Лекция 07

Протоколы RTP, RTCP, SRTP.

Протокол RTP

Протокол RTP (Real-time Transport Protocol) работает на прикладном уровне и используется при передаче трафика реального времени. Протокол был разработан Audio-Video Transport Working Group в IETF и впервые опубликован в 1996 году как **RFC 1889**, и заменён в **RFC 3550** в 2003 году.

Протокол RTP переносит в своём заголовке данные, необходимые для восстановления аудиоданных или видеоизображения в приёмном узле, а также данные о типе кодирования информации (JPEG, MPEG и т. п.). В заголовке данного протокола, в частности, передаются временная метка и номер пакета. Эти параметры позволяют при минимальных задержках определить порядок и момент декодирования каждого пакета, а также интерполировать потерянные пакеты.

RTP не имеет стандартного зарезервированного номера порта. Единственное ограничение состоит в том, что соединение проходит с использованием чётного номера, а следующий нечётный номер используется для связи по протоколу RTCP. Тот факт, что RTP использует динамически назначаемые адреса портов, создаёт ему трудности для прохождения межсетевых экранов, для обхода этой проблемы, как правило, используется STUN-сервер.

На практике протокол RTP не отделим от протокола **RTCP (Real-Time Transport Control Protocol — протокол управления передачей в реальном времени)**. Последний служит для мониторинга QoS и для передачи информации об участниках обмена в ходе сессии.

Описание протокола

RTP был разработан как протокол реального времени, из конца в конец (end-to-end), для передачи потоковых данных. В протокол заложена возможность компенсации джиттера и детектированию нарушения последовательности пакетов данных — типичных событий при передаче через IP-сети. RTP поддерживает передачу данных для нескольких адресатов через Multicast. RTP рассматривается как основной стандарт для передачи голоса и видео в IP-сетях и совместно с кодеками.

Приложения, формирующие потоки реального времени, требуют своевременной доставки информации и для достижения этой цели могут допустить некоторую потерю пакетов. Например, потеря пакета в аудио-приложении может привести к доле секунды тишины, которая может быть незаметна при использовании подходящих алгоритмов скрытия ошибок. Протокол TCP, хотя и стандартизирован для передачи RTP, как правило не используется в RTP-приложениях, так как надежность передачи в TCP формирует временные задержки. Вместо этого, большинство реализаций RTP базируется на UDP. Кроме этого, существуют другие спецификации для транспортных протоколов SCTP и DCCP, но они мало распространены.

Компоненты протокола

Протокол передачи данных, RTP, который взаимодействует с передачей данных реального времени. Информация, предоставляемая посредством этого протокола, включает в себя тайм-штамп (для синхронизации), последовательный номер (для детектирования потери и дублирования пакетов) и формат полезной нагрузки, который определяет формат кодирования данных.

Протокол контроля, RTCP, используемый для определения качества обслуживания (QOS), обратной связи и синхронизации между медиа-потоками. Занимаемая полоса пропускания RTCP мала в сравнении с RTP, обычно около 5 %.

Управляющий сигнальный протокол, такой как **SIP, H.323, MGCP или H.248**. Сигнальные протоколы управляют открытием, модификацией и закрытием RTP-сессий между устройствами и приложениями реального времени.

Управляющий протокол описания медиа, такой как **Session Description Protocol**.

Сессии

RTP-сессия устанавливается для каждого потока мультимедиа. Сессия состоит из IP-адреса и пары портов для RTP и RTCP. Например, аудио и видео потоки будут иметь различные RTP-сессии, позволяющие приемнику для этого выделить конкретный поток. Порты, которые образуют сессию, связываются друг с другом средствами других протоколов, таких как SIP. В соответствии со спецификацией, **RTP не имеет стандартного зарезервированного номера порта**. Единственное ограничение состоит в том, что **соединение проходит с использованием чётного номера, а следующий нечётный номер используется для связи по протоколу RTCP**. RTP и RTCP обычно используют непривилегированные UDP-порты (16k-32k), но могут использовать и другие протоколы, поскольку сам протокол RTP независим от транспортного уровня.

В некоторых случаях можно столкнуться с ситуацией, когда один из участников конференции подключен к сети через узкополосный канал. Было бы не слишком хорошо требовать от всех участников перехода на кодировку, соответствующую этой малой полосе. Для того чтобы этого избежать, можно установить преобразователь, называемый **смесителем**, в непосредственной близости от узкополосной области.

Смеситель преобразует поток аудио-пакетов в следовательность пакетов, которая соответствует возможностям узкополосного канала. Эти пакеты могут быть уникастными (адресованными одному получателю) или мультикастными. Заголовок RTP включает в себя средства, которые позволяют мультиплексорам идентифицировать источники, внесшие вклад. Так что получатель может правильно идентифицировать источник звукового сигнала.

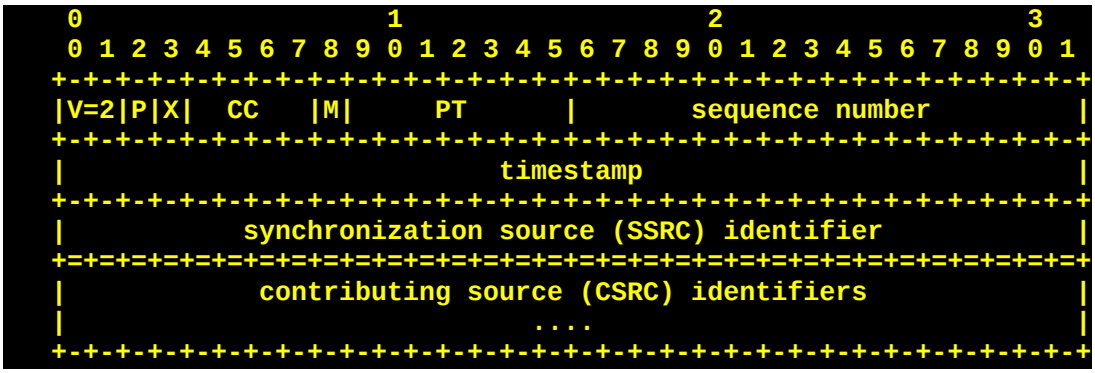
Некоторые участники конференции, использующие широкополосные каналы, не доступны для IP-мультикастинга (например, находятся за Firewall). Для таких узлов смесители не нужны, здесь используется другой RTP-уровень передачи, называемый трансляцией. Устанавливается два транслятора по одному с каждой из сторон Firewall. Внешний транслятор передает мультикастинг-пакеты по безопасному каналу внутреннему трансляторy. Внутренний же транслятор рассылает их подписчикам локальной сети обычным образом.

Транслятор. Переадресует RTP-пакеты, не изменяя их SSRC-идентификаторы. Это позволяет получателям идентифицировать отдельные источники, даже если пакеты от всех источников проходят через один общий транслятор и имеют сетевой адрес транслятора. Некоторые типы трансляторов передают данные без изменений, другие кодируют данные и соответственно изменяют коды типа данных и временные метки. Приемник не может

заметить присутствия транслятора.

Смеситель. Принимает потоки RTP-данных от одного или нескольких источников, может изменять формат данных, определенным образом объединяет потоки и затем формирует из них один общий поток. Так как объединяемые потоки не синхронизованы, смеситель производит синхронизацию потоков и формирует свою собственную временную шкалу для исходящего потока. Смеситель является источником синхронизации. Таким образом, все пакеты данных, переадресованные смесителем, будут помечены SSRC-идентификатором смесителя. Для того чтобы сохранить информацию об источниках исходных данных, смеситель должен внести свои SSRC-идентификаторы в список CSRC-идентификаторов, который следует за фиксированным RTP-заголовком пакета. Смеситель, который, кроме того, вносит в общий поток свою составляющую, должен включить свой собственный SSRC-идентификатор в CSRC-список для данного пакета.

Структура пакета



0-1 — Ver. (2 бита) указывает версию протокола. Текущая версия — 2.
2 — P (один бит) используется в случаях, когда RTP-пакет дополняется пустыми байтами на конце.
3 — X (один бит) используется для указания расширений протокола, задействованных в пакете.
4-7 — CC (4 бита) содержит количество CSRC-идентификаторов, следующих за постоянным заголовком.
8 — M (один бит) используется на уровне приложения и определяется профилем. Если это поле установлено, то данные пакета имеют какое-то особое значение для приложения.
9-15 — PT (7 бит) указывает формат полезной нагрузки и определяет её интерпретацию приложением.
Источник синхронизации (SSRC): Источник потока RTP-пакетов, определяется 32-битным числовым SSRC-идентификатором, который записывается в заголовок RTP-пакета и не зависит от сетевого адреса. Все пакеты от источника синхронизации образуют часть с идентичной временной привязкой и нумерацией. Эти данные используются принимающей стороной при воспроизведении. Источниками синхронизации могут служить источники первичного сигнала (микрофоны или видеокамеры), а также RTP-смесители. SSRC-идентификатор представляет собой случайное число, которое является уникальным для данной RTP-сессии. Участник сессии не должен использовать один и тот же SSRC-идентификатор для всех RTP-сессий мультимедийного набора. Если участник формирует несколько потоков в рамках одной RTP-сессии (например, от нескольких видеокамер), каждый участник должен быть снабжен уникальным SSRC-идентификатором.
Информационный источник CSRC (contributing source): Источник потока RTP-пакетов, который вносит вклад в общий поток, формируемый RTP-смесителем. Смеситель вставляет список SSRC-идентификаторов, которые идентифицируют парциальные источники, в заголовок RTP-пакетов. Этот список называется CSRC-списком. Примером приложения может быть аудио-конференция, где смеситель отмечает всех говорящих, чей голос порождает исходящие пакеты. Это позволяет принимающей стороне идентифицировать говорящего, хотя все пакеты имеют один и тот же SSRC-идентификатор.

Протокол RTCP

RTCP (Real-Time Transport Control Protocol — протокол управления передачей в реальном времени) — протокол, используемый совместно с RTP. Протокол описан в **RFC 3550**. RTCP базируется на периодической передаче управляющих пакетов всем участникам сессии, используя тот же механизм рассылки, что и для пакетов данных.

Протокол базируется на периодической передаче управляющих пакетов всем участникам сессии, используя тот же механизм рассылки, что и для пакетов данных. Не имеет самостоятельного значения и используется лишь совместно с RTP. Нижележащий протокол должен обеспечивать мультиплексирование пакетов данных и управления, используя разные номера портов. RTCP выполняет четыре функции:

1. Обеспечение обратной связи для контроля качества при рассылке данных. Обратная связь может быть непосредственно полезна при адаптивном кодировании, но эксперименты с IP мультикастингом показали, что для получателей крайне важно диагностировать ошибки при рассылке пакетов. Посылка сообщений-отчетов о приеме данных всем участникам позволяет тому, кто обнаружил какие-то проблемы, разобраться в том, являются ли эти трудности локальными или глобальными. При механизме рассылки типа IP-мультикастинга, сервис провайдер, который непосредственно не вовлечен в сессию, получив обратную связь, может независимо мониторировать ситуацию в сети.

2. RTCP имеет постоянный идентификатор транспортного уровня для RTP источника, который называется каноническим именем или cname. Так как SSRC-идентификатор может быть изменен, если будет зафиксировано столкновение или источник будет вынужден рестартовать, получатели нуждаются в cname, для того чтобы отслеживать каждого из участников. Получателям также нужно Cname, чтобы установить соответствие между многими потоками данных от одного участника при реализации нескольких сессий одновременно, например, чтобы синхронизовать аудио- и видео-каналы.

3. Первые две функции требуют, чтобы все участники посылали RTCP-пакеты, следовательно скорость передачи должна контролироваться для того, чтобы RTP мог работать с большим числом участников. При посылке каждым участником своих управляющих пакетов всем остальным любой партнер может независимо определить полное число участников сессии. Это число используется при вычислении частоты посылки пакетов.

4. Четвертая опционная функция служит для передачи минимальной управляющей информации, например идентификаторов участников, для графического интерфейса пользователя. Это полезно для "слабо управляемых" сессий, когда участники входят и выходят без должного контроля и без согласования параметров. RTCP выполняет функции удобного канала для контакта со всеми участниками, но он необязательно поддерживает все коммуникационные требования приложения.

Функции 1-3 являются обязательными, когда RTP используется в среде с IP мультикастингом, и рекомендательными для всех остальных сред.

Протокол SRTP

Secure Real-time Transport Protocol — Безопасный Протокол Передачи Данных В реальном времени определяет профиль RTP и предназначен для шифрования, установления подлинности сообщения, целостности, защиту от замены данных RTP в однонаправленных и multicast передачах медиа и приложениях. Впервые опубликован в IETF в марте 2004 как **RFC 3711**.

Так как RTP тесно связан с RTCP (Real-Time Control Protocol), который может использоваться, чтобы управлять сессией RTP, у SRTP также есть родственный протокол, названный **Secure RTCP** (или **SRTCP**). SRTCP обеспечивает те же самые функции, связанные с безопасностью в RTCP, для той же функциональности SRTP в RTP.

Использование SRTP или SRTCP является необязательным при использовании RTP или RTCP, но даже если SRTP/SRTCP используются, все дополнительные возможности (такие как шифрование и установление подлинности) опциональны и могут быть включены или выключены. Единственное исключение — функция аутентификации сообщений, которая обязательна при использовании SRTCP.

Для шифрования медиа потока (в целях конфиденциальности голосового соединения), SRTP (вместе с SRTCP) стандартизирует использование только единственного шифра, AES.

Помимо шифра AES, SRTP позволяет прямое шифрование, используя так называемый «пустой шифр», который может быть принят как второй поддерживаемый шифр. Фактически, пустой шифр не выполняет шифрования (то есть функции алгоритма шифрования, как если бы ключевой поток содержал только нули, и копирует входной поток в выходной без изменений). Это обязательно для этого способа шифрования, который должен быть обеспечен в любой SRTP-совместимой системе. Также это может использоваться, когда конфиденциальность, которую гарантирует SRTP, не требуется, однако другие возможности SRTP — аутентификация и целостность сообщения — могут использоваться.

Хотя технически в SRTP можно легко встраивать новые алгоритмы шифрования, стандарт SRTP заявляет, что новые алгоритмы шифрования, помимо описанных, не могут просто быть добавлены в реализацию протокола SRTP. Единственный юридически легальный способ добавить новый алгоритм шифрования для совместимости со стандартом SRTP, состоит в том, чтобы опубликовать новый стандарт RFC, где должно быть ясно определено использование нового алгоритма.

Вышеперечисленные алгоритмы шифрования непосредственно не обеспечивают целостность сообщения, давая возможность провести атаку Man-in-the-middle и подделать содержимое сообщения, или, по крайней мере, прослушать ранее переданные данные. Следовательно стандарт SRTP должен также обеспечивать средства защиты целостности данных и защиты от прослушивания.

Чтобы подтвердить подлинность сообщения и защитить его целостность, алгоритм хэширования HMAC-SHA1, определенный в RFC 2104, используется для получения 160-битового хэша, который затем урезается до 80 или до 32 битов, чтобы стать опознавательным признаком, включаемым в пакет. HMAC вычисляется по типу payload пакета и данных в заголовке пакета, включая номер последовательности пакета. Чтобы защитить против встраивания сообщений, приёмник поддерживает индексы ранее полученных пакетов, сравнивает их с индексом каждого нового полученного пакета и пропускает новый пакет, только если он не посылался прежде. Такой подход в большой степени полагается на полную

защиту целостности (чтобы лишить возможности изменить индексы последовательности пакетов для обмана).

Функция генерации ключей используется для получения сеансовых ключей, используемых для шифрования контекста (SRTP, шифровальные ключи контрольного протокола SRTCP и ключи сессий, опознавательные ключи SRTP и SRTCP) из одного единственного главного ключа. Таким образом, протокол обмена ключами позволяет обменяться только главными ключами, остальные необходимые сеансовые ключи будут получены используя данную функцию.

Периодическое изменение самой функции генерации ключей ведет к дополнительным мерам по усилению безопасности. Как правило это препятствует тому, чтобы атакующий собрал большое количество зашифрованного материала, зашифрованного с одним единственным сеансовым ключом. Некоторые взломы легче выполнить, когда имеется большое количество зашифрованного материала. Кроме того, многократное изменение функции генерации ключей обеспечивает прямую и обратную безопасность в том смысле, что расшифрованный сеансовый ключ не ставит под угрозу другие сеансовые ключи, полученные из того же самого главного ключа. Это означает, что, даже если взломщику удалось получить определенный сеансовый ключ, он не в состоянии расшифровать сообщения, обеспеченные с предыдущими и более поздними сеансовыми ключами, полученными из того же самого главного ключа. (Хотя, конечно, полученный главный ключ даст все сеансовые ключи, полученные из него.)

SRTP полагается на внешний протокол обмена ключами, чтобы установить главный начальный ключ. Разработаны два специальных протокола для использования с SRTP — **ZRTP** и **MIKEY**.

Есть и другие методы, чтобы договориться о ключах SRTP. Несколько различных производителей предлагают продукты, которые используют метод обмена ключей SDES.