{0, 1\}$  — индикатор, отражающий выбор битрейта  $j$  для пользователя  $i$ . Тогда требуется найти такой набор  $\{x_{ij}\}$ , что

$$\begin{cases} \sum_{i=1}^N \sum_{j=1}^{M_i} x_{ij} \log b_i^{(j)} \rightarrow \max \\ \sum_{i=1}^N \sum_{j=1}^{M_i} x_{ij} \frac{b_i^{(j)}}{C_i} F(Q_i) \leq \eta, \\ \sum_{j=1}^{M_i} x_{ij} = 1, \forall i. \end{cases} \quad (2)$$

### 3.2 Алгоритм

Задача (2) — это задача о рюкзаке с мультивыбором, которая является NP-полной. Оптимальное распределение битрейтов возможно найти только полным перебором всех значений  $\{x_{ij}\}$ . Однако такое решение имеет слишком высокую вычислительную сложность. В связи с этим, в данном разделе предложен жадный алгоритм, который, как показало имитационное моделирование, дает результат близкий к оптимальному.

Для построения алгоритма используется следующий подход. Будем предполагать, что битрейт, назначенный станции  $i$ , может принимать непрерывный набор значений. Обозначим его  $\hat{r}_i$ . В таком случае задача (2) имеет следующий вид:

$$\begin{cases} \sum_{i=1}^N \log \hat{r}_i \rightarrow \max, \\ \sum_{i=1}^N \frac{\hat{r}_i}{C_i} F(Q_i) \leq \eta. \end{cases} \quad (3)$$

Нетрудно показать, что точным решением задачи (3) является вектор битрейтов:

$$\hat{r}_i = \frac{\eta}{N} \frac{C_i}{F(Q_i)} \quad (4)$$

Далее необходимо привести полученный набор битрейтов к величинам из множеств  $\mathcal{B}_i$ . Для этого мы сначала находим максимальный битрейт  $b_{ij} \in \mathcal{B}_j$  меньший  $\hat{r}_i$ . В результате перехода к дискретным битрейтам освобождается часть канальных ресурсов. Поэтому далее мы последовательно увеличиваем битрейт, назначенный станциям, пока не будут исчерпаны все доступные канальные ресурсы. Для этого используется жадный подход. Пользователи располагаются в порядке возрастания метрики  $m_i = \frac{F(Q_i)}{C_i} (b_{i(j+1)} - \hat{r}_i)$ , определяющей требуемую долю канальных ресурсов для повышения битрейта с непрерывного значения битрейта  $\hat{r}_i$  до следующего допустимого битрейта  $b_{i(j+1)}$ . Затем алгоритм повышает битрейт пользователям в соответствии с метрикой, пока не будут исчерпаны доступные канальные ресурсы, либо не останется станций, для которых можно повысить битрейт.

## 4 Реализация решения на точке доступа Wi-Fi

Для реализации алгоритма, описанного в Разделе 3, на точке доступа Wi-Fi требуется, во-первых, собирать статистику, необходимую для вычисления входных параметров алгоритма (а именно,  $Q_i$ ,  $\mathcal{B}_j$ ,  $C_i$ ,  $\eta_i$ ), а, во-вторых, обеспечивать пропускную способность соединения со станциями не меньше назначенной алгоритмом:  $R_i^{min} = b_i F(Q_i)$ .

### 4.1 Сбор входных параметров алгоритма

Начиная передачу нового видеопотока, станция сообщает точке доступа доступный ей набор битрейтов  $\mathcal{B}_i$ . Далее, в процессе загрузки, с помощью

протокола SAND станция сообщает текущий уровень буфера  $Q_i$  при каждом запросе нового сегмента. Получив сообщение с размером буфера, точка доступа запускает алгоритм, описанный в Разделе 3.2. В соответствии с результатом, точка доступа сообщает станции  $i$  назначенный ей битрейт  $b_i$  и устанавливает требуемую пропускную способность  $R_i^{min}$  на время загрузки следующего сегмента. При этом точка доступа не меняет битрейты и требуемые пропускные способности другим станциям, пока те не закончат загружать свои текущие сегменты.

В данной работе предполагается, что все станции и точка доступа используют одинаковые параметры доступа к каналу. На основе предыдущих передач точка доступа измеряет доступную ей долю канальных ресурсов  $\eta$ . Среднюю пропускную способность беспроводного канала со станцией  $i$  точка доступа измеряет с помощью экспоненциально сглаженного скользящего среднего:  $C_i(t_2) = (1 - \gamma)C_i(t_1) + \gamma \frac{L_i}{\Delta t}$ , где  $L_i$  — объем данных переданных станции  $i$  за интервал времени  $\Delta t = t_2 - t_1$ ,  $t_1$  и  $t_2$  — моменты измерений, а  $\gamma$  — параметр сглаживания.

## 4.2 Планировщик

Стандарт Wi-Fi [13] не регламентирует, какой алгоритм управления очередью пакетов должен использоваться на станциях и точке доступа. Единственное ограничение заключается в том, что передача пакетов одного и того же потока должна выполняться в порядке, соответствующем порядку прихода пакетов на точку доступа. В данной работе мы считаем, что точка доступа хранит пакеты каждого потока в отдельной очереди. Обслуживание очередей осуществляется с помощью алгоритма, называемого планировщиком PF (англ. Proportional Fair).

Для каждого клиента точка доступа на границах интервалов доступа к каналу должна обновлять среднюю скорость передачи данных следующим образом:  $\bar{R}_i(t) = \exp(\frac{\Delta t}{T})\bar{R}_i(t - \Delta t) + (1 - \exp(\frac{\Delta t}{T}))R_i I_i$ , где  $R_i$  — скорость передачи данных, соответствующая выбранной сигнально-кодовой конструкции,  $I_i$  — индикатор того, что передача велась пользователю  $i$ . Используя данную величину, точка доступа выбирает очередь для обслуживания, исходя из следующей метрики:

$$i^* = \arg \max_i \frac{R_i}{\bar{R}_i(t)} \quad (5)$$

Планировщик PF предоставляет передачам каждой станции равную долю временных ресурсов канала, что обеспечивает баланс между суммарной пропускной способностью сети и равномерностью распределения канальных ресурсов. С другой стороны, для работы алгоритма, описанного в Разделе 3, необходимо поддерживать пропускные способности соединений со станциями не меньшими назначенных алгоритмом  $R_i^{min}$ . Для решения данной задачи мы используем подход, предложенный в работе [14]. Для этого вводится функция  $T_i$ , определяющая, сколько бит требуется передать станции, чтобы выполнить заданные ограничения:  $T_i(t + \Delta t) = \max\{0, T_i(t) + R_i^{min} \Delta t - L_i\}$ ,

Таблица 1: Параметры эксперимента

Параметр	Значение
Протокол физ. уровня	IEEE 802.11n
Канал	40 МГц @ 5.18 ГГц
Модель затухания сигнала	$46.7 + 33 \log_{10}(r)$ дБ, $r$ — расстояние в м
Мощность передатчика	16.02 дБм
Агрегация пакетов	A-MSDU (до 7935 байт)
TCP congestion control	NewReno
Maximum Segment Size (MSS)	1460 байт
TCP initial window	10 MSS

где  $L_i$  — объем данных, переданных станции за интервал  $\Delta t$ . В работе [14] показано, что, чтобы выполнить заданные ограничения на пропускные способности соединений с клиентами, точке доступа требуется выбирать обслуживаемый поток в соответствии со следующим правилом:

$$i^* = \arg \max_i (e^{\alpha_i T_i} \frac{R_i}{R_i}), \quad (6)$$

где  $\alpha_i$  — коэффициент, определяющий, насколько сильно повышается приоритет той станции, для которой не выполнялись заданные ограничения. В нашей работе, чтобы повысить приоритет станций с малым объемом буфера, мы задаем  $\alpha_i = \alpha / (R_i^{\min} \max(\epsilon, Q_i(t)))$ , где  $\alpha$  — общий для всех станций коэффициент,  $\epsilon$  — малый коэффициент, позволяющий избежать деления на ноль.

## 5 Численные результаты

### 5.1 Постановка экспериментов

Для оценки производительности разработанного алгоритма используется среда имитационного моделирования NS-3 [15]. Рассматривается следующий сценарий: станции случайным образом размещены в круге радиусом 50 м, в центре которого расположена точка доступа. Точка доступа соединена с видеосервером с помощью проводного соединения с пропускной способностью 1 Гбит/с и круговой задержкой передачи пакета 10 мс. Другие параметры экспериментов приведены в Таблице 1.

Станции используют протокол MPEG-DASH для просмотра видеоклипов. Поведение пользователей моделируется следующим образом. Пользователь выбирает видеоклип и начинает загружать его с сервера. Спустя  $T_w$  секунд (величина  $T_w$  имеет экспоненциальное распределение со средним  $\bar{T}_w$ , минимальным и максимальным значением  $T_w^{\min}$  и  $T_w^{\max}$  соответственно), пользователю надоедает текущий видеоклип, и он выполняет одно из следующих действий: (i) с вероятностью  $p$  он начинает смотреть новый фрагмент



текущего видеоклипа, который еще не был загружен в буфер, (ii) с вероятностью  $1 - p$  он заканчивает просмотр текущего видеоклипа и начинает поиск нового, т.е. приостанавливает загрузку на время  $T_p$  ( $T_p$  — экспоненциально распределенная величина со средним  $\bar{T}_p$ , минимальным и максимальным значением  $T_p^{min}$  и  $T_p^{max}$  соответственно). Спустя  $T_p$  он начинает запрашивать сегменты другого видеоклипа. В экспериментах, представленных ниже, использованы следующие значения параметров, описывающих поведение пользователя:  $\bar{T}_w = 20$  с,  $T_w^{min} = 5$  с,  $T_w^{max} = 180$  с,  $\bar{T}_p = 30$  с,  $T_p^{min} = 1$  с,  $T_p^{max} = 180$  с,  $p = 2/3$ . Видеоклипы разбиты на сегменты длительностью  $\tau = 5$  с. Все сегменты закодированы в следующих битрейтах:  $\mathcal{B} = \{0.3, 0.5, 1.0, 1.8, 2.5, 5, 8, 16, 24, 40\}$  Мбит/с. Число пользователей  $N = 10$ .

В экспериментах сравниваются 2 подхода к выбору битрейтов и распределению ресурсов: (i) классический, включающий в себя алгоритмы выбора битрейта Instant или BOLA и планировщик PF на точке доступа; (ii) алгоритм, описанный в Разделах 3 и 4. Как уже было указано выше, предложенный алгоритм централизованного выбора битрейтов (от англ. Centralized Algorithm, CA) зависит от трех параметров:  $I_0$ ,  $A$  и  $Q_{opt}$ . Поэтому далее данное решение обозначается, как  $CA_{\{I_0\}_{\{A\}_{\{Q_{opt}\}}$ , где  $Q_{opt}$  измеряется в сегментах длительности  $\tau$ . Для оценки QoE используются следующие метрики: (i) средний геометрический битрейт  $\bar{b} = \exp\left[\frac{\sum_{i=1}^{N_s} \tau_i \log b_i}{\sum_{i=1}^{N_s} \tau_i}\right]$ , где  $\tau_i$  — продолжительность просмотра сегмента  $i$ ,  $N_s$  — общее число сегментов, просмотренных пользователями в течение эксперимента; (ii) средняя задержка воспроизведения первого кадра видеоклипа; (iii) средняя доля пауз воспроизведения.

Продолжительность каждого эксперимента 1000 с. Рассматривается среднее значение метрик по 50 прогонам эксперимента.

## 5.2 Анализ численных результатов

В первом эксперименте исследуется, зависимость метрик QoE от длительности просматриваемых пользователем видеоклипов. Результаты эксперимента представлены на рис. 1. Из графиков видно, что с уменьшением длительности видеоклипов наблюдается значительное ухудшение метрик QoE в случае локально работающих алгоритмов. Это вызвано недостатком информации о состоянии канала на клиенте в начале загрузки видеоклипа, а также большей частотой флуктуации числа пользователей, соревнующихся за беспроводной канал, приводящей к ошибкам в оценке пропускной способности канала локальными алгоритмами. В то же время централизованный алгоритм (кривая  $CA_{0.3c\_1.5\_1.5}$ ), во-первых, выдерживает заданные ограничения на задержку начала воспроизведения видеоклипа, во-вторых, обеспечивает долю пауз менее 0.01%, в-третьих, позволяет повысить средний битрейт сегментов более чем на 20% по сравнению с алгоритмом Instant. Кроме того, отметим, что повышение среднего геометрического битрейта в случае  $F(Q_i) = 1$ ,  $Q_i > 0$  (кривая  $CA_{0.3c\_1.0\_1.5}$ ) приводит к значительному

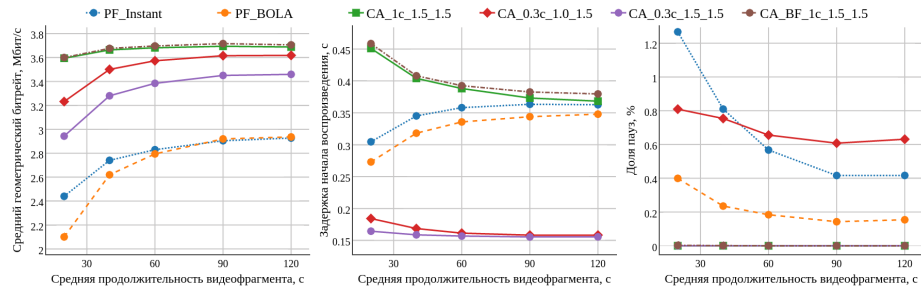


Рис. 1: Зависимость QoE от длительности просматриваемого видеоклипа

росту числа прерываний воспроизведения. Наконец, отметим, что результаты предложенного жадного алгоритма и полного перебора (см. кривую *CA\_BF*) почти совпадают, что говорит о том, что алгоритм, предложенный в Разделе 3.2, дает решение, близкое к оптимальному.

Во втором эксперименте исследуется, как выбор параметров функции запаса влияет на производительность разработанного алгоритма. Из графиков, представленных на рис. 2, видно, что с ростом величины  $A$  доля пауз воспроизведения значительно сокращается. Однако это приводит к снижению среднего битрейта. Также из приведенных графиков видно, что изменение требования на продолжительность начальной загрузки с 0,3 с до 1 с позволяет значительно увеличить средний битрейт загружаемых видеоклипов. Следовательно, настраивая параметры функции запаса, возможно добиться требуемого соотношения между метриками QoE. Например, чтобы максимизировать средний геометрический битрейт при доле пауз меньше, чем 0.1%, в рассматриваемом сценарии следует установить значения  $A = 1.5$  и  $Q_{opt} = 1.5$ . Значение  $I_0$  напрямую задает ограничение на задержку начала воспроизведения.

Однако производительность предложенного алгоритма зависит от конкретного сценария и может меняться с изменением состояния канала, числа активных пользователей и других факторов. Таким образом, возникает задача разработки алгоритма адаптации параметров функции запаса к состоянию сети, что и предполагается в качестве направления дальнейших исследований.

## 6 Заключение

Протокол SAND, являющийся расширением протокола MPEG-DASH, предоставляет возможность для обмена информацией между клиентом, сервером и промежуточными узлами. В данной работе был разработан алгоритм централизованного управления битрейтами видеопотоков и распределения канальных ресурсов на точке доступа Wi-Fi, использующий новые возможности, предоставляемые протоколом SAND. Результаты имитационного моде-

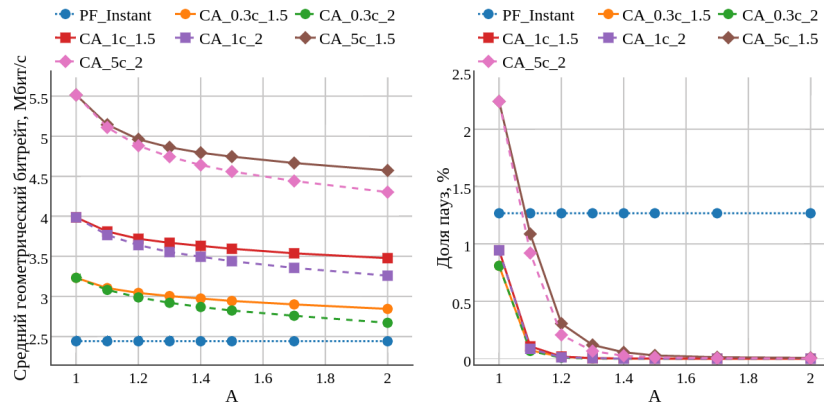


Рис. 2: Зависимость QoE от величины запаса

лирования показывают, что предложенное решение обеспечивает значительное увеличение всех метрик QoE по сравнению с существующими алгоритмами выбора битрейта, работающими локально на клиентских устройствах.

## Список литературы

1. Cisco, "Cisco visual networking index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper," 2016.
2. "What's the optimal length for a youtube vs. facebook video?" [Online]. Available: <http://tubularinsights.com/optimal-video-length-youtube-facebook/>
3. Wistia, "How long should your next video be?" [Online]. Available: <https://wistia.com/blog/optimal-video-length>
4. eMarketers, "What video ad length is best on facebook?" [Online]. Available: <https://www.emarketer.com/Article/What-Video-Ad-Length-Best-on-Facebook/1014438>
5. MPEG, "Iso/iec 23009-1:2014 MPEG-DASH 2nd Edition Specification," ISO, Tech. Rep.
6. Е. А.Н.Красилов, М.В.Любогощев, "Исследование передачи коротких видеопотоков mpeg-dash в сетях wi-fi," in *ИТuС-2016*, Репино, Санкт-Петербург, 2016.
7. MPEG, "Iso/iec 23009-5:2017 Dynamic adaptive streaming over HTTP (DASH) – Part 5: Server and network assisted DASH (SAND)," ISO, Tech. Rep.
8. Y. Wang, "Survey of objective video quality measurements," EMC Corporation Hopkinton, Tech. Rep.
9. L. R. Romero, "A dynamic adaptive http streaming video service for google android," Master's thesis, Royal Institute of Technology (KTH), Stockholm, October 2011.
10. R. U. K. Spiteri and R. K. Sitaraman, "Bola: Near-Optimal Bitrate Adaptation For Online Videos," in *IEEE INFOCOM*, 2016, pp. 1–9.

11. "Dash-js project." [Online]. Available: <https://github.com/Dash-Industry-Forum/dash.js/wiki>
12. A. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multi-user throughput allocation," *Operations Research*, vol. 53, no. 1, pp. 12–25, 2005.
13. "802.11-2016 - ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, Tech. Rep.
14. A. S. M. Andrews, L. Qian, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *INFOCOM'2005*, Miami, March 2005.
15. "Network simulator 3 (ns-3)." [Online]. Available: <https://www.nsnam.org/>