

ОБЗОР (White Paper)

RTMP

vs.

SRT

Сравнение задержки по времени и максимальной скорости передачи

Haivision
VIDEO AT WORK



СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
СРАВНЕНИЕ ЗАДЕРЖКИ ПО ВРЕМЕНИ ПРОТОКОЛОВ RTMP И SRT	3
ТЕСТОВАЯ СИСТЕМА	4
Источник	4
Экраны	5
Видео энкодер	5
Видео декодер	6
Сервер Wowza	6
Сервер Haivision Media Gateway	6
Интернет соединение	6
Временные метки, кэш и буферы	6
РЕЗУЛЬТАТЫ ТЕСТА НА ЗАДЕРЖКУ ПО ВРЕМЕНИ	9
Результаты теста на задержку по времени в детальном рассмотрении	10
Сравнение аппаратного и программного кодирования	11
СРАВНЕНИЕ МАКСИМАЛЬНОЙ СКОРОСТИ ПЕРЕДАЧИ ПОТОКОВ С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛОВ RTMP И SRT	13
Тестовая система для определения максимальной скорости потоков	13
Результаты тестов на определение максимальной скорости потоков	14
ЗАКЛЮЧЕНИЕ	14



Протоколы RTMP и SRT: сравнение задержки по времени и максимальной скорости передачи

ВВЕДЕНИЕ

Для тех, кто хочет организовать трансляцию видео в режиме LIVE через публичный интернет с малой временной задержкой, существует весьма ограниченно число интернет-протоколов, из которых можно сделать выбор. Это особенно актуально, если использовать публичный интернет в качестве линии передачи данных, так как с его использованием возникают ряд серьезных трудностей, которые необходимо преодолеть, например - потеря пакетов данных или джиттер. В этом обзоре мы проведем сравнение и оценку двух наиболее популярных протоколов передачи данных через интернет - это RTMP и SRT протоколы.

Протокол потоковой передачи данных RTMP (Real-Time Messaging Protocol) - это хорошо продуманный и зарекомендовавший себя на практике протокол потоковой передачи данных с устоявшейся репутацией (особенно в области надежности), благодаря возможности ретрансляции пакетов данных на основе протокола TCP (Transmission Control Protocol) и настраиваемым буфером данных.

Протокол потоковой передачи данных SRT (Secure Reliable Transport Protocol) - это протокол с открытым исходным кодом, впервые разработанным компанией Haivision, который использует интеллектуальный механизм повторной передачи пакетов данных на основе UDP-протокола (User Datagram Protocol) совместно с AES-256 шифрованием.

В этом обзоре мы проведем исследование, как протокол SRT функционирует в реальных условиях в публичном интернете по сравнению с RTMP протоколом. Мы рассмотрим следующие вопросы: какой размер буфера требуется, какая задержка возникает при передаче данных и есть ли предел пропускной способности сети, которая вам может потребоваться. Мы также ответим на вопрос - как далеко можно транслировать видео в масштабах всего земного шара без искажения изображения и звука.

СРАВНЕНИЕ ЗАДЕРЖКИ ПО ВРЕМЕНИ ПРИ ПЕРЕДАЧЕ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛОВ RTMP И SRT

Сперва давайте рассмотрим сквозную задержку по времени, которая возникает при передаче данных от источника к приемнику видео. Сквозная задержка - это общее количество времени, которое занимает прохождение одного кадра видео от видеокамеры к экрану. Существует множество факторов, которые влияют на задержку при передаче данных, которые зависят от пути прохождения данных и количества этапов обработки видео. Хотя по отдельности эти задержки могут быть минимальными, в совокупности они играют значительную роль. Эти задержки включают в себя: задержку при кодировании, декодировании, линию передачи данных (интернет, спутник, радио, оптоволокно и т.д.), источники видео (камеры, видеоплееры) и устройства отображения - дисплеи и экраны.

Чтобы узнать больше о задержке которая возникает при передаче данных, что ее вызывает и почему ее необходимо учитывать, почитайте наш [пост в блоге по основам кодирования видео](#)

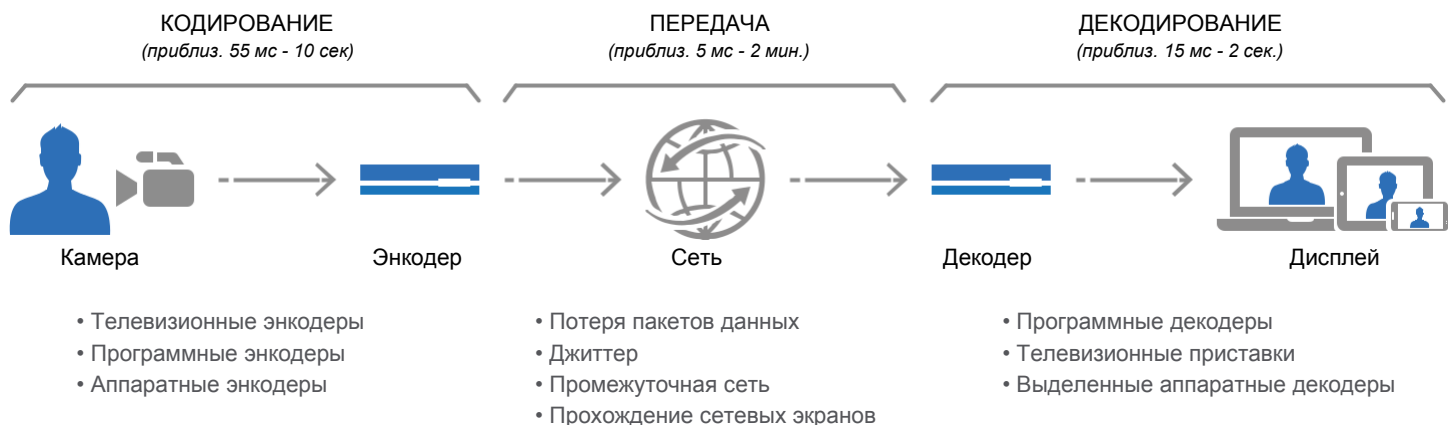


Рисунок 1: Составные части из которых складывается сквозная задержка при передаче данных



Основной целью данного обзора является исследование влияния обоих протоколов - RTMP и SRT на величину сквозной задержки при передаче данных. Для достижения сопоставимых и объективных результатов в исследовании были использованы одни и те же устройства с одинаковыми настройками, единственное что менялось - это переключение между протоколами RTMP и SRT. В некоторых случаях, однако, конфигурацию пришлось изменять с целью обеспечения стабильного потока RTMP- данных, более подробно об этом будет объяснено далее.

ТЕСТОВАЯ СИСТЕМА

На следующем рисунке показана тестовая система, по которой проводилось исследование.

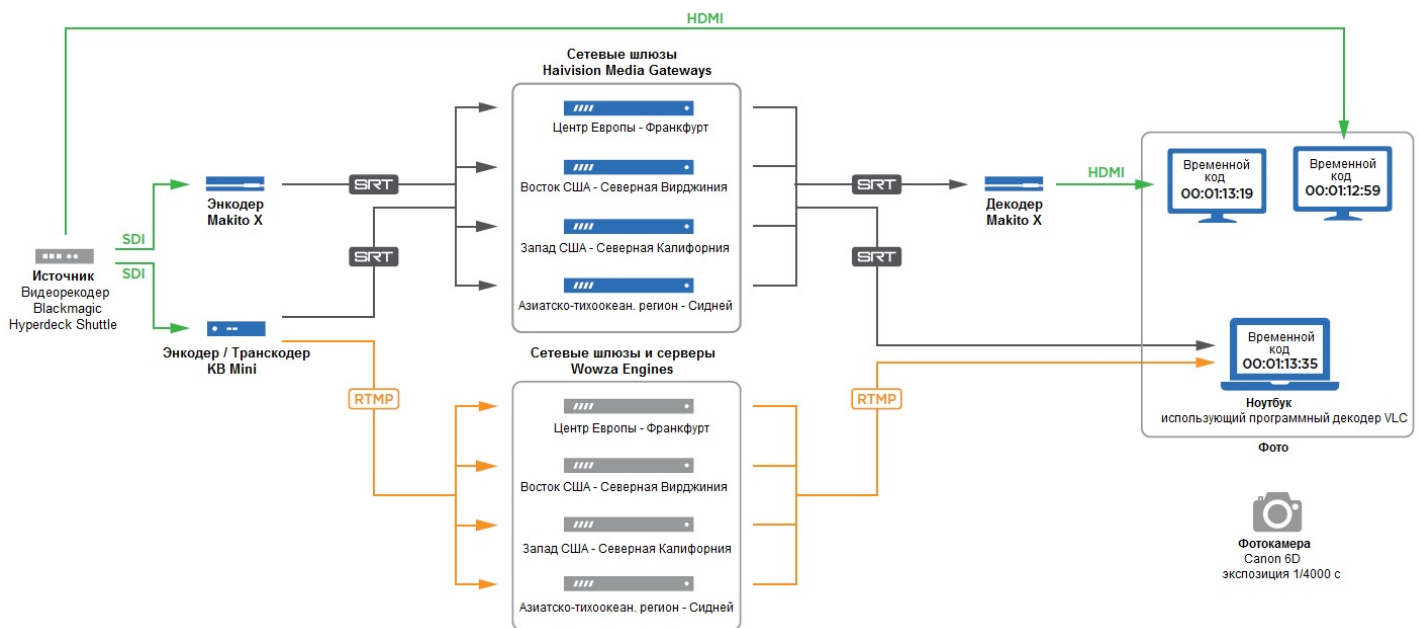


Рисунок 2 - Обзор тестовой системы (все компоненты)

Тестовая система специально была разработана в простом исполнении, чтобы эксперимент легко можно было повторить без специального дополнительного тестового оборудования. Неподвижная камера использовалась для захвата изображения двух рядом стоящих экранов, отображающих фотографию видеокadra и тайм-кода. Один из экранов был подключен напрямую к источнику, а другой - к выходу декодера, так чтобы можно было четко видеть сквозную задержку прохождения данных в направлении туда и обратно.

Для того чтобы получить точный результат с помощью тестовой системы, источник и получатель данных должны находиться в одном и том же месте. Важно отметить, что для целей теста мы учитывали путь прохождения данных туда и обратно при определении сквозной задержки. Который включает в себя: кодирование видео сигнала, время необходимое потоку данных чтобы дойти до определенного места назначения и вернуться обратно к месту отправки, декодирование видео сигнала и, наконец, задержка отображения на дисплее, буферизация данных в задействованных серверах, программных проигрывателях или аппаратных декодерах.

Источник

В качестве источника видеосигнала использовался видеорегиcтpатор Blackmagic Hyperdeck Shuttle, непосредственно передающий данные на первый экран, при этом параллельно использовались энкодеры Haivision KV и Makito X. Видеосигнал передавался со встроенным тайм кодом, позволяющим идентифицировать каждый отдельный видеокadp. Видео воспроизводилось в разрешении 720p с 60 кадрами в секунду. Каждый отдельный кадр видео отображался в течение 16,67 миллисекунд.



Экраны

Чтобы была возможность получить фотографию, на которой непосредственно отображалось время, необходимое для кодирования, потоковой передачи и декодирования видео, использовалось несколько экранов. Чтобы быть уверенными, что экраны не отличаются друг от друга по временной задержке, все дисплеи были подключены к источнику видео параллельно, после чего была сделана пробная фотография. Все экраны показали одинаковый кадр в один и тот же момент времени. Этот тест был повторен с обоими экранами, подключенными в этот раз к ноутбуку, чтобы убедиться, что дисплей ноутбука не является более медленным или быстрым по временной задержке, чем используемые экраны.



Рисунок 3: Рядом стоящие экраны отображают видео напрямую с камеры и выходное видео с декодера (цифры на рисунке - временной код / задержка по времени)

Видео энкодер

Кодирование исходного видеосигнала осуществлялось с помощью энкодера Haivision KB (версия программного обеспечения 5.4), который способен генерировать RTMP и SRT- потоки видеоданных параллельно. Оба исходящих потока видеоданных имели одинаковые характеристики аудио и видео потоков. Различие в уровне битрейта были обусловлены индивидуальными издержками протоколов RTMP и SRT при формировании потока данных.

В ходе тестирования были использованы следующие настройки энкодера:

Разрешение	1280 x 720 пикселей
Частота кадров	60 кадров в секунду
Битрейт (видео)	2136 кбит/с
Видео кодек	H.264
GOP (фикс.)	2000 мс
Видео энтропия	CAVLC
Битрейт (аудио)	192 кбит/с

Таблица 1: Настройки энкодера

Общее значение битрейта исходящего потока составляло примерно 3 Мбит/с, включая издержки протоколов.

Дополнительно также использовались энкодер и декодер серии Makito X (версия программного обеспечения 5.4). Это устройства с аппаратным ускорением, которые обеспечивают значительно меньшую задержку сигнала по сравнению с программно реализованными энкодерами и декодерами. При этом, в энкодере и декодере Makito X использовались те же настройки, что и в программном энкодере серии KB.



Видео декодер

Поскольку возможность простого повторения эксперимента была ключевым фактором, в качестве программного декодера использовался видеоплеер VLC Player, один из наиболее распространенных видеоплееров как в корпоративном, так и в домашнем использовании. Чрезвычайно популярный и универсальный видеоплеер VLC Player декодирует множество форматов видео, включая RTMP и SRT потоки данных.

В нашем тесте использовался VLC Player со стандартными настройками. Стандартный уровень кэша в 250 миллисекунд работал в большинстве тестов, но в некоторых случаях его приходилось менять. Более подробно об этом можно прочитать в разделе: [Временные метки, кэш и буферы](#).

Сервер Wowza

В качестве конечного места назначения для потоков RTMP использовался сервер Wowza Streaming Engine (Версия 4.7.5 - сборка 21763). Сервер Wowza Streaming Engine был размещен на объектах AWS, расположенных в тех же центрах обработки данных AWS, что и сетевые шлюзы Haivision Media Gateways, которые выступали в качестве конечного места назначения для потоков SRT. Декодер (VLC) принимал RTMP поток данных обратно в исходном месте из этого конкретного сервера.

Wowza Streaming Engine также поддерживает SRT протокол, но на момент тестирования в нем использовалась очень старая версия протокола, которая не обеспечивает весь потенциал текущей версии SRT протокола. Кроме того, во время тестирования в сервисе Streaming Cloud от Wowza не было реализации SRT протокола.

Время передачи данных туда и обратно Round Trip Time (RTT) до сервера Wowza и до шлюза Haivision Media Gateway, размещенного в одном и том же центре обработки данных AWS, никогда не отличалось более чем на 2 миллисекунды. В большинстве тестов общее время передачи данных RTT был одинаковым.

Сервер Haivision Media Gateway

Подобно RTMP потокам, в качестве конечной точки для SRT потоков использовался сетевой шлюз Haivision Media Gateway (версия 3.0.1), размещенный на объектах AWS.

В рамках системы AWS, сетевой шлюз Haivision Media Gateway доступен как программное обеспечение, а также как дополнительная услуга и может быть запущен в дата-центрах AWS по вашему выбору.

Вход и выход для SRT потоков были сконфигурированы как SRT порты с определенным буфером задержки. Размер этого буфера зависит от времени передачи данных RTT до места назначения. Более подробную информацию об этом можно найти в разделе [Временные метки, Кэш и Буферы](#).

Интернет соединение

Интернет соединение было установлено с помощью DSL-модема Fritzbox 7590 со скоростью 100 Мбит/с для входящих данных и 42 Мбит/с - для исходящих. Параметры соединения постоянно отслеживались, чтобы гарантировать отсутствие влияния других приложений на перегрузку линии связи.

Временные метки, кэш и буферы

Основное различие между RTMP и SRT протоколами заключается в отсутствии временных меток в заголовках пакетов данных в RTMP протоколе. RTMP протокол содержит только временные метки фактического потока данных в соответствии с его частотой кадров. Отдельные пакеты не содержат эту информацию, поэтому получатель RTMP данных должен отправлять каждый полученный пакет в течение фиксированного интервала времени в процессе декодирования. Чтобы сгладить разницу во времени необходимо для перемещения отдельных пакетов данных, требуются большие объемы буферов данных.

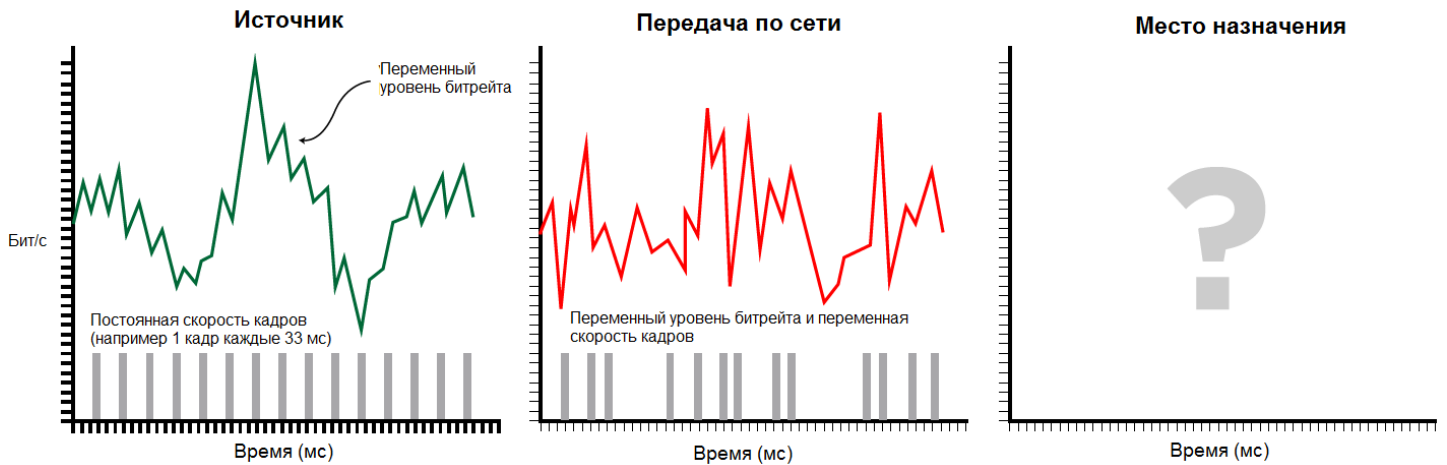


Рисунок 4: Характеристики входного сигнала не соответствуют характеристикам передачи по сети

SRT протокол, с другой стороны, включает временные метки для каждого отдельного пакета данных. Это позволяет воссоздать характеристики сигнала на стороне приемника и значительно снижает потребность в буферизации. Другими словами, поток битов данных, покидающий приемник выглядит точно так же как поток, поступающий в SRT-отправитель.

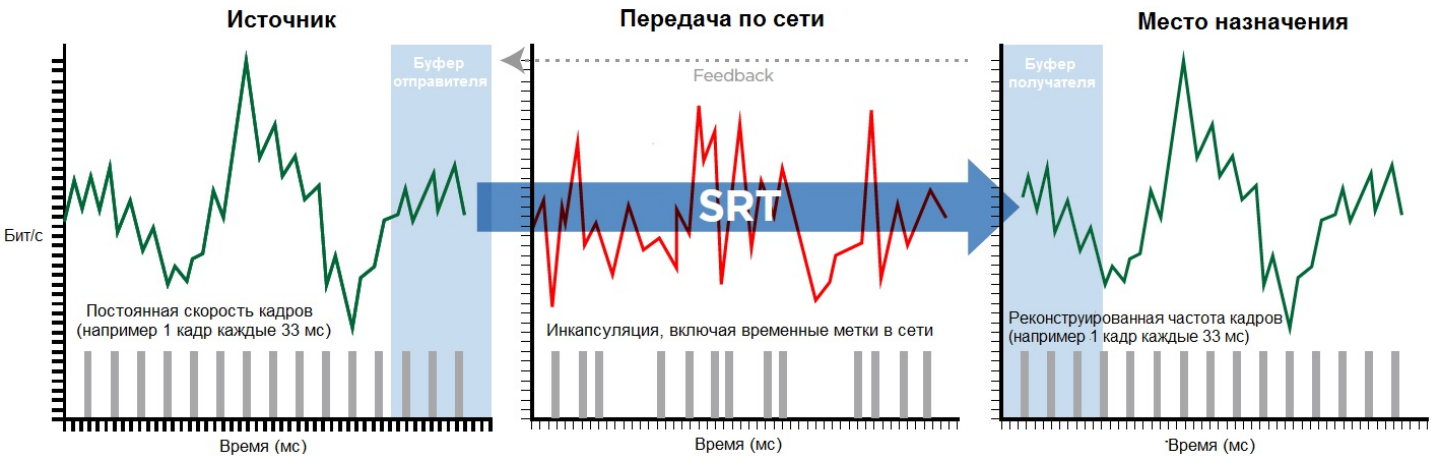


Рисунок 5: Восстановление характеристик сигнала на стороне приемника SRT потока

Другим существенным отличием между RTMP и SRT протоколами является реализация повторной передачи пакетов данных. RTMP протокол основан на TCP протоколе, который основан на подтверждениях о приеме данных, отправленных получателем. Однако эти подтверждения (ACK) не уходят отправителю немедленно, чтобы экономить обратный трафик и поддерживать его на низком уровне. Только после того, как определенная последовательность пакетов данных будет получена, отчет о подтверждениях (ACK) или отрицательных подтверждениях (NACK) будет отправлен обратно. Если в этой последовательности будут потерянные пакеты данных, тогда полная последовательность пакетов данных (до последнего ACK) будет передана повторно.



С другой стороны, SRT протокол может идентифицировать отдельный потерянный пакет данных по его порядковому номеру. Если разница порядковых номеров больше чем один пакет, запускается повторная передача этого пакета. В данном случае только этот конкретный пакет данных отправляется снова, чтобы задержка по времени и издержки были минимальными.

Согласно форуму [Wowza Forums](#) размер буфера рассчитывается следующим образом:

«Большой объем буфера отправки и буфера приема обеспечит лучшую пропускную способность в соединениях, где наблюдается большая задержка по времени (с высоким временем пинга). Уровень возможной пропускной способности соединения в байтах/секунду должен быть меньше отношения размера буфера ко времени прохождения данных туда и обратно (buffer size/RTT). Размер буфера определяет какое количество данных может быть отправлено до того момента, когда отправляющая сторона должна остановиться и дождаться ответа от принимающей стороны. Размер, используемый для соединения будет меньше размера буфера отправки у отправителя или буфера приема у получателя. Как правило, большинство современных клиентов используют автоматические настройки, поэтому буфер отправки на сервере Wowza может ограничивать скорость соединения для соединений с высоким уровнем задержки. По умолчанию Wowza использует значение в 65000 байтов для буфера отправки, которое является оптимальным средним значением. Это обеспечит следующие показатели производительности:

$65000 \text{ байт} / 50 \text{ мс (RTT)} = 1300000 \text{ байт/сек} = 10.4 \text{ Мбит/с}$

$65000 / 100 \text{ мс} = 5.2 \text{ Мбит/с}$

$65000 / 200 \text{ мс} = 2.6 \text{ Мбит/с}$

Как видите, чем дальше находится клиентский видеоплеер, тем меньшая пропускная способность передачи потока видеоданных будет у вашего сервера. Если ваш видеоплеер расположен не очень далеко, настройки по умолчанию должны хорошо подойти как для потока данных, так и для видеоплеера, которые вы описали. Но вы можете серьезно ограничить пропускную способность, если установите ей слишком низкое значение. Настройки с низкой задержкой предназначены для тех случаев, когда видеоплеер находится очень близко к серверу или уровень битрейта у потока данных очень низкий.

$16000 \text{ байт} / 50 \text{ мс (RTT)} = 320000 \text{ байт/сек} = 2.56 \text{ Мбит/с}$

$16000 / 100 \text{ мс} = 1.28 \text{ Мбит/с}$

$16000 / 200 \text{ мс} = 640 \text{ Кбит/с}$

Размеры принимающих буферов на сервере Wowza обычно влияют только на потоковую передачу видео в режиме LIVE, поэтому их следует настроить таким образом, чтобы ваши энкодеры имели достаточный запас по пропускной способности. Если вы установите слишком большие размеры буферов, вы можете столкнуться с проблемами, когда возникнет перегрузка, поскольку серверу потребуется больше времени, чтобы обнаружить и внести коррективы для преодоления возникшей ситуации. По этой причине вам не следует устанавливать эти значения слишком большими. Если вы приблизительно знаете, как далеко находятся ваши клиентские видеоплееры и источник видео, вам будет лучше использовать ручные настройки ».

В заключение: если вы не знаете, как далеко вы находитесь от точки приема (с точки зрения времени прохождения данных туда и обратно RTT), и вы не уверены в характеристиках вашего соединения - используйте SRT протокол.

В проведенных тестах там где это было возможно использовался размер буфера данных по умолчанию - 65 000 байт. Однако при отправке потока данных в Австралию, где время прохождения данных туда и обратно составляет 360 мс, размер буфера был увеличен до 260 000 байт с целью достижения стабильности передаваемого потока данных.



РЕЗУЛЬТАТЫ ТЕСТА НА ЗАДЕРЖКУ ПО ВРЕМЕНИ

Протоколы RTMP и SRT сравнение задержки по времени и времени прохождения данных RTT

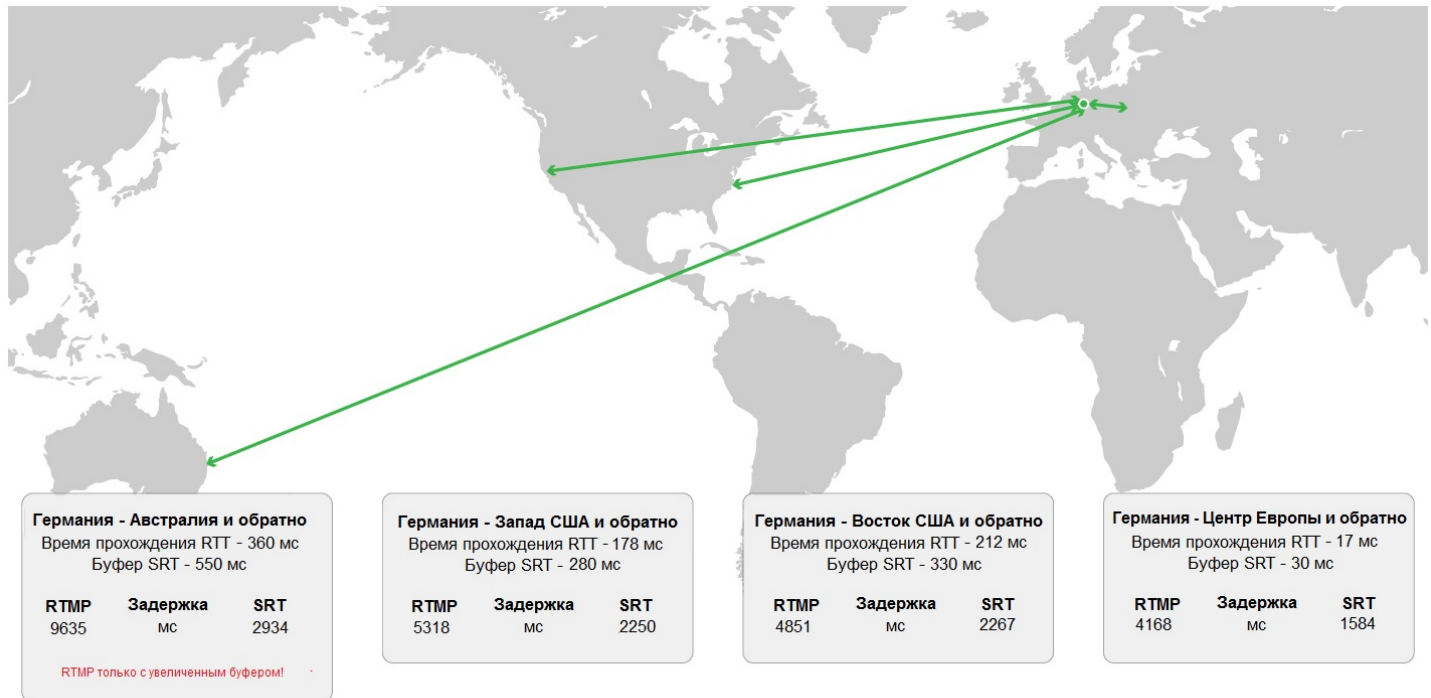


Рисунок 6: Результаты теста на определение задержки по времени при передаче данных туда и обратно

Как и ожидалось, чем больше расстояние проходят данные до места назначения, тем больше величина сквозной задержки. Важно отметить, что эти числа являются абсолютными значениями, которые включают в себя задержку при кодировании, передаче данных, декодировании и в устройствах отображения. Следует отметить, что значение задержки указано при передаче данных в оба направления - туда и обратно, например из Германии в Австралию и обратно в Германию.

Путь	Время RTT	Буфер VLC Player (RTMP / SRT)	Суммарная задержка по времени	
			RTMP протокол	SRT протокол
Германия → Сидней → Германия	360 мс	2000 / 250 мс	9635 мс	2934 мс
Германия → США Калифорния → Германия	178 мс	700 / 250 мс	5318 мс	2250 мс
Германия → США Вирджиния → Германия	212 мс	250 / 250 мс	4851 мс	2267 мс
Германия → Франкфурт → Германия	17 мс	250 / 250 мс	4168 мс	1584 мс

Таблица 2: Результаты теста на определение задержки по времени



Результаты теста на задержку по времени в детальном рассмотрении

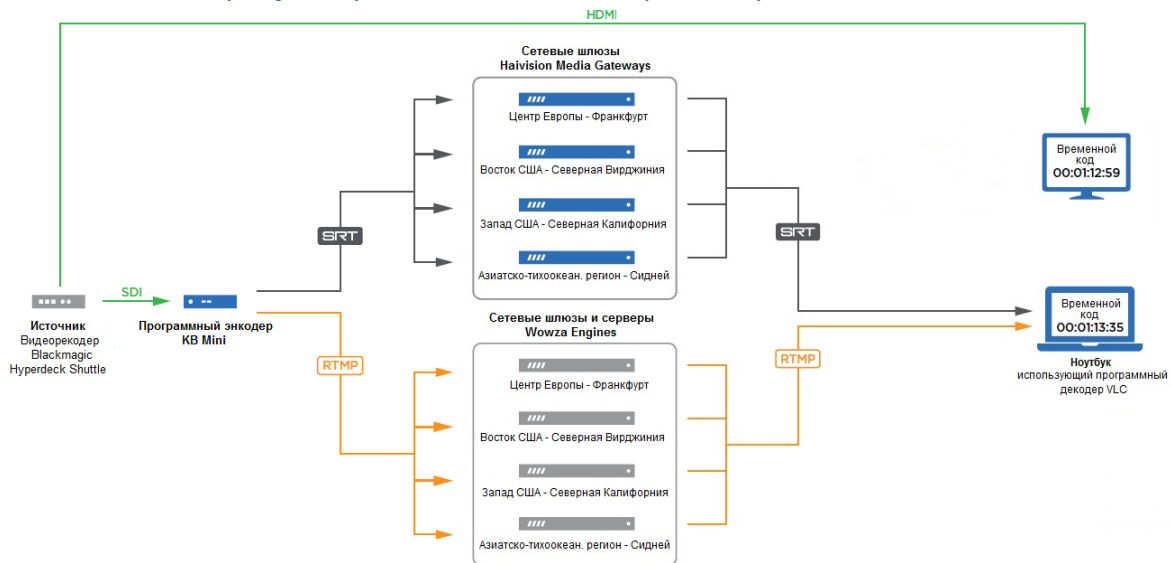


Рисунок 7: Программное кодирование / декодирование - тестовая система с программным энкодером KB и декодером VLC

Чем дальше от вас находится место назначения, тем больше времени потребуется потоку данных, чтобы туда добраться. Таким образом, неудивительно, что передача данных с одной стороны земного шара на другую, в данном случае - из Германии в Австралии, заняла больше всего времени.

Германия - Сидней - Германия

Чтобы обеспечить доставку стабильного потока видео и аудио в Сидней из Германии с помощью RTMP протокола, размер буфера приемника на сервере Wowza Streaming Engine пришлось увеличить до 260 000 байтов. Это в четыре раза больше, чем его объем по умолчанию - 65 000 байтов. Так как в основе теста лежала передача данных туда и обратно, принимающий буфер VLC Player также пришлось увеличить с 250 мс (значение по умолчанию) до 2000 мс. Ниже этих значений качество передаваемого потока значительно ухудшалось из-за аудио и видео артефактов или даже не воспроизводилось вообще.

Германия - Калифорния - Германия

Хотя время прохождения потоком данных пути в Калифорнию туда и обратно было приблизительно равно половине такого же времени при передаче данных в Австралию, не только это время влияет на итоговую величину задержку. Поток данных обратно в Германию казался испытывал сильный джиттер, что привело к отклонению доставки индивидуальных пакетов данных во времени. Хотя передача данных из Германии в Калифорнию работала без перебоев с использованием размера буфера по умолчанию - 65 000 байт, обратный путь потребовал увеличение буферизации до 650 мс в проигрывателе VLC Player.

Германия - Вирджиния - Германия

Несколько удивительно, результаты теста показали, что время передачи данных до места назначения в Вирджинии было всего немного большим (менее 3 кадров или 50 мс) чем до Калифорнии, несмотря на то, что географически на карте это явно более короткое расстояние от Германии. Отсюда можно сделать вывод, что кратчайший географический путь не всегда может быть самым быстрым. Данные перемещаются со скоростью света между дата-центрами и их маршрутизаторами.

В зависимости от использования пропускной способности каналов передачи данных, ваш видеосигнал может не всегда проходить по кратчайшему интернет-пути, но быстрее, чем по прямому географическому маршруту. Хотя время передачи данных до места назначения было незначительно выше, чем в калифорнийском тесте, связь была более стабильной с меньшим джиттером и позволяла использовать меньшие объемы буферов для RTMP потоков. В этом случае видеопоток был стабильным со стандартными значениями по умолчанию - объемом буфера в 65000 байт на сервере Wowza Streaming Engine и 250 мс для проигрывателя VLC Player.



Тюрингия - Франкфурт - Тюрингия

В тесте при отправке потока данных из Тюрингии в Германии в датацентр AWS во Франкфурте, время передачи данных до места назначения и обратно (RTT) составило всего 17 мс. Задержка по времени при передаче RTMP данных туда и обратно не сильно уменьшилась по сравнению с результатами для Вирджинии или Калифорнии. Однако протокол SRT смог выиграть более одной секунды при меньшем значении времени передачи данных до пункта назначения (RTT) по сравнению с местоположениями в США.

В этих тестах, SRT протокол был в 2,5-3,2 раза быстрее по сравнению с RTMP протоколом.

Сравнение задержки SRT и RTMP протоколов (энкодер KB)

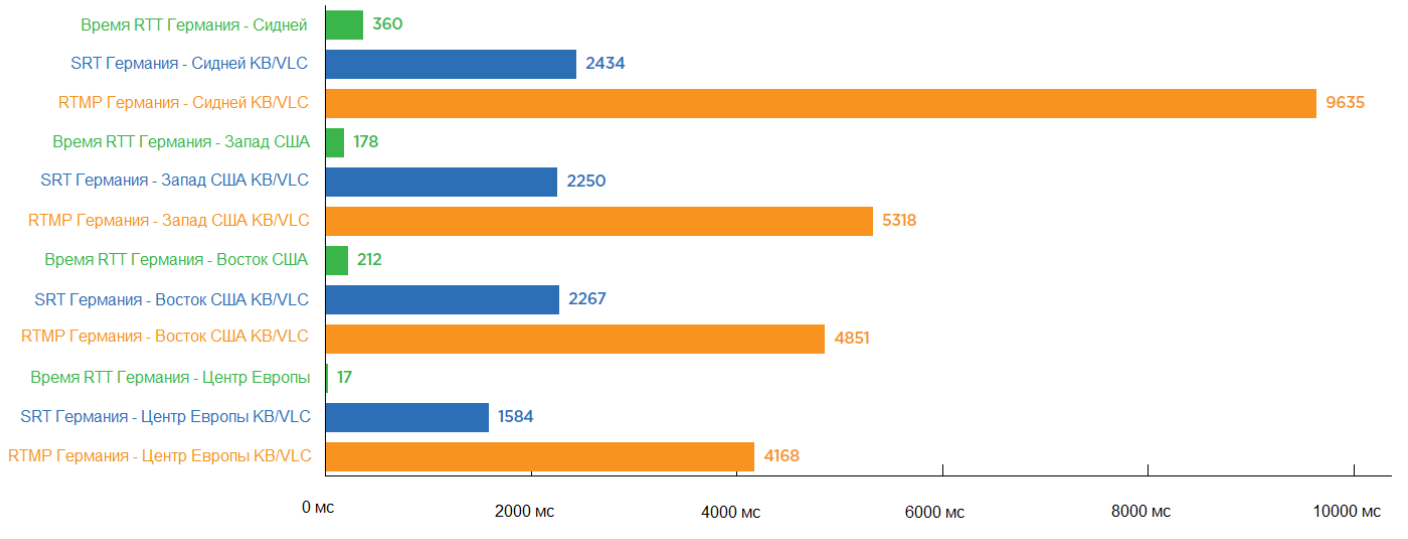


Рисунок 8: Результаты теста на определение задержки по времени при передаче данных туда и обратно с использованием программного энкодера и декодера

Сравнение аппаратного и программного кодирования



Рисунок 9: Тестовая система с использованием аппаратных энкодеров и декодеров Makito X

До сих пор мы исследовали программное кодирование и декодирование, и даже самые быстрые результаты, достигнутые с помощью SRT протокола были все еще выше 1,5 секунд. Для случаев, когда задержка по времени является критическим параметром, 1,5 секунды намного превышает желаемое пороговое значение при взаимодействии разнонаправленных потоков данных, например при организации прямых трансляций двусторонних интервью.



Задержка заметна во время прямой трансляции новостей, например когда репортер в США дает интервью европейскому телеканалу с использованием спутниковой линии связи. Любые задержки между заданным вопросом и предоставленным ответом всегда заметны. Как правило, интервьюируемые предварительно обучаются и часто знают задаваемый вопрос заранее, поэтому они начнут отвечать еще до того, как другая сторона закончит задавать вопрос. Однако с задержкой в 1,5 секунды задержка будет чрезвычайно высокой.

Хотя SRT протокол помогает значительно снизить задержку при передаче данных, аппаратные энкодеры и декодеры, такие как Haivision Makito X Series, также способны ускорить процессы кодирования и декодирования видео. В следующей таблице показаны результаты теста на определение задержки по времени при передаче данных и кодировании с помощью программного энкодера Haivision KB и видеоплеера VLC Player (передача данных по RTMP и SRT протоколам) по сравнению с аппаратным энкодером и декодером Haivision Makito X (передача данных по SRT протоколу).

Результаты с использованием энкодера и декодера Makito X с SRT протоколом показывают существенную разницу. Двусторонняя сквозная задержка в Австралию и обратно уменьшается с 9,6 секунды до 1,6 секунды при использовании выделенного аппаратного кодирования и декодирования. В Германии результат составляет всего 333 миллисекунды для SRT протокола, в то время как для RTMP протокола в программном исполнении требуется 4,1 секунды. Это в 12 раз быстрее и может рассматриваться как сверхнизкая задержка по времени.

Задержка по времени SRT и RTMP протоколов

Аппаратное кодирование и декодирование (Haivision Makito X)
в сравнении с программным кодированием и декодированием (Haivision KB Encoder/VLC)

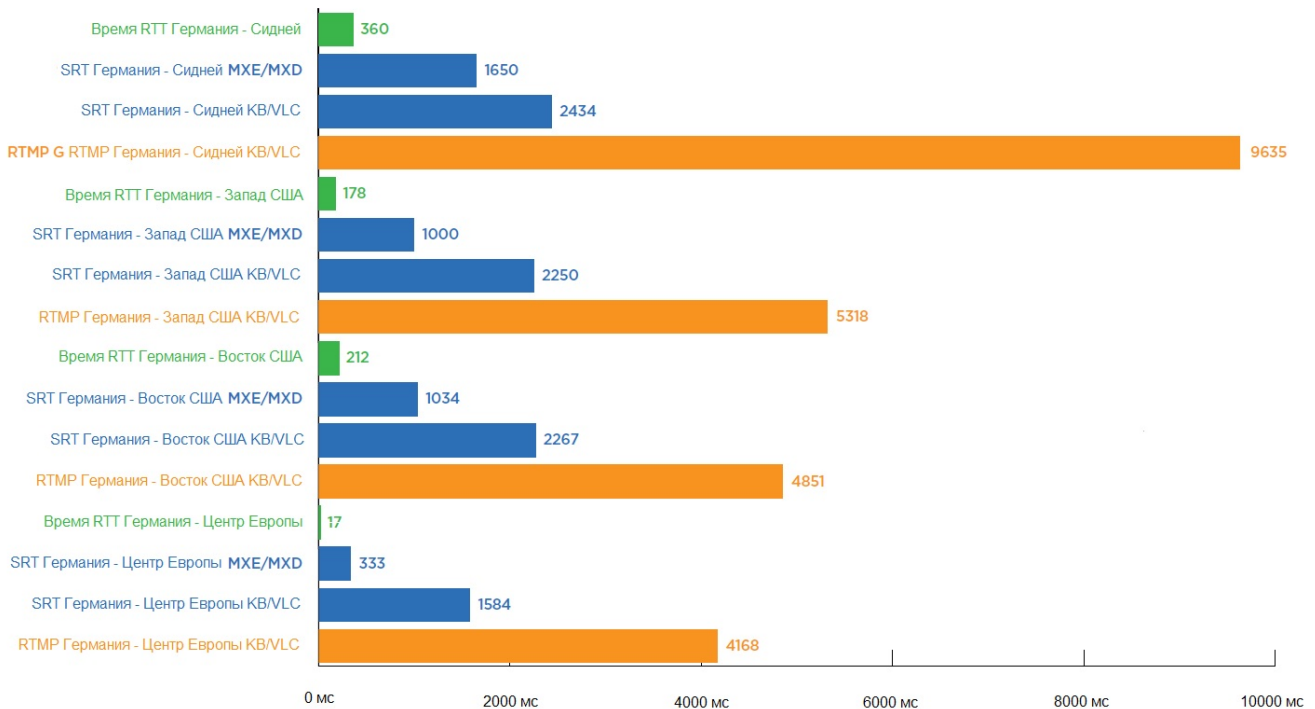


Рисунок 10: Результаты теста на определение задержки по времени при передаче данных туда и обратно с использованием аппаратного кодирования / декодирования



СРАВНЕНИЕ МАКСИМАЛЬНОЙ СКОРОСТИ ПЕРЕДАЧИ ПОТОКОВ ПО ПРОТОКОЛУ RTMP И SRT

Результаты теста показывают, что SRT протокол оказывает существенное влияние на снижение задержки по времени при передаче видео. Но как насчет его влияния на качество видео? Простой способ улучшить качество видео и аудио - просто увеличить скорость потоков видео (битрейт). Но каково значение максимальной скорости при передаче потоков на большие расстояния и как далеко вы можете передать видеопоток с высокой скоростью?

Тестовая система для определения максимальной скорости потоков

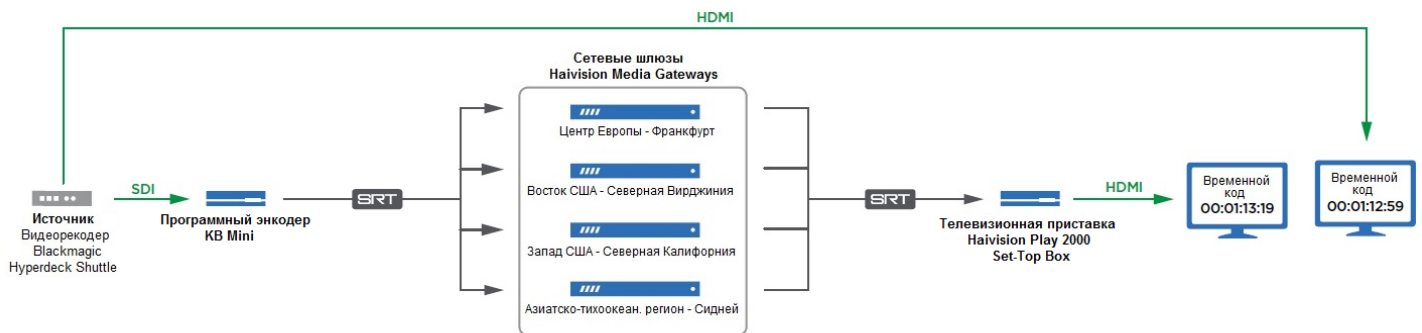


Рисунок 11: Тестовая система для определения максимальной скорости потоков

Чтобы протестировать скорость передачи потоков, мы выбрали место с отличным интернет-соединением - Microsoft Production Studios в Редмонде, штат Вашингтон. Следующий крупный интернет-узел находится всего в 2 шагах, и все подключенные устройства имели скорость соединения 1 Гбит/с с интернетом.

Учитывая, что сохранение низкого значения задержки являлось ключевым фокусом тестирования, настройки буфера оставались неизменными при увеличении скорости передачи во время тестов. Размеры буфера были протестированы при скорости 2 Мбит/с. Как только поток стал стабильным, тесты проводились с использованием скоростей 1, 2, 6, 10 и 20 Мбит/с.

Поток видео с энкодера Haivision KV 5.4 был отправлен на сетевой шлюз Haivision Media Gateway в центр обработки данных AWS в Калифорнии и Северной Вирджинии в США, во Франкфурт в Германии и в Сидней. Чтобы оценить качество видео, поток был отправлен обратно в Редмонд по SRT протоколу и воспроизведен на телевизионной приставке Haivision Play 2000 Set Top Box.

Результаты теста на определение максимальной скорости потока

Результаты теста убедительны. SRT протокол не испытывал проблем с потоковой передачей до 20 Мбит/с в любой регион мира. RTMP протокол хорошо работал когда отправитель и получатель находились на одном континенте, в данном случае в Северной Америке. Из Редмонда можно было отправлять потоки видео со скоростью до 20 Мбит/с по протоколу RTMP в Калифорнию или Вирджинию. Однако потоковая передача видео в Европу и Австралию была невозможна при скорости передачи выше 2 Мбит/с. С SRT протоколом были бы возможны даже более высокие скорости, но поскольку RTMP поток уже вышел из строя на больших расстояниях при более низком уровне битрейта, скорость в 20 Мбит/с была хорошим верхним пределом для этого теста.



RTMP И SRT - Как далеко можно передать видео через Интернет?

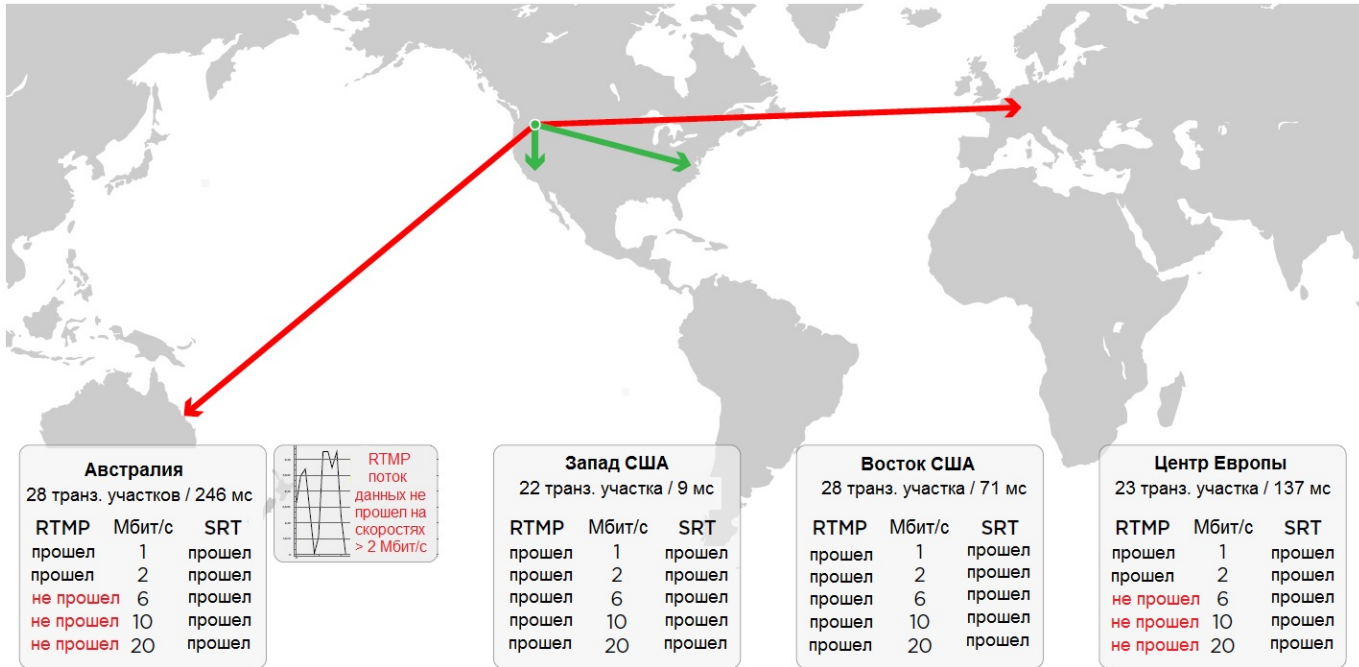


Рисунок 12: Результаты теста на скорость передачи потоков

ЗАКЛЮЧЕНИЕ

С точки зрения сквозной задержки по времени и максимальной скорости потоков (битрейта), SRT протокол намного превосходит RTMP. Хотя эти тесты могли быть проведены также и в лабораторной среде с использованием инструментов для моделирования ситуации с потерей пакетов данных и большим джиттером, мы намеренно решили выполнять их в реальных условиях с помощью публичного интернета. Реальные условия трудно воспроизвести в лаборатории, и они всегда будут приближительными.

Заглядывая в будущее, в скором времени на горизонте появится новая инновация от Haivision - SRT Hub, которая предложит альтернативу потоковой передаче видео через публичный интернет. SRT Hub, работающий на базе SRT протокола, представляет собой облачный сервис маршрутизации, который использует масштабируемость и охват глобальной сети Microsoft Azure. Это сведет к минимуму количество переходных транзитных участков при передаче потоков видео через публичный интернет, что в свою очередь приведет к еще большему снижению задержки по времени.

Если вы хотите узнать больше о решениях для потоковой передачи видео от Haivision, отмеченных наградой Emmy®, свяжитесь с нами

info@haivision.ru