

## РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА АУДИО-ВИДЕО КОНФЕРЕНЦИЯ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ GOLANG

*Гридюшко Б.О., студент*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Шамына А.Ю. – старший преподаватель*

Современный этап развития информационных технологий характеризуется глубокой интеграцией систем видеоконференцсвязи (ВКС) в повседневные бизнес-процессы и образовательную деятельность. Удаленная работа и распределенные команды стали стандартом, что предъявляет повышенные требования к качеству, задержке и стабильности мультимедийного взаимодействия в реальном времени. В данной работе рассматриваются аспекты проектирования и реализации системы ВКС, использующей инновационные транспортные механизмы для оптимизации передачи потоковых данных.

Актуальность исследования обусловлена архитектурными и эксплуатационными ограничениями классического стека технологий WebRTC. Несмотря на свою повсеместную поддержку в браузерах, WebRTC, ориентированный изначально на одноранговые соединения (Peer-to-Peer), создает значительную инфраструктурную нагрузку при построении крупномасштабных кластерных решений. Необходимость терминации множества сессий DTLS и SCTP, а также сложность механизмов обхода NAT (STUN/TURN) на стороне сервера приводят к избыточному потреблению ресурсов CPU и ограничивают возможности горизонтального масштабирования. Альтернативным и более эффективным решением выступает протокол WebTransport, функционирующий поверх транспортного протокола QUIC (RFC 9000) [1]. Технология QUIC объединяет транспортный и защищенный уровни в единое рукопожатие (TLS 1.3), а также решает фундаментальную проблему протокола TCP – блокировку начала очереди (Head-of-Line Blocking) за счет мультиплексирования независимых потоков данных в рамках одной сессии. Применение WebTransport позволяет реализовать передачу медиаданных через ненадежные датаграммы, что минимизирует задержки в условиях сетевого джиттера и потери пакетов, неизбежных в беспроводных и мобильных сетях [2].

В рамках реализации транспортного уровня системы используется разделение данных на два типа каналов WebTransport: Streams (потоки) и Datagrams (датаграммы). Надежные двунаправленные потоки применяются для передачи управляющих сигналов, сообщений чата и метаданных конференции, где важна гарантированная доставка. В свою очередь, механизм датаграмм задействован для трансляции аудио- и видеофреймов. Поскольку датаграммы в QUIC не требуют подтверждения получения и повторной отправки при потере, это позволяет избежать «замирания» видео при кратковременных сетевых сбоях, обеспечивая минимально возможный RTT.

Одной из ключевых инноваций системы является использование архитектуры Selective Forwarding Unit (SFU) в сочетании с низкоуровневым кодированием на стороне клиента. В отличие от традиционных MCU-систем (Multipoint Control Unit), которые выполняют декодирование, микширование и повторное кодирование всех входящих потоков в один результирующий поток, SFU-сервер выполняет роль интеллектуального маршрутизатора пакетов. Это позволяет радикально снизить задержку обработки данных на стороне сервера и обеспечить возможность индивидуальной адаптации битрейта для каждого получателя без дополнительных вычислительных затрат. Использование языка Go для реализации SFU-движка позволяет эффективно управлять памятью и сетевыми буферами, минимизируя влияние циклов сборки мусора на стабильность потока.

Проектируемая система ВКС представляет собой распределенный программный комплекс. Архитектурное решение базируется на строгом разделении управляющей логики и контура высоконагруженной обработки трафика. Взаимодействие компонентов организовано через микросервисную топологию, включающую следующие функциональные узлы:

1 REST API Server (Golang): реализует бизнес-логику управления виртуальными комнатами, авторизацию и координацию доступа. Выбор языка программирования Go обоснован его эффективностью в задачах конкурентной обработки сетевых запросов благодаря легковесным потокам и оптимизированному планировщику задач. Сервер отвечает за создание уникальных идентификаторов конференций и генерацию асимметрично подписанных токенов доступа.

2 Media Server Cluster: группа независимых узлов, работающих по принципу Selective Forwarding Unit (SFU). Основной задачей узла является интеллектуальная маршрутизация бинарных пакетов между участниками сессии. Реализация SFU-движка на языке Go позволяет достичь максимальной пропускной способности при минимальном времени обработки пакета, так как сервер выполняет лишь селективную пересылку данных без ресурсоемкого транскодирования видеопотока. Применение SFU-архитектуры позволяет серверу динамически адаптировать количество пересылаемых потоков в зависимости от пропускной способности канала конкретного клиента.

3 Распределенная шина данных NATS: используется в качестве высокопроизводительного транспорта для межсерверной синхронизации состояний. Использование модели «Издатель-

Подписчик» (Pub/Sub) позволяет реализовать паттерн распределенной маршрутизации трафика. Когда участник отправляет медиапакет на локальный узел, сервер инкапсулирует его и публикует в соответствующий топик NATS. Остальные узлы кластера, обслуживающие других участников этой же комнаты, мгновенно получают пакет и пересылают его конечным клиентам. Каждый медиа-сервер кластера подписывается на топики активных комнат, что обеспечивает бесшовное взаимодействие пользователей, даже если они физически подключены к разным вычислительным узлам в различных регионах.

Для обеспечения персистентности данных в системе используется СУБД PostgreSQL, отвечающая за хранение метаданных пользователей, настроек виртуальных комнат и прав доступа. Взаимодействие с базой данных оптимизировано для обеспечения минимального времени отклика при авторизации. Помимо оперативного обмена данными, в системе реализован узел хранения, работающий с S3-совместимыми объектными хранилищами. После завершения конференции архивные файлы видеозаписей в формате MP4 автоматически выгружаются в хранилище, а соответствующие метаданные (длительность, список участников, ссылка на файл) становятся доступными пользователям через REST API, что позволяет организовать полноценный доступ к архивам встреч.

Особое внимание в работе уделено механизмам информационной безопасности. В отличие от традиционного использования JWT-токенов, подверженных рискам атак типа algorithm agility, в системе внедрен современный стандарт PASETO версии v4.public [3]. Данный подход позволил реализовать полноценную концепцию Stateless-авторизации: каждый медиа-сервер кластера проводит автономную криптографическую верификацию подписи токена с использованием открытого ключа (алгоритм Ed25519). Это исключает необходимость выполнения gRPC-запросов к центральной базе данных или серверу авторизации при установлении каждого нового соединения, что снижает задержку входа в комнату и общую нагрузку на сетевой контур.

На стороне клиентского веб-приложения применяется технология WebCodecs API, предоставляющая прямой низкоуровневый доступ к аппаратным видекодекам абонентского устройства (H.264, VP9, AV1). Процесс кодирования и декодирования кадров выносится в отдельные фоновые потоки, что позволяет минимизировать нагрузку на основной поток интерфейса и обеспечить плавность воспроизведения видео в разрешении Full HD и выше. Для сериализации данных используется формат Protocol Buffers (Protobuf), который обеспечивает минимальный размер бинарных заголовков по сравнению с текстовыми форматами (JSON, XML), что критически важно для экономии трафика в WebTransport датаграммах. Строгая типизация Protobuf в связке с языком Go исключает ошибки интерпретации структуры пакета на разных концах соединения, а скорость десериализации бинарного потока происходит быстрее парсинга JSON-строк.

Для повышения надежности системы в условиях нестабильного сетевого окружения разработан алгоритм ускоренного переподключения, использующий возможности 0-RTT протокола QUIC. В случае кратковременного обрыва связи или переключения между сетями, клиентское приложение восстанавливает сессию связи практически мгновенно, без повторного прохождения полной процедуры аутентификации и согласования параметров шифрования.

Перспективы дальнейшего развития системы связаны с внедрением алгоритмов глубокого обучения для интеллектуального подавления эха и фоновых шумов непосредственно в браузере клиента, а также исследованием возможности использования кодека AV1 для еще более эффективного сжатия данных без потери качества визуального восприятия.

Таким образом, разработка программного средства ВКС на базе синергии технологий Go, WebTransport и NATS является перспективным направлением. Предложенные архитектурные решения позволяют создавать отечественные коммуникационные платформы, способные эффективно функционировать в условиях экстремально высоких нагрузок, обеспечивая при этом высокий уровень безопасности и качества обслуживания пользователей.

**Список использованных источников:**

1. RFC 9000. QUIC: A UDP-Based Multiplexed and Secure Transport. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://datatracker.ietf.org/doc/html/rfc9000>.
2. WebTransport Over HTTP/3. W3C Working Draft. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.w3.org/TR/webtransport/>.
3. PASETO (Platform-Agnostic SEcurity TOkens) Specification. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://paseto.io/>.