

РЕФЕРАТ

Расчетно-пояснительная записка 37 с., 7 рис., 19 ист., 1 прил.

Ключевые слова: ретрансляция видеопотока, повторная передача видеопотока, ABR, TCP, PAR, UDP, RTP, RTCP, RTSP, RTMP, HLS, MPEG-DASH, MPTCP, поток, видеопоток, адаптивный битрейт.

Объектом исследования является алгоритм организации ретрансляции видеопотоков.

Цель работы — проведение анализа алгоритмов организации ретрансляции видеопотоков.

В работе выполняется анализ предметной области и существующих решений, выделение критерий сравнений и сравнения существующих решений. На основе сравнения сделан вывод о существующих решениях.

СОДЕРЖАНИЕ

| | |
|---|-----------|
| РЕФЕРАТ | 3 |
| ОПРЕДЕЛЕНИЯ | 5 |
| ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ | 7 |
| ВВЕДЕНИЕ | 8 |
| 1 Анализ предметной области | 10 |
| 1.1 Ретрансляция видеопотоков | 10 |
| 1.2 Организация ретрансляции видеопотока | 10 |
| 1.3 Модель взаимодействий открытых систем | 11 |
| 2 Обзор существующих решений | 13 |
| 2.1 TCP | 13 |
| 2.2 UDP | 15 |
| 2.3 RTP | 16 |
| 2.4 RTCP | 17 |
| 2.5 RTSP | 18 |
| 2.6 RTMP | 21 |
| 2.7 HLS | 22 |
| 2.8 MPEG-DASH | 24 |
| 2.9 MPTCP | 26 |
| 3 Сравнение существующих решения | 28 |
| ЗАКЛЮЧЕНИЕ | 33 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 34 |
| ПРИЛОЖЕНИЕ А | 37 |

ОПРЕДЕЛЕНИЯ

Пакет — это блок данных, содержащий информацию, необходимую для доставки [4].

Задержка — время, которое требуется для передачи данных отправки данных по Сети [4].

Джиттер — изменчивость или нестабильность во времени задержки при передаче данных [4].

Пропускная способность — максимально возможное количество данных, которые могут быть переданы через канал за некоторый промежуток времени [4].

Мультиплексирование — образование из нескольких отдельных потоков общего агрегированного потока, который может быть передан по одному физическому каналу связи [5].

Компьютерная сеть — набор связанных между собой автономных компьютеров. Два компьютера называются связанными между собой, если они могут обмениваться информацией [6].

Канал передачи данных — канал электросвязи для передачи сигналов данных [6].

Трансляция — непрерывная передача аудио или видеоданных с источника на платформу для дальнейшего распространения [9].

Ретрансляция — повторная передача видеоданных без изменения его содержания [9].

Видеопоток — последовательность видеоданных, которая передается по компьютерным сетям или другим каналам связи [9].

Протокол — набор соглашений логического уровня.

Потоковая передача данных (передача потоков) — способ передачи, при котором транспортировка и воспроизведение мультимедийных данных на удаленном компьютере осуществляются в режиме реального времени [9].

Соединение — канал транспортного уровня, установленный между двумя программами с целью обмена данными [13].

Сеанс — механизм передачи непрерывного медиапотока или запуск потока с помощью данных управления [13].

Битрейт — количество бит, используемых для передачи или обработки данных в единицу времени [16].

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В текущей расчетно-пояснительной записке применяются следующие сокращения и обозначения.

ISO — International Standards Organization, международной организацией по стандартизации.

OSI — Open Systems Interconnect, взаимодействия открытых систем.

TCP — Transmission Control Protocol, протокол управления передачей.

UDP — User Datagram Protocol, протокол пользовательских дейтаграмм.

PAR — Positive Acknowledgment with Retransmission.

RTP — Real-Time Transport Protocol, транспортный протокол реального времени.

RTCP — Real-Time Transport Control Protocol, протокол управления передачей в реальном времени.

RTSP — Real-Time Streaming Protocol, потоковый протокол реального времени.

RTMP — Real-Time Messaging Protocol, протокол обмена сообщениями в реальном времени.

HTTP — HyperText Transfer Protocol, протокол передачи гипертекста.

HLS — HTTP Live Streaming, прямая передача потоковых данных по протоколу HTTP.

MPEG-DASH — Dynamic Adaptive Streaming over HTTP, технология адаптивной потоковой передачи данных.

MPTCP — Multipath TCP, многопутевой TCP.

ABR — Adaptive Bit-Rate Streaming, трансляция с адаптивным битрейтом.

QoS — Quality of Service, качество обслуживания.

ВВЕДЕНИЕ

Одной из тенденций развитий современного общества является развитие интернет-технологии, которые проникли в каждый аспект жизни человека. Ежедневно растет количество устройств и платформ с поддержкой видеоконтента, критически важных предприятий и государственных служб зависят от корректной работы Сети [1].

Поэтому повышаются требования к качеству и надежности передачи видеoinформации всем пользователям. Провайдеры сетей стремятся обеспечить стабильность передачи информации, улучшая существующую инфраструктуру, но существуют факторы которые могут оказать негативное влияние на процесс передачи видеопотоков [2, 3]:

- нагрузка на сеть в текущий момент времени;
- устаревшая или неэффективная инфраструктура;
- нестабильное соединение может привести к потере пакетов данных;
- физические помехи и условия окружающей среды;
- низкая пропускная способность.

Таким образом, чтобы доставить видеoinформацию произвольному количеству пользователей возникает необходимость организации ретрансляции видеопотоков от устройства-захвата для обеспечения стабильной и надежной передачи видеoinформации. Организация ретрансляции видеопотоков становится ключевым шагом для решения этих проблем. Путем передачи трансляции видео через устройства-захвата на сервере, задача которого повторно передавать или хранить видеоматериал для повторной передачи пользователям.

Целью работы является проведение анализа алгоритмов организации ретрансляции видеопотоков.

Чтобы достигнуть поставленной цели, необходимо решить следующие задачи:

- провести анализ предметной области ретрансляции видеопотоков;
- провести обзор существующих алгоритмов организации ретрансляции видеопотоков;
- сформулировать критерии сравнения алгоритмов организации ретрансляции видеопотоков;
- классифицировать существующие алгоритмы организации ретрансляции видеопотоков.

1 Анализ предметной области

1.1 Ретрансляция видеопотоков

Ретрансляция видеопотоков — это повторная передача видеопотоков через компьютерные Сети с целью распространения или обработки. Ретрансляция начинается с захвата видеопотока с источника, такого как веб-камера, IP-камера, экран компьютера или другое устройство, которое генерирует видеоданные. Ретрансляция может быть использована для одновременной трансляции видеопотока на несколько устройств, при этом видеопотоки могут подвергаться различным видам обработки: сжатию, кодированию, фильтрации или масштабированию, в зависимости от конкретных потребностей.

1.2 Организация ретрансляции видеопотока

Трансляция формируется надежной системой ретрансляции, в качестве такой системы выступает сервер, который принимает видеопоток от устройства-захвата. Трансляция совершается путем использования протокола, таких как UDP, RTP, RTSP, RTMP и других. Сетевой протокол в свою очередь должен гарантировать доставку и управление трансляцией, при этом подстраивающийся под условия Сети. Таким образом для передачи обычно используется один из двух способов.

Последовательный — видеофайл воспроизводится с жесткого диска сервера провайдера услуг. Как правило, при передаче таким способом качество изображения и звука выше. К недостаткам стоит отнести то, что невозможно переключить ролик с одного момента на другой, не дождавшись его буферизации. То есть, интересующий фрагмент должен быть загружен, чтобы пользователь смог его просмотреть.

В реальном времени — требует наличия потокового сервера. Этот метод больше подходит для передачи видеофайлов большой длительности. Пользователь может выбрать место, с которого он хочет начать просмотр. Также

этот вид потокового передачи мультимедиа используется для трансляции с веб-камеры или захвата экрана.

1.3 Модель взаимодействий открытых систем

Все устройства в компьютерной сети взаимодействуют по концептуальной модели взаимодействия открытых систем OSI/ISO. Её предложили в 1984 году инженеры из ISO, которая работала над единым стандартом передачи данных по Интернету. Модель OSI включает семь уровней, причём каждый из них выполняет определённую функцию и определяется набором протоколов [4, 5].

Модель TCP/IP — это стек протоколов, которые задают правила передачи данных по Сети. TCP/IP состоит из 4 уровней и покрывает функционал семи уровневой модели OSI/ISO [4, 5].

На рисунке 1.1 представлено сравнение моделей OSI/ISO и TCP/IP.

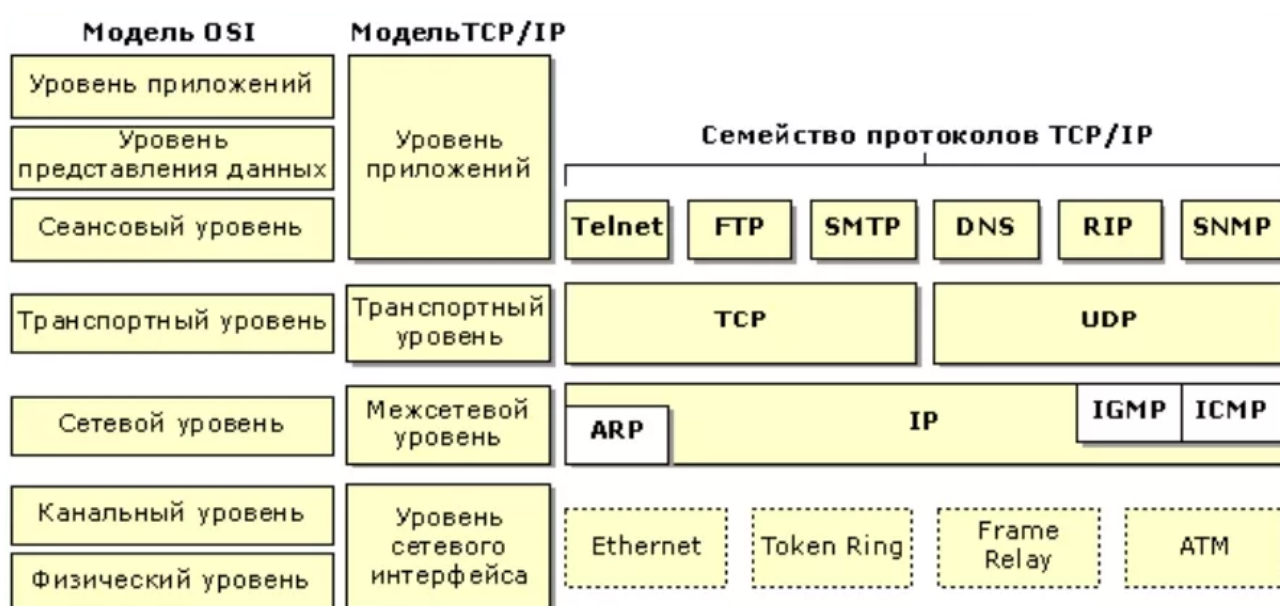


Рисунок 1.1 – Модель OSI и TCP/IP [4, 5]

В таблице 1.1 представлены уровни ТСП/IP.

Таблица 1.1 – Уровни ТСП/IP

| Обозначение | Уровень | Назначение |
|-------------|--------------|---|
| L4 | Прикладной | Взаимодействие с приложениями и процессами; установление, поддержание и корректного завершения сеансов соединений, аутентификация; корректное отображение данных. |
| L3 | Транспортный | Обеспечивает надежную доставку данных со сквозным обнаружением и устранением ошибок. |
| L2 | Сетевой | Поиск и выбор оптимального маршрута между географически удаленными сетями. |
| L1 | Канальный | Отвечает за механические, электрические, процедурные и функциональные характеристики установления и завершения соединения между устройствами; отвечает за методы передачи и контроля данных; определяет формат данных для передачи. |

2 Обзор существующих решений

Согласно общепринятым нормам взаимодействия открытых систем, передачи данных между сервером и клиентом регламентируется протоколами транспортного уровня, на котором выполняется сегментация данных от отправителя и организацию потока на получателе [4, 6]. Вместе с этим решаются задачи:

- мультиплексирование;
- установление и управление соединением;
- управление потоком;
- надежность передачи данных.

Таким образом, протоколы прикладного и транспортного уровня для передачи видеопотоков строятся поверх двух важных протоколов транспортного уровня — TCP и UDP.

2.1 TCP

TCP — протокол с установлением соединения, обеспечивающий надежную доставку данных со сквозным обнаружением и устранением ошибок. Надежность в TCP обеспечивается благодаря механизму подтверждения приема повторной передачей PAR, в случае его потери или искажения. Согласно TCP гарантируется доставка данных и соединение ложно установиться до начала передачи данных, путем «тройного рукопожатия». По такому же принципу происходит разъединение соединения [4, 5].

На рисунках 2.1 представлены процессы соединения и разъединения между сервером и клиентом.

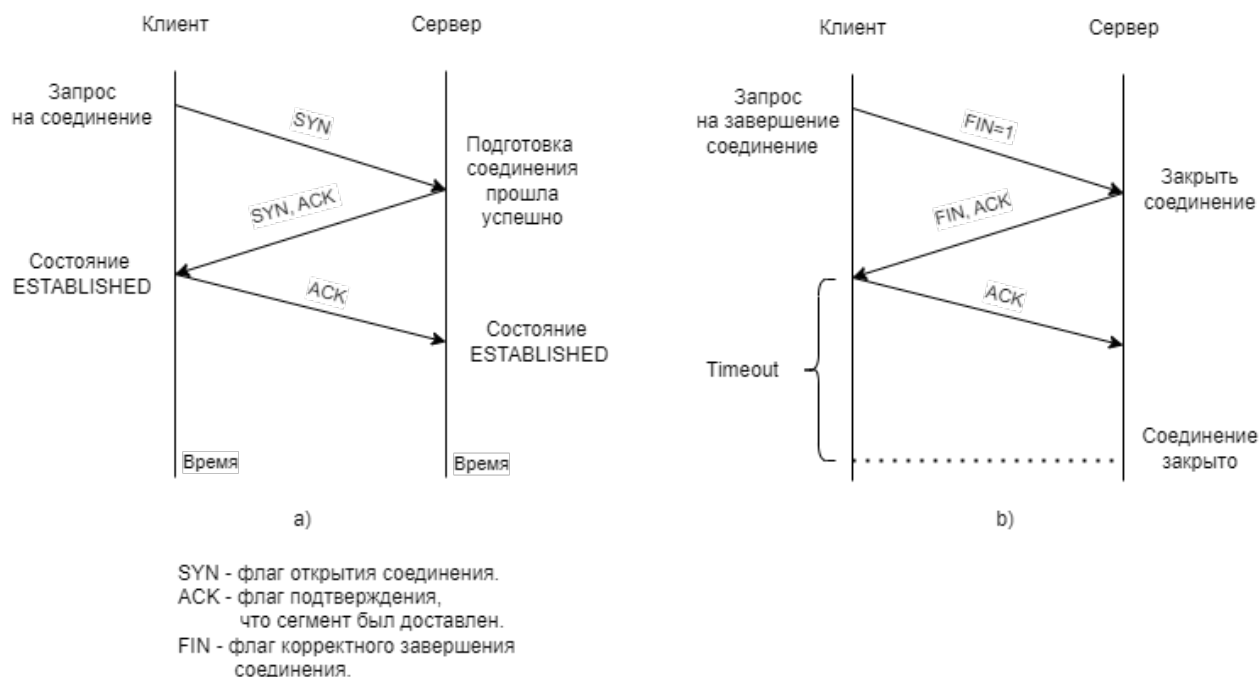


Рисунок 2.1 – Схема процессов соединения (a) и разъединения (b) [4]

Также TCP регулирует скорость отправки в соответствии со своей оценкой пропускной способности Сети, потери и задержку, и если они ухудшились, это снизит скорость потока, а если они улучшились, это увеличит скорость потока. В случае потери пакета во время передачи, пакет передается повторно [4, 5, 7].

TCP использует метрики, собираемые в процессе работы протокола, чтобы определить дальнейшее взаимодействие. Из полученных метрик вычисляется размер TCP-окна, показывающий количество байт, которые принимающая сторона готова принять в текущий момент без подтверждения [4].

RTT (Round-Trip Time) необходимое время для отправки данных клиенту и обратно, которое включает время задежки на передачу данных и подтверждения получения пакета. У RTT есть отклонение — devRTT (deviation of Round-Trip Time). DevRTT и среднее RTT используются для управления потоком, определения промежутка времени, после которого пакеты будут считаться утерянными [8].

RTT рассчитывается по следующей формуле:

$$RTT_i = t_r - t_0,$$

где t_r — время получения подтверждения пакета, t_0 — время отправки пакета, RTT_i — RTT i -ого пакета.

Среднее значение RTT рассчитывается по следующей формуле:

$$RTT_{average} = \alpha RTT_{average} - \beta RTT_i,$$

где $\alpha = 0,875$ и $\beta = 0,125$.

Отклонение RTT рассчитывается по следующей формуле:

$$DevRTT_i = |RTT_i - RTT_{average}|,$$

Среднее отклонение RTT рассчитывается по формуле:

$$DevRTT_{average} = \alpha DevRTT_{average} - \beta DevRTT_i,$$

где $\alpha = 0,75$ и $\beta = 0,25$.

Timeout рассчитывается по следующей формуле:

$$timeout = RTT_{average} - 4DevRTT_{average},$$

2.2 UDP

UDP — протокол без установления соединения. Поскольку протокол UDP таких проверок не совершает и нет механизма PAR, что обеспечивает более быструю передачу данных, но не предоставляет надежность их передачи.

UDP выполняет три действия:

- идентифицирует процесс отправки и получения, используя номера портов;
- запускает проверку на ошибку в заголовке;
- записывает проверку в заголовке.

Поэтому UDP не восстанавливает потерянные пакеты [4, 5, 7].

Таким образом, UDP применяется для передачи видеопотоков в реальном времени, потеря пакета данных не приведет к задержке видео для восстановления пакета, но при этом качество видео будет ухудшаться в зависимости от того, сколько пакетов было потеряно.

Для передачи видеопотоков используются протоколы транспортного уровня без установления соединения. Так как UDP не может гарантировать качество видео при потере последовательности пакетов, то его используют в сочетании с специальными протоколами, которые преобразуют исходные данные таким образом, что они могут быть переданы в сеть, как непрерывная последовательность, обеспечивается наилучшая потоковую передачу видео, чем TCP. Использование передовых технологий сжатия и буферизации позволяет просматривать потоковый контент с любого места, не дожидаясь его полной загрузки на компьютер пользователя. При этом протоколы формируют полезную нагрузку пакетов, включая необходимые поля и данные [9].

2.3 RTP

RTP [10] — прикладной протокол сквозной доставки данных в режиме реального времени, включая видео и аудио.

RTP позволяет компенсировать негативное влияние задержек на качество видео и аудио, но при этом не гарантирует своевременную доставку пакетов, т.е. не обеспечивает QoS, также не предоставляет функции исправления ошибок и управления потоком. RTP совместно используется с UDP используя его функции. RTP предусматривает индикацию типа полезной на-

грузки и порядкового номера пакета в потоке, а также применение временных меток. Отправитель помечает каждый RTP-пакет временной меткой, получатель извлекает ее и вычисляет суммарную задержку. Разница в задержке разных пакетов позволяет определить джиттер и смягчить его влияние — все пакеты будут выдаваться приложению с одинаковой задержкой [10, 11, 12].

Однако RTP может использоваться с другими подходящими базовыми сетевыми или транспортными протоколами. Также RTP поддерживает передачу данных нескольким адресатам с использованием многоадресной рассылки, если это предусмотрено базовой сетью [10].

На рисунке 2.2 представлена схема передачи пакета между клиентом и сервером по протоколу RTP поверх UDP.

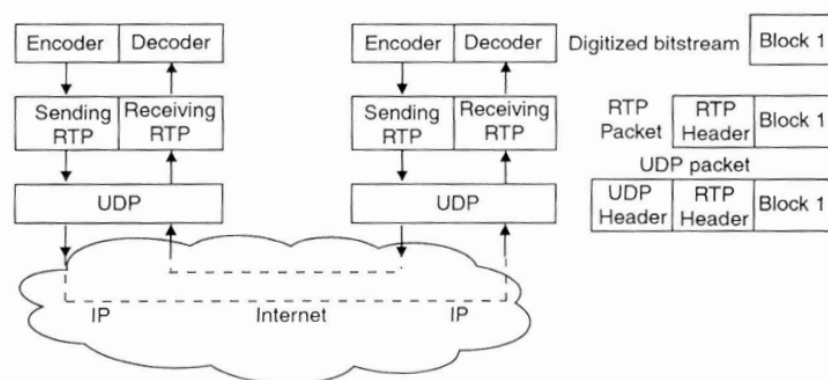


Рисунок 2.2 – Схема передачи пакета между клиентом и сервером по RTP поверх UDP [12]

2.4 RTCP

RTCP [10] — протокол управления, созданный для совместной работы с RTP, он помогает осуществлять синхронизацию видео и звука, обеспечивать QoS, т.е. обратную связь участников сеанса RTP и контроль качества передачи данных. Также RTCP передает сведения о числе переданных и потерянных пакетов, значения джиттера, задержке и т.д. Базовый протокол должен обеспечивать мультиплексирование пакетов данных и управления, в UDP это обычно реализуется с использованием отдельных номеров портов.

Основными функциями RTCP являются:

- мониторинг качества обслуживания и контроль перегрузки;
- идентификация источника RTP;
- оценка размера сеанса и масштабирование.

Пакеты RTCP содержат прямую информацию для мониторинга качества обслуживания. Отчеты отправителя (SR) и отчеты получателя (RR) обмениваются информацией о потерях пакетов, задержке и джиттера. Эта информация может быть использована для реализации механизма управления потоком, подобного TCP. Средство управления сетью может отслеживать загрузку Сети на основе пакетов RTCP без получения фактических данных или обнаруживать неисправные участки Сети [10, 12].

На рисунке 2.3 представлена схема передачи пакета между клиентом и сервером по протоколу RTCP.

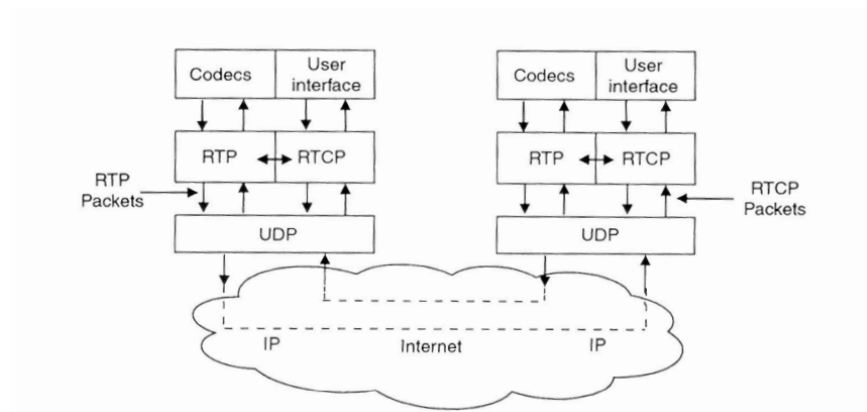


Рисунок 2.3 – Схема передачи пакета между клиентом и сервером по RTCP [12]

2.5 RTSP

RTSP [13] — протокол управления прикладного уровня для инициализации и направления потоковых данных от видеосервера, реализующий возможности «удаленного управления», который был разработан в 1996 году для управления развлекательными и коммуникационными системами видеотрансляций.

RTSP требует двух типов компонентов для успешного выполнения потоковой передачи — клиента и сервера. Сервер обслуживает потоковые данные, а клиент запрашивает их. отдельного сервера для приема запросов от нескольких клиентов и потоковой передачи данных, который поддерживает сеанс RTSP постоянного соединения с клиентами RTSP, помеченный идентификатор, при этом во время соединения сеанс не привязывается к использованию транспортного протокола, поскольку он реализует надежность на прикладном уровне. RTSP не навязывает использования определенного формата медиафайлов. Во время сеанса, клиент RTSP может открывать и закрывать множество надежных транспортных подключений к серверу для выдачи запросов RTSP. Таким образом, могут передаваться несколькими различными способами [13]:

- постоянные транспортные соединения, используемые для нескольких транзакций запрос-ответ;
- одно подключение на транзакцию запроса-ответа;
- режим без установления соединения.

RTSP управляет потоком, который может быть отправлен по отдельному протоколу, независимому от канала управления. Так RTSP используется TCP для передачи и приема команд управления, в то время как данные передаются по UDP. При этом RTSP использует RTP и RTCP для передачи потоков в реальном времени, что в свою очередь обеспечивает воспроизведение RTSP-потока во время загрузки потока и получение потока продолжается если сервер не получает запросов, что называется сегментация потока передачи данных. Кроме того, в течение всего срока службы один медиапоток может управляться запросами RTSP, выдаваемыми последовательно по разным TCP-соединениям, поэтому сервис поддерживать «состояние сеанса», чтобы сопоставлять запросы с потоком [13].

RTSP не конкретизируется на формате передачи данных, поэтому позволяет клиенту выбрать подходящую комбинацию видеоданных, но поддержи-

ка кодеков ограничена, что может ограничивать качество видео. У RTSP передается поток с минимальной задержкой, которая достигается до 2 секунд. Каждый поток идентифицируется по URL-адресу RTSP, который указывает на сервер обрабатывающий этот поток. Помимо параметров видео, необходимо определять сетевой адрес назначения и порт, что выделяет несколько режимов работы [13]:

- многоадресная рассылка и сервер выбирает адрес;
- многоадресная рассылка и клиент выбирает адрес;
- одноадресная рассылка.

На рисунке 2.4 представлена схема передачи видеопотока между серверами и клиентом по протоколу RTSP.

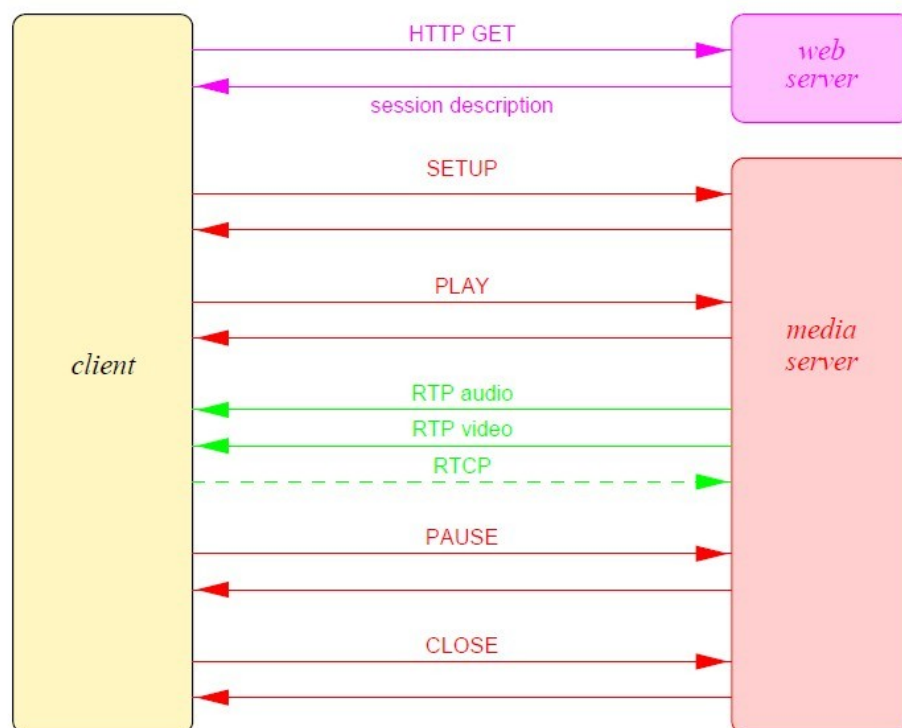


Рисунок 2.4 – Схема передачи видеопотока между серверами и клиентом по протоколу RTSP

Основные методы протокола, который идентифицируется по запросу:

- DESCRIBE — запрос описания содержимого, например, в формате SDP;
- OPTIONS — запрос поддерживаемых методов;
- PLAY — запрос начала вещания содержимого;
- PAUSE — запрос временной остановки вещания;
- RECORD — запрос на записывание содержимого сервером;
- REDIRECT — запрос на перенаправление на другое содержимое;
- SETUP — запрос установки транспортного механизма для содержимого;
- ANNOUNCE — запрос на обновление данных описания содержимого;
- TEARDOWN — запрос на остановку потока и освобождение ресурсов.

По функциональности RTSP совпадает с HTTP и может взаимодействовать с ним, только если первоначально взаимодействие с потоком осуществляется через веб-страницу с помощью специального программного обеспечения. Однако RTSP принципиально отличается от HTTP тем, что доставка данных осуществляется вне диапазона по другому протоколу, но для работы с кэшами, прокси-серверами и аутентификацией используется аналогичный функционал HTTP. Таким образом RTSP несовместим с HTTP. Также RTSP требует большей пропускной способности, что делает его менее подходящим для мобильных устройств [13, 14].

2.6 RTMP

RTMP [15] — протокол прикладного уровня, разработанный Adobe в 2009 году, для высокоэффективной потоковой передачи видео в реальном времени через Интернет между Adobe Flash и сервером. RTMP предоставляет двустороннюю многоканальную службу передачи сообщений с использованием надёжного транспортного потока путем использования протокола TCP, используемого для передачи параллельных потоков видео, аудио и текстовых сообщений с второстепенной тактовой информацией между парой общающихся пользователей. Различные классы сообщений получают разные приорите-

ты, что влияет на их очередность в транспортном потоке, когда пропускная способность ограничена.

RTMP нацелена на обеспечение стабильной и плавной передачи увеличивающихся объемов данных, необходимых для передачи и приема видео в реальном времени, что достигается посредством сегментации потока данных на небольшие одинаковые части (аудио — 64 байта, видео — 128 байтов), их последовательную передачу через постоянное TCP-соединение на принимающее устройство, которое затем снова собирает их в видеопоток. Для трансляции с устройства-источника видеопоток передается сервер кодировщик, данный этап называется «первой милей». Затем сервер обрабатывает поток и передает его дальше для ретрансляции на устройства клиентов, этап называется «последняя миля». С этого момента поток перехватывается поток другой связи (HLS, MPEG-DASH или WebRTC) или Adobe Flash [15].

Таким образом постоянное соединение обеспечивает надежность доставки видеопотока, которое не отключается при нестабильной передаче данных между сервисом и клиентом, что обеспечивает низкую задержку передачи данных до 5 секунд. Благодаря постоянному соединению клиент может подключаться к и отключаться от трансляции, перематывать ее, но оно трудно поддерживается в определенных условиях Сети. У RTMP низкая пропускная способность, что может стать причиной потери пакетов без восстановления, что может ухудшить качество видео. RTMP протокол не поддерживает HTML5-плеерами и не совместим с HTTP. Поскольку он требует аутентификации, RTMP является гораздо более безопасным протоколом потоковой передачи. [15].

2.7 HLS

HLS [16] — протокол прикладного уровня для потоковой передачи медиа на основе HTTP, разработанный компанией Apple в 2009 году как часть программного обеспечения QuickTime, Safari, OS X и iOS. В основе работы

лежит принцип разбиения цельного потока на небольшие фрагменты, последовательно скачиваемые по HTTP, развертывая контент с помощью обычных веб-серверов и сетевой доставки контента. Поскольку протокол использует стандартный протокол HTTP.

На рисунке 2.5 представлена схема организации передачи видеопотока по протоколу HLS.

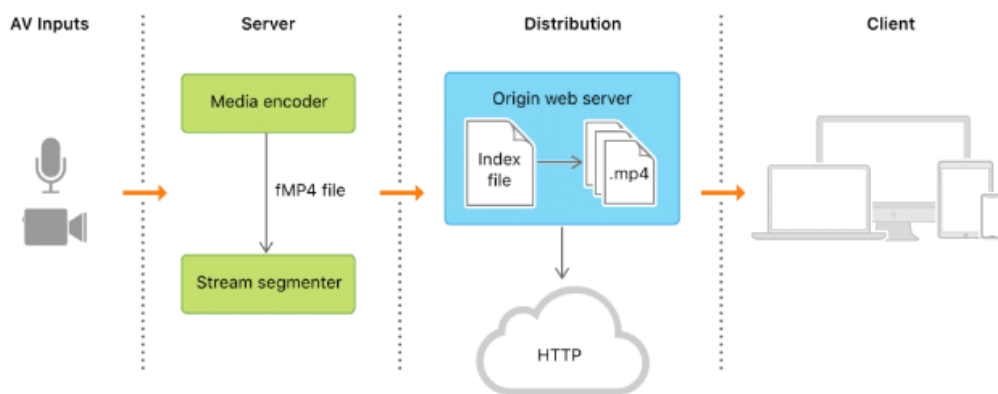


Рисунок 2.5 – Схема организации передачи видеопотока по HLS [16]

Устройства-источники транслируют видеопоток на сервер, где данные поступают HLS-кодера для передачи потока в реальном времени и кодируются в формате H.265 или HEVC и может соответствовать стандартам спецификации Apple. После чего поток разбивается на фрагменты равного размера и создается файл, который называется «списком воспроизведения» в формате M3U, содержащий названия, местоположение и последовательность воспроизведения фрагмента, наряду с метаданными. Данный процесс называется упаковкой пакета для дальнейшей передачи. После упаковки переданного потока, он передается клиенту, где данный поток принимает любой плеер, который принимает HLS, например HTML5-плеер. Воспроизведение начинается с загрузки списка воспроизведения, а затем использования того, чтобы собрать последовательность фрагментов пришедших непрерывным потоком. Клиент скачивает первый индексный файл через URL и далее несколько доступных файлов медиа. Программное обеспечение для проигрывания собирает всё в

последовательность для воспроизведения. Существует два режима работы HLS — по запросу и трансляции в реальном времени. В режиме по запросу список воспроизведения содержит информацию о всей последовательности фрагментов. В режиме трансляции реального времени список воспроизведения содержит только ссылки на последние несколько фрагментов, кроме того при последующих обращениях к списку воспроизведения, сегменты будут меняться, отражая текущее состояние трансляции [16].

HLS предусматривает поддержку адаптивного битрейта, который предусматривает наличие нескольких одновременно доступных потоков, каждый из которых может содержать одинаковый контент, закодированный в разных битрейтах, а также имеющий другие отличающиеся характеристики. По мере воспроизведения клиент может выбирать из числа нескольких доступных потоков, что позволяет адаптировать сессию к внешним условиям передачи по Сети, на основе чего происходит переключение потоков в ответ на изменение пропускной способности Сети, то есть во время воспроизведения появляется возможность менять качество передаваемого видеопотока. Так как HLS основан на HTTP то ему предоставляется функционал HTTP, что обеспечивает шифрование мультимедиа и аутентификация клиента [16].

Трансляция по HLS приводит к задержке, что влияет на передачу потоков в реальном времени до 10 секунд. LHLS — это модификация HLS для потоковой передачи в реальном времени, где задержка достигается до 2 секунд [16].

2.8 MPEG-DASH

MPEG-DASH [17] — прокол прикладного уровня на основе HTTP для передачи потокового видео, созданный ISO и MPEG в 2011 году и опубликован в 2012 году, но окончательно был обновлен в 2019 году как MPEG-DASH ISO/IEC 23009-1:2019 [18].

MPEG-DASH основан на адаптивной потоковой передаче ABR, как и в целом работает аналогично HLS. Упаковщик MPEG-DASH разбивет поток на фрагменты, которые могут содержать любые мультимедийные данные, однако придерживается формата мультимедийных файлов ISO — MP4 или используется в формате MPEG-TS стандарта MPEG-2 и отправляются на основе TCP. Вместе с фрагментами формируется MPD — файл манифест или список воспроизведения в формате XML документа, который содержит информацию для описания фрагментов — видео, аудио, скорость звука и видео, последовательность фрагментов и т.д.

На рисунке 2.6 представлена схема организации передачи видеопотока по протоколу MPEG-DASH.

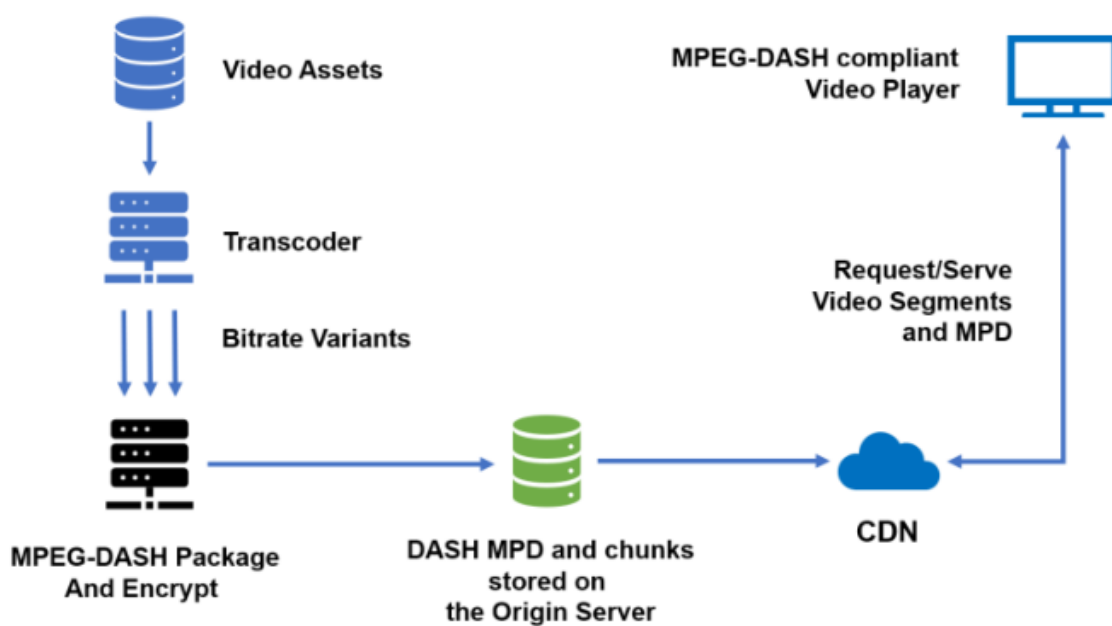


Рисунок 2.6 – Схема организации передачи видеопотока по MPEG-DASH [17]

Упакованное видео и манифест хранятся на сервере в ожидании доставки проигрывателя на устройстве клиента. По запросу на воспроизведения присылается MPD-файл, после получения которого анализируется видеопоток. После чего запрашиваются фрагменты потока в определенной последовательности, которые собираются повторно на устройстве клиента на осно-

ве полученного MPD-файла. DASH используется адаптивный битрейт ABR, что способствует переключение между потоками в зависимости от пропускной способности Сети и возможностей устройства клиента, т.е. одно и то же видео может быть представлено в разных вариантах. MPEG-DASH поддерживает кодирования с использованием любого кодека (H.265, H.264, VP9 и т.д.) и разрешения до 4K [17, 18].

Высокая задержка MPEG-DASH обусловлена главным образом потерей сетевых пакетов и методом восстановления, используемым во всех сетях на основе TCP, которая достигает до 10 секунд. И хотя MPEG-DASH предлагает некоторый контроль над размером фрагмента мультимедиа, возможность уменьшить задержку ограничена — особенно, если сервер требует загрузки среднего фрагмента определённого размера [17, 18].

2.9 MPTCP

MPTCP [19] — протокол транспортного уровня, использующий несколько каналов передачи данных, который является набором расширений к протоколу TCP, разработанный IETF в 2013 году.

Протокол использует обратное мультиплексирование, разбивая один MPTCP-поток на несколько TCP-потоков, каждый из которых проходит через отдельный канал передачи данных и имеет отдельный идентификатор. Каждый TCP-поток может использовать разные пути для передачи данных. На устройстве клиента MPTCP собирает все TCP-потоки в MPTCP-поток, т.е. в единую последовательность. MPTCP обеспечивает контроль за передачей данных, адаптируясь к пропускной способности Сети. К примеру, если один из путей недоступен или обладает плохим качеством связи, то автоматически переключается на другой путь. Таким образом, MPTCP поддерживает беспроводные сети, где качество соединений может меняться в зависимости от местности и времени [19, 20].

МРТСП может использовать все доступные сетевые интерфейсы для передачи данных одновременно и балансировать нагрузки между интерфейсами, что позволяет достичь увеличение пропускной способности и уменьшение задержки, так как данные могут быть переданы по самому быстрому каналу. Однако, внедрение МРТСП требует поддержки со стороны устройств, настройки и управления соединением.

3 Сравнение существующих решения

В качестве критерий сравнение протоколов для организации ретрансляции видеопотоков необходимо наличие тех или иных возможностей:

- 1) уровень модели TCP/IP;
- 2) установка соединения;
- 3) гарантированная доставка;
- 4) упорядоченная передача — данные доставляются в том порядке, в котором были отправлены;
- 5) поддержка нескольких интерфейсов;
- 6) зависимость — протокол реализован таким образом, что работает поверх нижележащих протоколов;
- 7) задержка;
- 8) многопоточность;
- 9) управление и контроль потоком.

В таблицах 3.2–3.1 представлено сравнение рассмотренных протоколов для организации передачи видеопотоков.

Таблица 3.1 – Сравнение рассмотренных протоколов (Часть 1)

| Протокол | Номер критерия | | | | | | | | |
|----------|----------------|-----------------------|--|-----|-----|-------------|------------|-----|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| ТСР | L3 | «Тройное рукопожатие» | Механизмы подтверждения, повторной передачи и контроля порядка пакетов | Да | Нет | Нет | от 100 мс. | Нет | ТСР-окна |
| UDP | L3 | Нет | Нет | Нет | Нет | Нет | от 1 мс. | Нет | Нет |
| МРТСП | L3 | «Тройное рукопожатие» | Да | Да | Да | ТСР | 30-100 мс. | Да | Использует несколько путей для передачи данных |
| RTT | L4 | Нет | Нет | Нет | Нет | ТСР/ UDP | от 10 мс. | Да | Контроль битрейта |
| RTСП | L4 | Нет | Нет | Нет | Нет | RTT | от 10 мс. | Да | Сбор информации о качестве передачи данных, синхронизация участников и управления параметрами потока данных |

Таблица 3.2 – Сравнение рассмотренных протоколов (Часть 2)

| Протокол | Номер критерия | | | | | | | | |
|------------------|----------------|-----------------------|-------------|-------------|-----|----------------------|----------|----|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| RTSP | L4 | Постоянное соединение | Функции TCP | Функции TCP | Нет | RTSP/ TCP/ UDP | 2 с. | Да | Управление воспроизведением, паузой, перемоткой и другими параметрами потока |
| RTMP | L4 | Постоянное соединение | Функции TCP | Функции TCP | Нет | TCP | 5 с. | Да | Контроль битрейта и параметров передачи потока |
| HLS | L4 | Нет | Функции TCP | Функции TCP | Нет | HTTP | от 10 с. | Да | Использует ABR, сегментацию потока со списком воспроизведения и контроль потока путем двух режимов работы |
| MPEG-DASH | L4 | Нет | Функции TCP | Функции TCP | Нет | HTTP | от 10 с. | Да | Использует ABR и сегментацию потока с манифестом |

Протокол UDP обладает ограниченным набором действий, на основе которого можно будет реализовать свой протокол для передачи видеопотока. Как было уже сказано TCP в Сети не используется для передачи видеопотоков из-за высокой задержки, хотя присутствуют TCP-окна, что позволяет подстроиться под пропускную способность Сети. Все же TCP и UDP напрямую не используются для передачи видеопотоков, но поверх них реализуются протоколы RTP, RTSP, RTMP, HLS и MPEG-DASH.

Протокол RTP предоставляет минимальные набор действия для передачи потоков в реальном времени, потоки на отправку и получения данных. Хотя RTP обладает неупорядоченной доставкой, на получателе восстанавливает порядок данных на основе временных меток, чтобы обеспечить корректное воспроизведение в реальном времени. Протокол RTCP работает вместе с RTP и предназначен для сбора статистики и информации о RTP-потоке, обеспечения QoS, контроль стабильности потока и синхронизации участников сессии. Таким образом благодаря RTCP/RTP обеспечивается передачи видеопотока с адаптацией к Сети, но по сравнению с RTSP и RTMP набор функций намного меньше.

RTSP и RTMP предоставляют набором функций для работы пользователю с видеопотоком и обеспечивает упорядоченную доставку и реализованы поверх с RTP/TCP/UDP, что делает их более гибкими, но RTMP на получателе принимается Adobe плеером.

HLS и MPEG-DASH отличается в основном поддержкой ABR, что делает данные инструменты более гибкими к условиям Сети, но имеют достаточно высокую задержку в сравнение с остальными протоколами. HLS и MPEG-DASH подходят для адаптивной передачи на разные устройства, путем совместимости с HTML5.

MPTCP хоть и предоставляет возможность использовать сразу несколько каналов, но имеет избыточных функций, так как основывается на TCP. MPTCP находится в разработке и из проведенного сравнения видно, что дан-

ный протокол является надежным и адаптируемым к условиям Сети, что делает его многообещающим в качестве использования, разработке алгоритма или улучшения для передачи видеопотоков. Также данный протокол может использоваться совместно с протоколами RTSP, RTP или RTMP, по той причине, что данные протоколы были реализованы поверх TCP или UDP.

ЗАКЛЮЧЕНИЕ

В ходе данной работы были изучены:

- проведен анализ предметной области ретрансляции видеопотоков;
- проведен обзор существующих алгоритмов организации ретрансляции видеопотоков;
- сформулированы критерии сравнения алгоритмов организации ретрансляции видеопотоков;
- классифицированы существующие алгоритмы организации ретрансляции видеопотоков.

Была достигнута поставленная цель данной работы. В ходе работы было выяснено, что многие протоколы для организации передачи видеопотока реализованы поверх одного из протоколов транспортного уровня TCP и UDP, или могут использовать их оба — TCP для надежности, а UDP для уменьшения задержки. Для организации передачи данных были рассмотрены протоколы: RTP/RTCP, RTSP, RTMP, HLS, MPEG-DASH и MPTCP, каждый из которых имеют свои преимущества и недостатки.

Список использованных источников

1. Mediascope представила данные о медиапотреблении россиян в 2022 году [Электронный ресурс]. Режим доступа: <https://licensingrussia.ru/article/11208-mediascope-predstavila-dannye-o-mediapotreblenii-rossiian-v-> (Дата обращения: 30.10.2023).
2. Обмен видеоданными посредством Ethernet [Электронный ресурс]. Режим доступа: <https://rep.bntu.by/bitstream/handle/data/94438/20-22.pdf?sequence=1&isAllowed=y> (Дата обращения: 30.10.2023).
3. Тренды медиапотребления в России [Электронный ресурс]. Режим доступа: <https://adindex.ru/publication/analitics/search/2023/11/20/317349.shtml> (Дата обращения: 30.10.2023).
4. Хант Крэйг. TCP/IP: Сетевое администрирование, 3-е издание. — СПб: Символ-Плюс, 2008. — С. 816.
5. Чешпел Л. Титтел Э. TCP/IP: учебный курс, Пер. с англ. — СПб: БХВ-Петербург, 2003. — С. 976.
6. Tanenbaum. Andrew S. . Компьютерные сети, 4-е издание. — М. : Питер, 2003. — С. 994.
7. А. Лейкин. Протоколы транспортного уровня UDP, TCP и SCTP: достоинства и недостатки [Электронный ресурс]. Режим доступа: https://www.researchgate.net/publication/336748145_Protokoly_transportnogo_urovna_UDP_TCP_i_SCTP_dostoinstva_i_nedostatki (Дата обращения: 05.11.2023).
8. RFC 1889. RTP: A Transport Protocol for Real-Time Applications [Элек-

- тронный ресурс]. Режим доступа: <https://www.ietf.org/rfc/rfc1889.txt> (Дата обращения: 06.11.2023).
9. Беднаж В.А. Огурцова А.М. Основы протоколы передачи потокового видео [Электронный ресурс]. Режим доступа: <https://vdocuments.mx/-5750a9aa1a28abcf0cd1ffec.html?page=1> (Дата обращения: 05.11.2023).
 10. K. Tommi. Protocol overview: RTP and RTCP [Электронный ресурс]. Режим доступа: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.842&rep=rep1&type=pdf> (Дата обращения: 06.11.2023).
 11. Гольдштейн Б. С. Гойхман В. Ю. Столповская Ю. В. Протоколы IP-телефонии: RTP, RTCP : учебное пособие. — СПб: СПбГУТ, 2014. — С. 59.
 12. K. Tommi. RTP, RTCP and RTSP — Internet Protocols for RealTime Multimedia Communication [Электронный ресурс]. Режим доступа: <https://www.cse.wustl.edu/~jain/books/ftp/rtp.pdf> (Дата обращения: 06.11.2023).
 13. RFC 2326. Real Time Streaming Protocol (RTSP) [Электронный ресурс]. Режим доступа: <https://2rfc.net/2326> (Дата обращения: 13.11.2023).
 14. RFC 2069. An Extension to HTTP : Digest Access Authentication [Электронный ресурс]. Режим доступа: <https://www.ietf.org/rfc/rfc2326.txt> (Дата обращения: 13.11.2023).
 15. Adobe RTMP Specification [Электронный ресурс]. Режим доступа: <https://rtmp.veriskope.com/docs/spec/> (Дата обращения: 13.11.2023).
 16. HTTP Live Streaming [Электронный ресурс]. Режим доступа:

<https://developer.apple.com/documentation/http-live-streaming?language=data> (Дата обращения: 14.11.2023).

17. MPEG-DASH ISO/IEC 23009-1:2012 [Электронный ресурс]. Режим доступа: https://ossrs.io/lts/zh-cn/assets/files/ISO_IEC_23009-1-DASH-2012-0e19daf31a9902aed8755e868cc39113.pdf (Дата обращения: 03.12.2023).
18. MPEG-DASH ISO/IEC 23009-1:2019 [Электронный ресурс]. Режим доступа: <https://www.iso.org/obp/ui/#iso:std:iso-iec:23009-1:ed-4:v1:en> (Дата обращения: 03.12.2023).
19. RFC 6824. TCP Extensions for Multipath Operation with Multiple Addresses [Электронный ресурс]. Режим доступа: <https://datatracker.ietf.org/doc/html/rfc6824#section-3> (Дата обращения: 04.12.2023).
20. RFC 8684. TCP Extensions for Multipath Operation with Multiple Addresses [Электронный ресурс]. Режим доступа: <https://datatracker.ietf.org/doc/html/rfc6824#section-3> (Дата обращения: 04.12.2023).

ПРИЛОЖЕНИЕ А

Презентация к данной работе

Презентация содержит 17 слайдов.