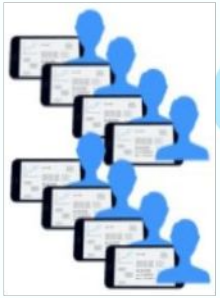
# SRT — предпосылки к появлению и его возможности

GaraninApps.com

SRTMiniServer.ru

# Что делаем мы



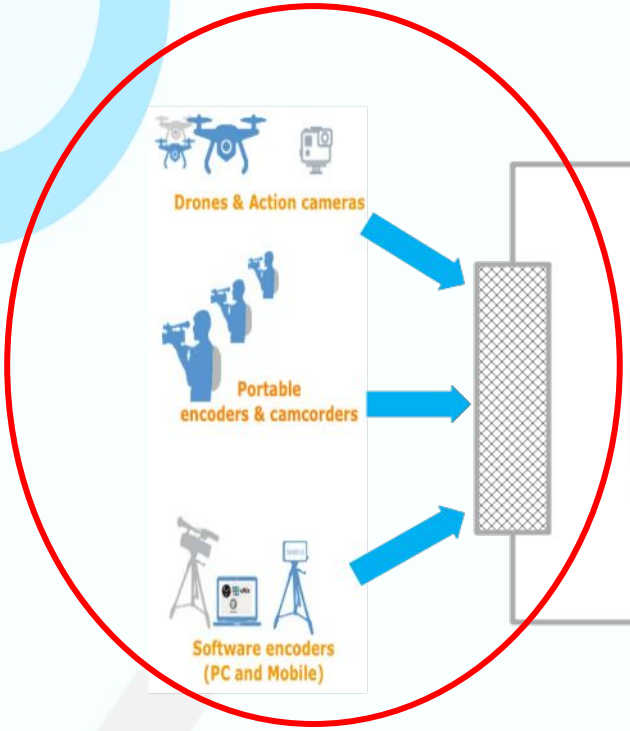
ИСТОЧНИК

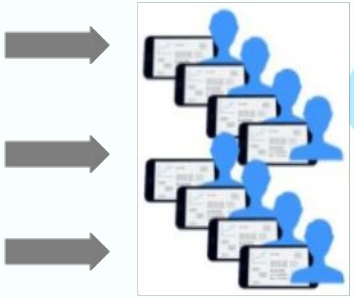
СТУДИЯ

ЗРИТЕЛЬ

SRTMiniServer.ru

# Что делаем мы





ИСТОЧНИК

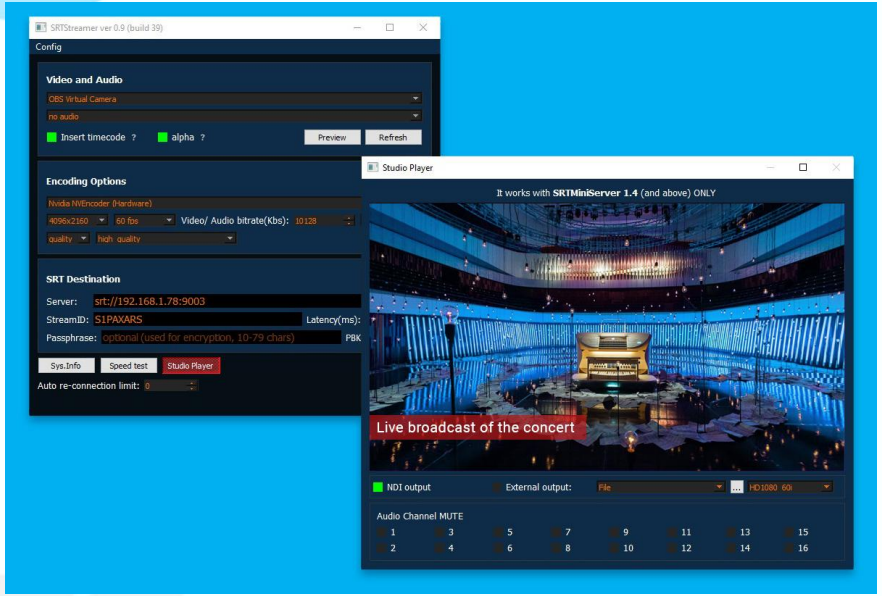
СТУДИЯ

ЗРИТЕЛЬ

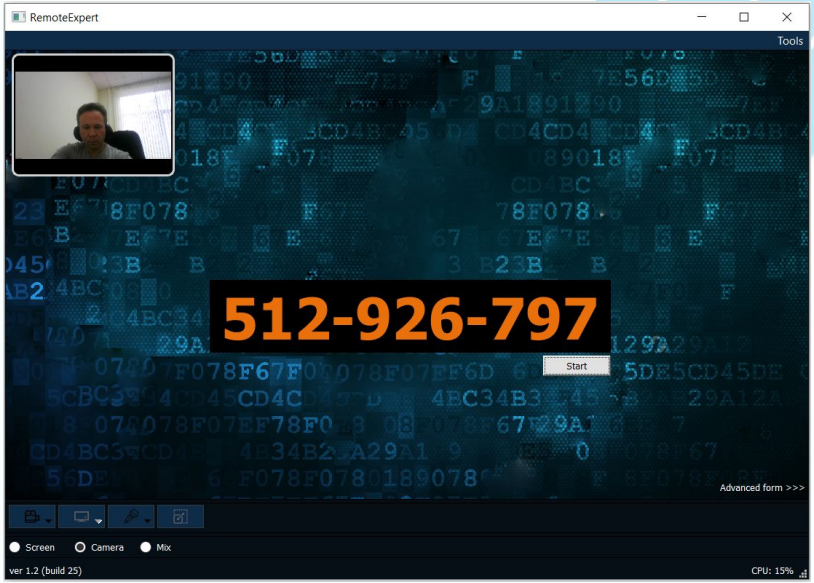
SRTMiniServer.ru

# Пример 1: free SRT-энкодеры

SRT Streamer

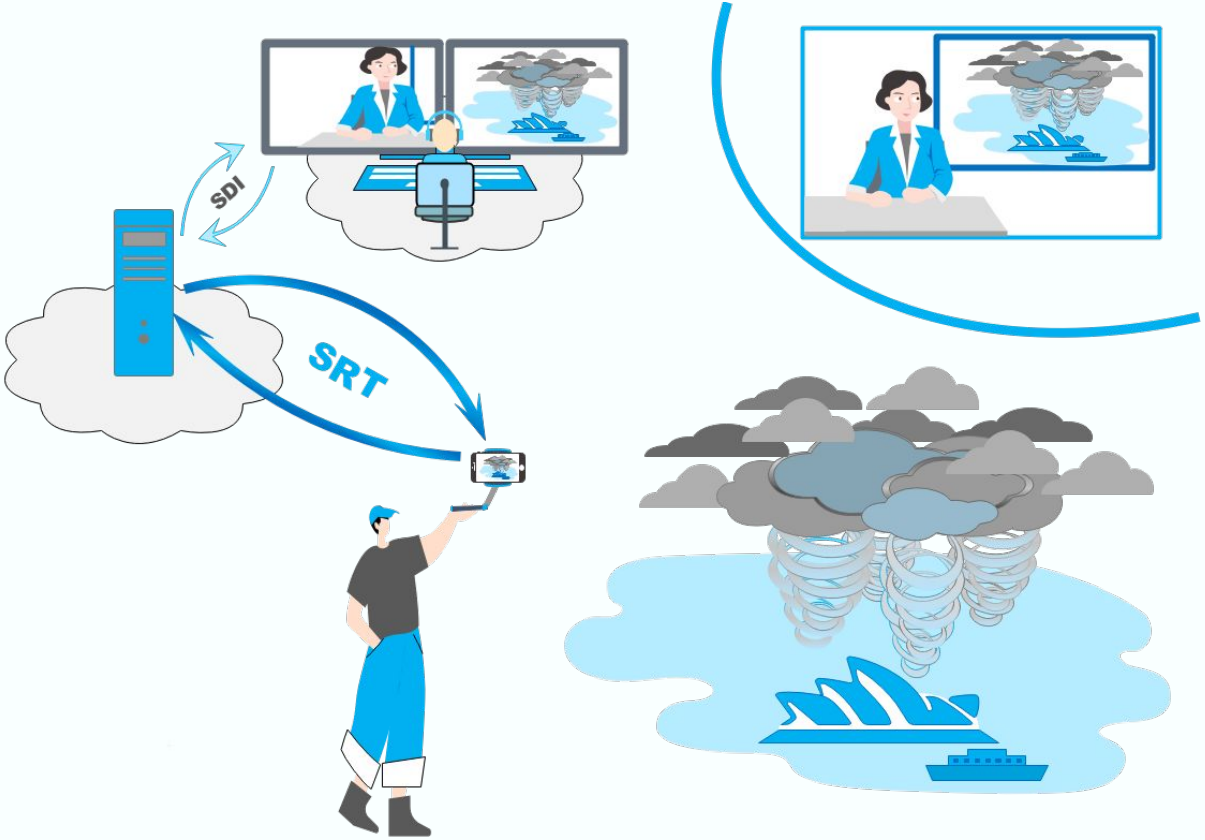



RemoteExpert



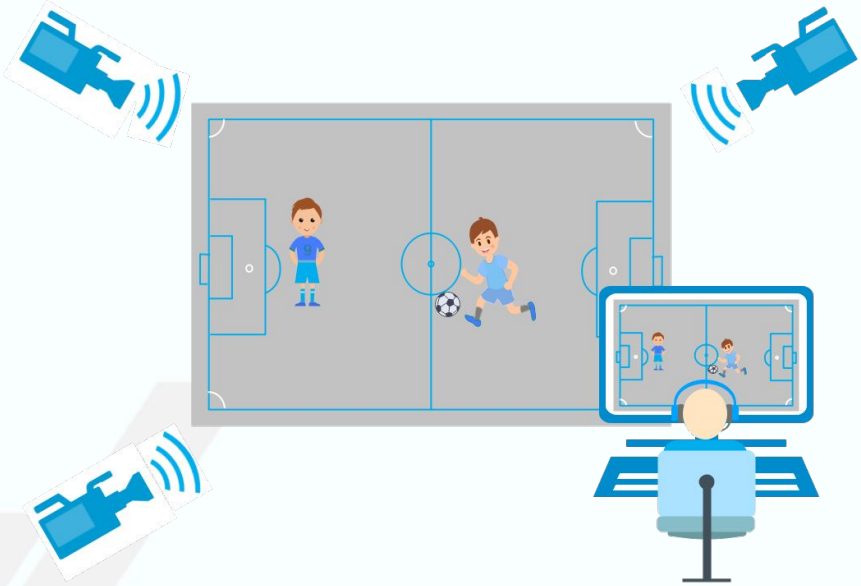

SRTMiniServer.ru

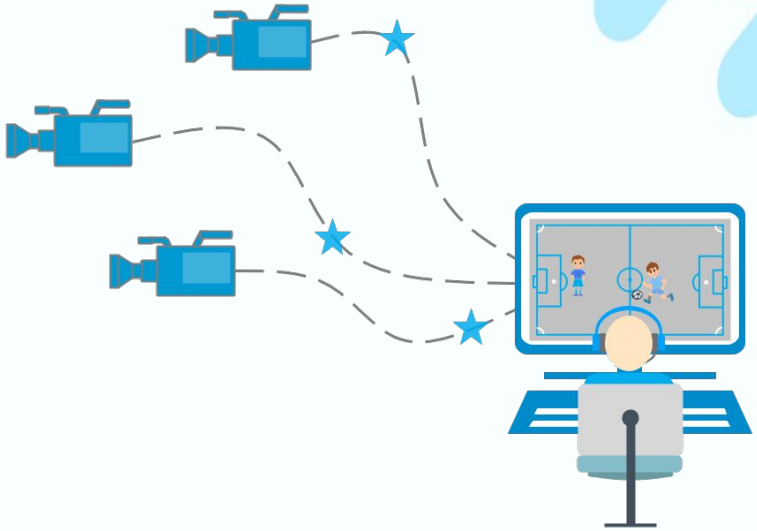
# Пример 2: мобильные репортеры




SRTMiniServer.ru

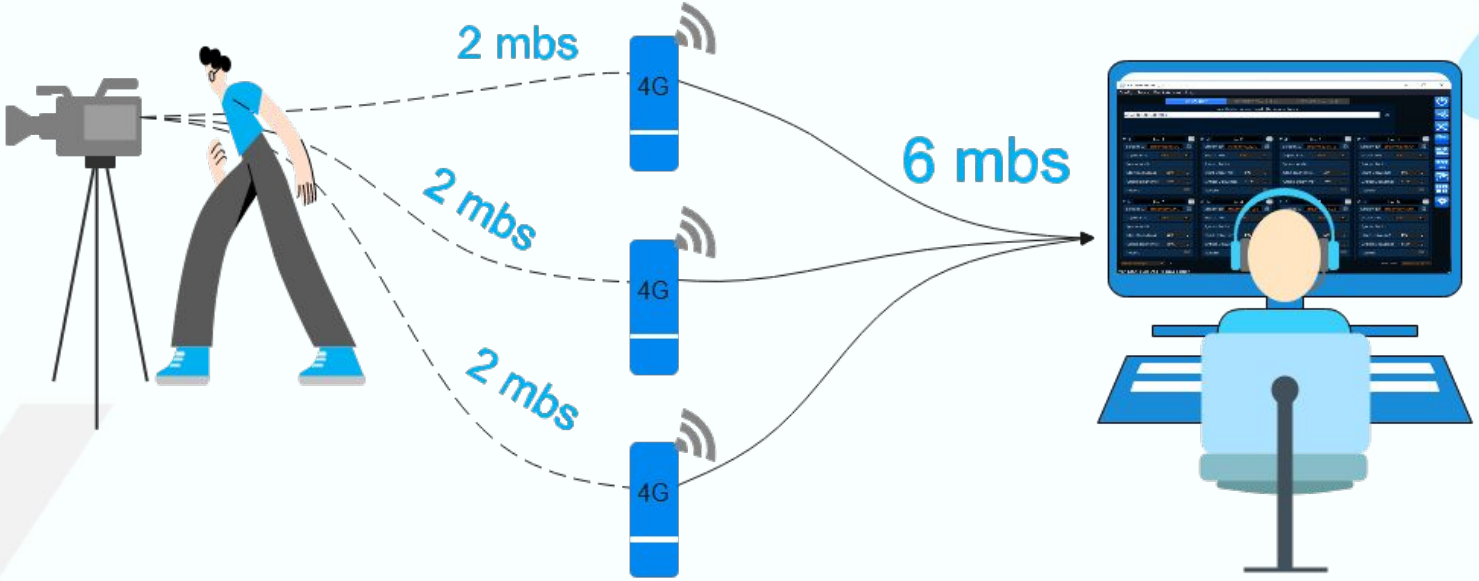
# Пример 3: Тайм синхронизация





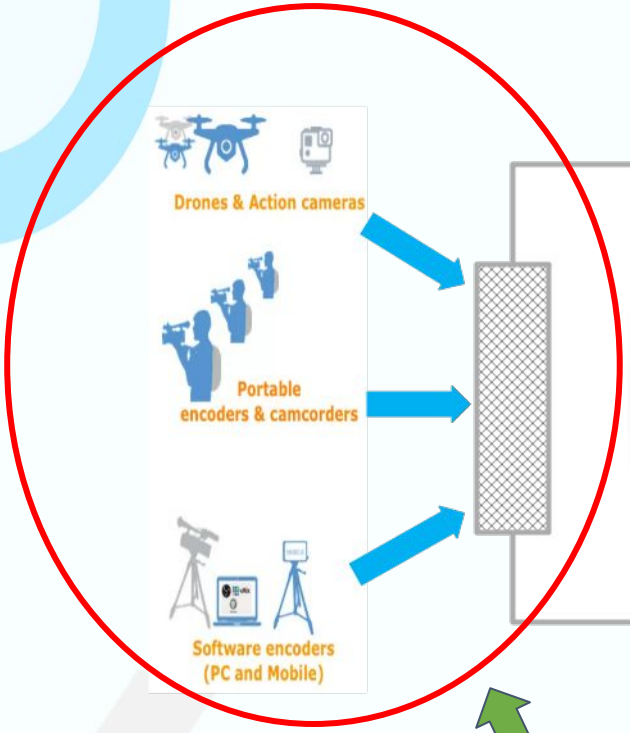
SRTMiniServer.ru

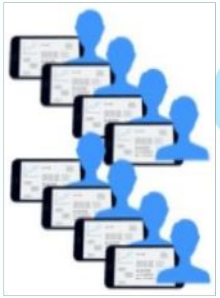
# Пример 4: бондинг




SRTMiniServer.ru

# Что делаем мы





ИСТОЧНИК

СТУДИЯ

ЗРИТЕЛЬ

SRTMiniServer.ru

# ПРЕДПОСЫЛКИ К ПОЯВЛЕНИЮ SRT

// не много истории

SRTMiniServer.ru

# Расклад в начале 2000

Desktop, Flash, RTMP

ИСТОЧНИК

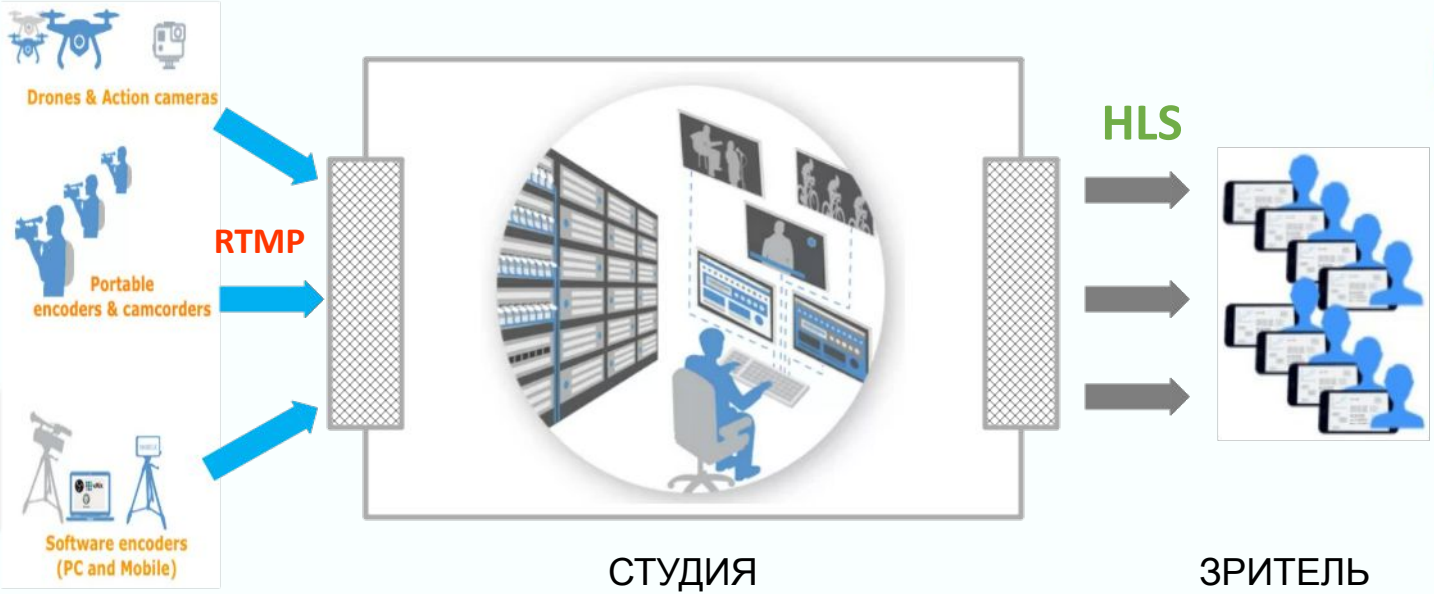
СТУДИЯ

ЗРИТЕЛЬ

SRTMiniServer.ru

# Расклад на недавний момент

Mobile, HLS,  RTMP




ИСТОЧНИК

СТУДИЯ

ЗРИТЕЛЬ

SRTMiniServer.ru

# Претензии к RTMP

* проприетарный. игнор разработчиков. Adobe его не развивает.

* нет поддержки новых кодеков (только h264 и 2ch audio)

* базируется на TCP

SRTMiniServer.ru

# SRT: ответ от Haivision

* Open-source протокол. Ориентация на коммунити разработчиков. Активное развитие.

* Codec-agnostic: можно передавать что угодно

* Базируется на UDP

SRTMiniServer.ru

# SRT: Codec Agnostic

- HEVC, AV1 etc
- Multi-audio
- Прикладные данные: файлы, фото, сообщения

SRTMiniServer.ru

# SRT построен на UDP ?!!! Какого чёрта….

TCP

UDP

Типовые возражения:

- UDP это потеря данных
- UDP блокируется провайдерами

SRTMiniServer.ru

# TCP - заложник возраста (1976)

В каких условиях создавался TCP в 1976:

- Медь
- “Слабые” ЭВМ, борьба за каждый байт и такт процессора
- Наивные представления о возможном трафике

SRTMiniServer.ru

# UDP - залог скорости

Специализированные протоколы построенные на UDP - тренд поддерживаемый крупными игроками, включая Google

Позволяют добиться значительно большей скорости передачи данных без потери

SRTMiniServer.ru

# UDT

// UDP-based Data Transfer Protocol

**UDT**

(Yunhong Gu)

Стартовал в 2001
Последний стабильный релиз: 2013 год

Имеет разные награды по скорости передачи файлов

**UDT - для передачи файлов (целостность данных)**

UDT использует периодические подтверждения ACK, чтобы подтвердить пакетную доставку, а отрицательные ACK используются, чтобы сообщить о пакетной потере.

Периодические ACK помогают облегчить управление движением контрольных сообщений на обратном пути, когда скорость передачи данных высока,

потому что в этих ситуациях количество ACK пропорционально времени, а не количеству пакетов данных.

SRTMiniServer.ru

# SRT построен на UDT

```
/*****************************************************************************
written by
   Yunhong Gu, last updated 07/09/2011
modified by
   Haivision Systems Inc.
*****************************************************************************/

#include "platform_sys.h"

#include <exception>
```

```
int srt_connect_bind(SRTSOCKET u,
        const struct sockaddr* source,
        const struct sockaddr* target, int target_len)
{
    return CUDT::connect(u, source, target, target_len);
}

int srt_rendezvous(SRTSOCKET u, const struct sockaddr* local_name, int local_namelen,
        const struct sockaddr* remote_name, int remote_namelen)
{
    bool yes = 1;
    CUDT::setsockopt(u, 0, SRTO_RENDEZVOUS, &yes, sizeof yes);

    // Note: PORT is 16-bit and at the same location in both sockaddr_in and sockaddr_in6.
    // Just as a safety precaution, check the structs.
    if ( (local_name->sa_family != AF_INET && local_name->sa_family != AF_INET6)
            || local_name->sa_family != remote_name->sa_family)
        return CUDT::APIError(MJ_NOTSUP, MN_INVAL, 0);

    const int st = srt_bind(u, local_name, local_namelen);
    if (st != 0)
        return st;

    return srt_connect(u, remote_name, remote_namelen);
}
```

SRTMiniServer.ru

# Промежуточное резюме

* Open-source протокол ( https://github.com/Haivision/srt )

* Codec-agnostic: можно передавать что угодно

* Базируется на прочном фундаменте “авторитетного” протокола UDT (построенного на базе UDP), обеспечивающего целостность данных.

- Хорошая документация

SRTMiniServer.ru

# Немного деталей

* Режимы соединения

* Режимы передачи данных (Buffer, Message, Live)

SRTMiniServer.ru

# Детали: режимы соединения

* Caller & Listener

* режим Рандеву ( rendezvous)

SRTMiniServer.ru

# Детали: режимы соединения

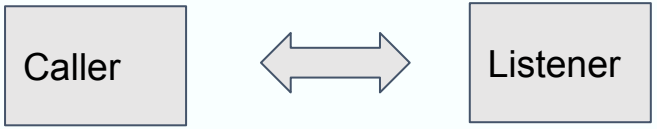
Listener - слушает UDP-порт

Caller - инициирует коннект на IP:PORT  Listener’a

Направление данных может быть любым

1) Соединение
(нужно знать IP:PORT листенера)

Нужен “белый IP”




2) Передача данных

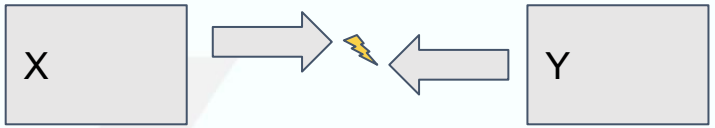
SRTMiniServer.ru

# Детали: режимы соединения

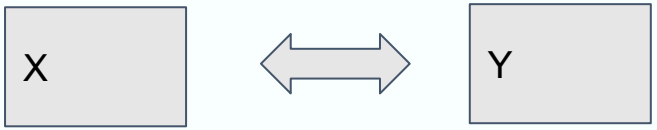
Rendezvous  (пришло из UDT) - для пробивки Firewall’ов

базируется на старом методе “UDP Hole punching”.

Не нужен белый IP.
Но обе стороны должны знать исходящие IP и порты.




1) Соединение




2) Передача данных

- В отличие от WebRTC : аналога ICE в SRT нет. Делаем сами.

SRTMiniServer.ru

# Немного деталей

* Режимы соединения

* Режимы передачи данных (Buffer, Message, Live)

SRTMiniServer.ru

# Детали: режимы передачи данных

Buffer - для целостной передачи всех данных

Message - допускает пропадание сообщения

Live - специальный режим для видео-стриминга (самый медленный)

```
SRT_TRANSTYPE ttype = SRTT_LIVE;
set_srt_option(srt_sock,SRTO_TRANSTYPE, "SRTO_TRANSTYPE", &ttype, sizeof (ttype) );
```

SRTMiniServer.ru

# Детали: режимы передачи данных

## Transmission Method: Buffer

Setting `SRTO_TRANSTYPE` to `SRTT_FILE` sets the following socket options:

- `SRTO_TSBPDMODE` = false
- `SRTO_RCVLATENCY` = 0
- `SRTO_PEERLATENCY` = 0
- `SRTO_TLPKTDROP` = false
- `SRTO_LINGER` = 180 s
- `SRTO_MESSAGEAPI` = false
- `SRTO_NAKREPORT` = false
- `SRTO_RETRANSMITALGO` = 0
- `SRTO_PAYLOADSIZE` = 0
- `SRTO_CONGESTION` = "file"

SRTMiniServer.ru

# Детали: режимы передачи данных

**Transmission Method:** Message

Setting `SRTO_TRANSTYPE` to `SRTT_FILE` sets the following socket options:

- `SRTO_TSBPDMODE` = false
- `SRTO_RCVLATENCY` = 0
- `SRTO_PEERLATENCY` = 0
- `SRTO_TLPKTDROP` = false
- `SRTO_LINGER` = 180 s
- `SRTO_MESSAGEAPI` = true
- `SRTO_NAKREPORT` = false
- `SRTO_RETRANSMITALGO` = 0
- `SRTO_PAYLOADSIZE` = 0
- `SRTO_CONGESTION` = "file"

SRTMiniServer.ru

# Детали: режимы передачи данных

## Transmission Method: Live

Setting `SRTO_TRANSTYPE` to `SRTT_LIVE` sets the following socket options:

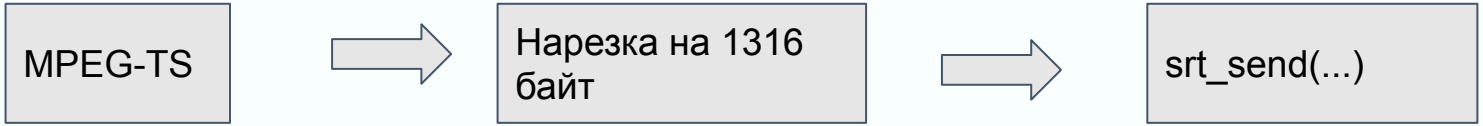
- `SRTO_TSBPDMODE` = true
- `SRTO_RCVLATENCY` = 120
- `SRTO_PEERLATENCY` = 0
- `SRTO_TLPKTDROP` = true
- `SRTO_LINGER` = 0
- `SRTO_MESSAGEAPI` = true
- `SRTO_NAKREPORT` = true
- `SRTO_RETRANSMITALGO` = 1
- `SRTO_PAYLOADSIZE` = 1316
- `SRTO_CONGESTION` = "live"

SRTMiniServer.ru

# Детали: режимы передачи данных

Live - для видео-стриминга

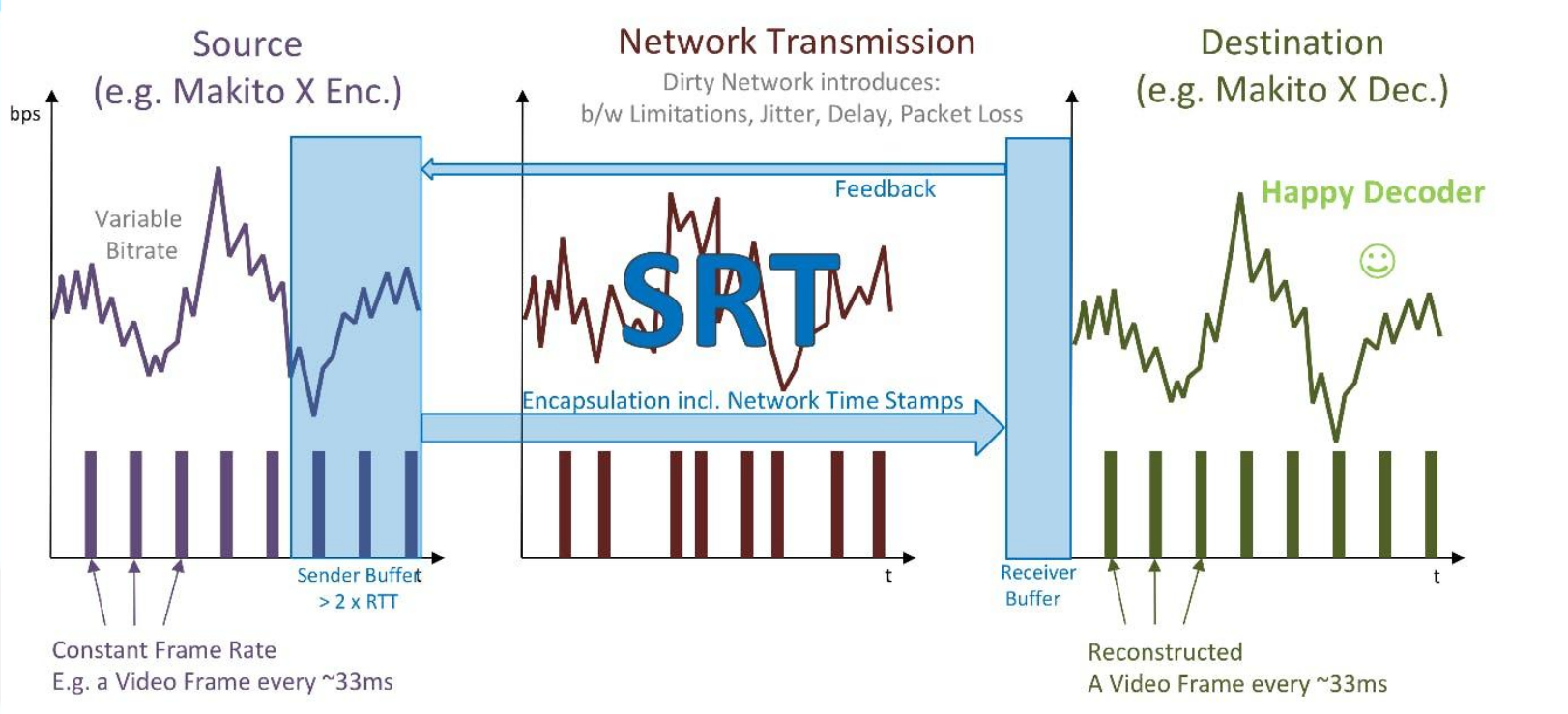
Общий Workflow:




SRTMiniServer.ru

# Детали: режимы передачи данных

Live - самый медленный из-за стабилизирующего буфера.




SRTMiniServer.ru

# Немного деталей

* Режимы соединения
// Caller, Listener, Randezvous

* Режимы передачи данных
// Buffer, Message, Live

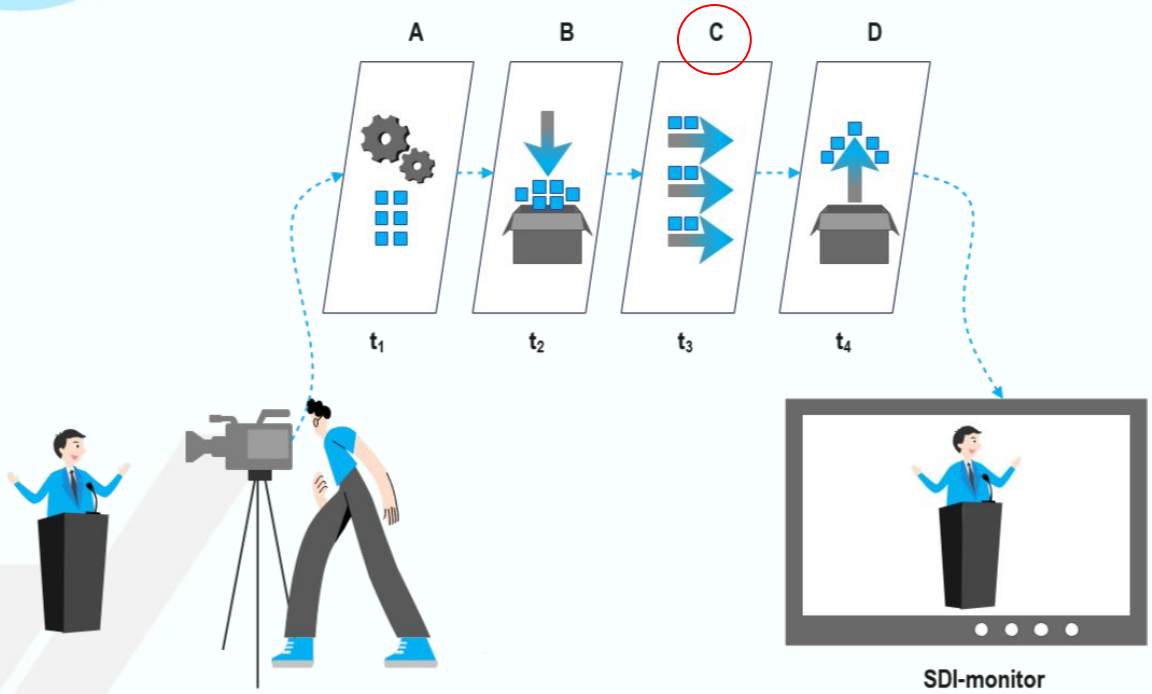
SRTMiniServer.ru

# Популярные вопросы

- Что там с реальной задержкой от стекла до стекла.
- SRT и Firewall.  Ура,  революция?
- SRT vs NDI
- SRT и Бондинг (свой, поверх MPTCP)

SRTMiniServer.ru

# SRT: что там с задержкой




- SRT - отвечает за транспорт
- Регулируется через параметр Latency
- На реальную задержку сильно влияет энкодер

SRTMiniServer.ru

# SRT: что там с задержкой

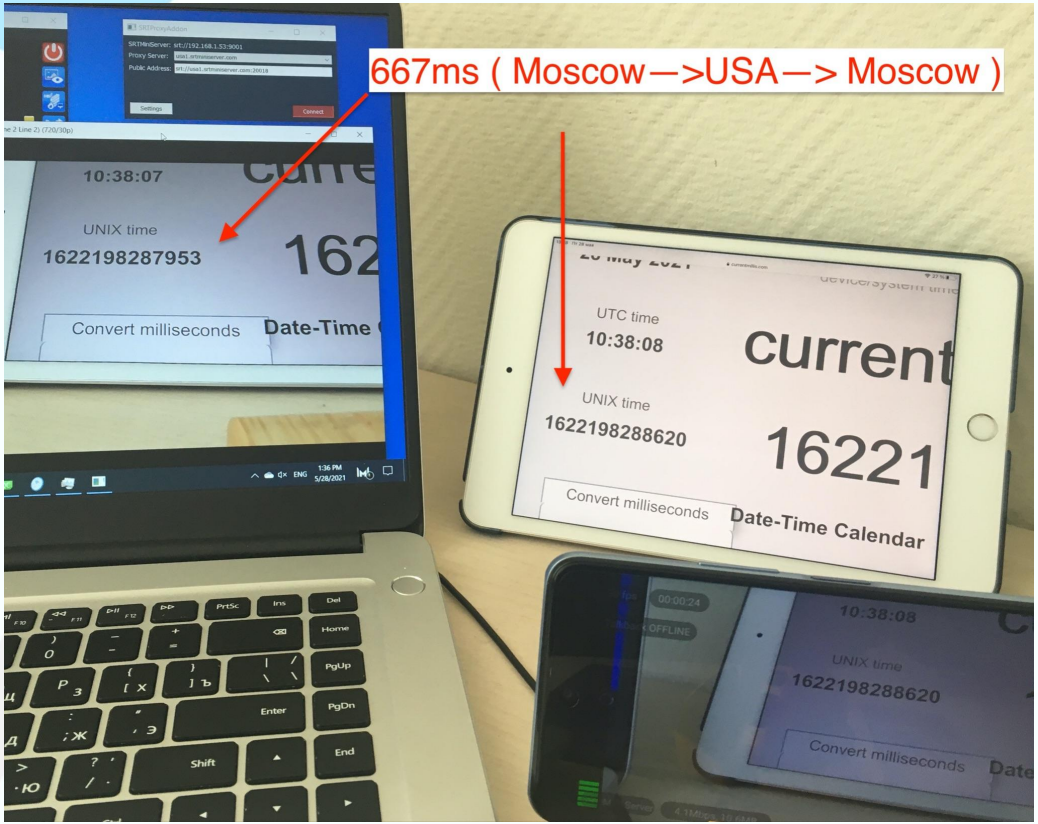



Схема:
Larix по LTE(Megafon) стримит на SRT MiniServer (офисный провайдер) через USA-прокси.

Задержка: 667 ms

SRTMiniServer.ru

# SRT: что там с задержкой

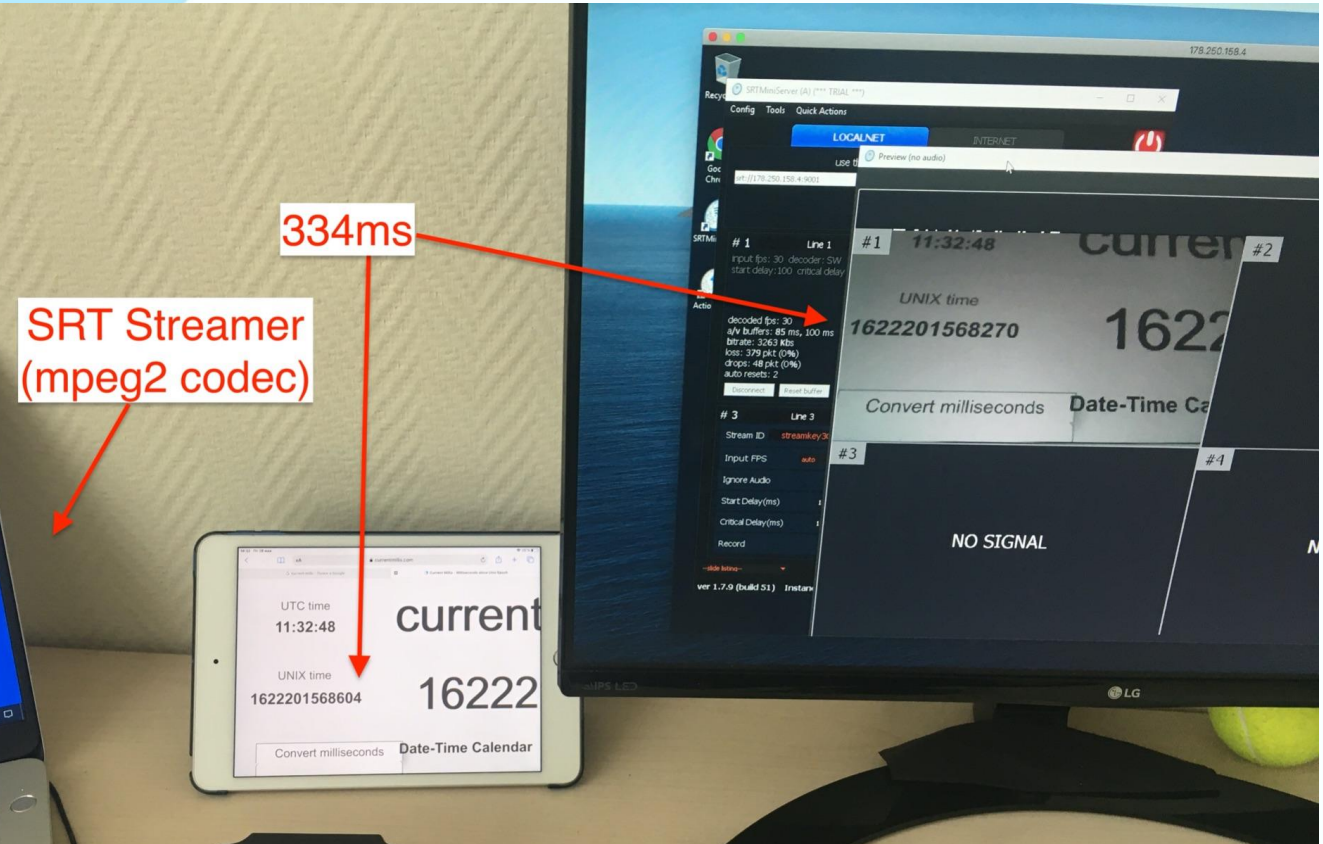


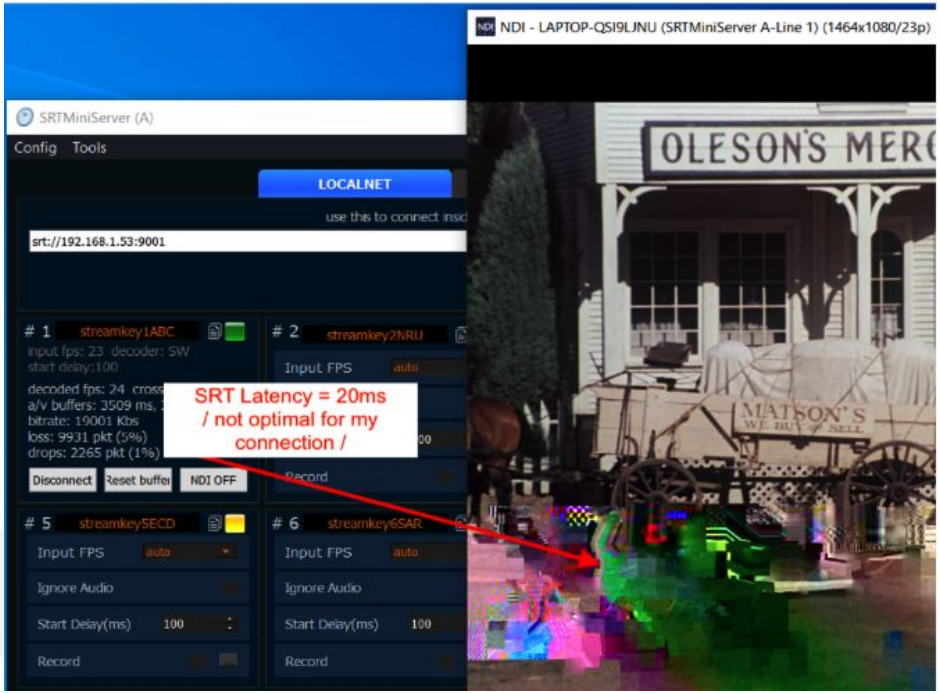

Схема:
SRTStreamer (msk) стримит на облачный сервер в msk, используя кодек mpeg2 (мы смотрим по RDP)

Задержка: 334ms

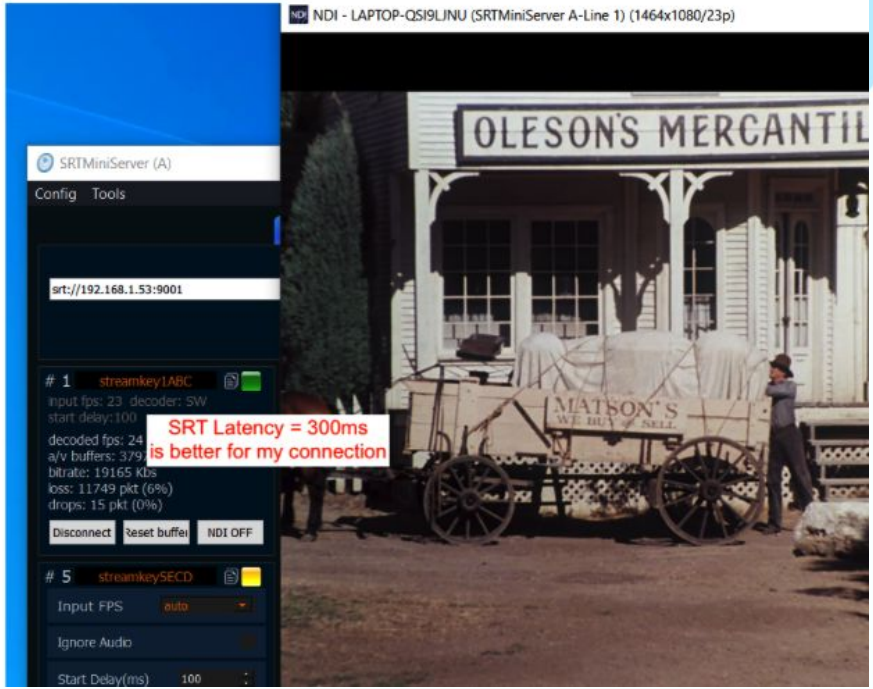
SRTMiniServer.ru

# Про “квадратики” (скорость vs качество)

SRT Latency 20 ms

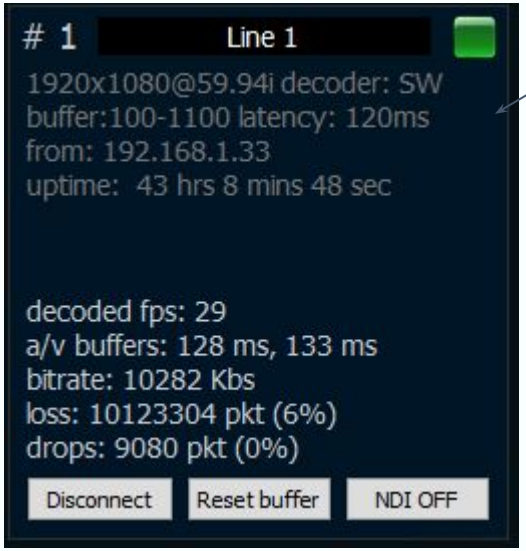



SRT Latency 300 ms




SRTMiniServer.ru

# Про “квадратики”

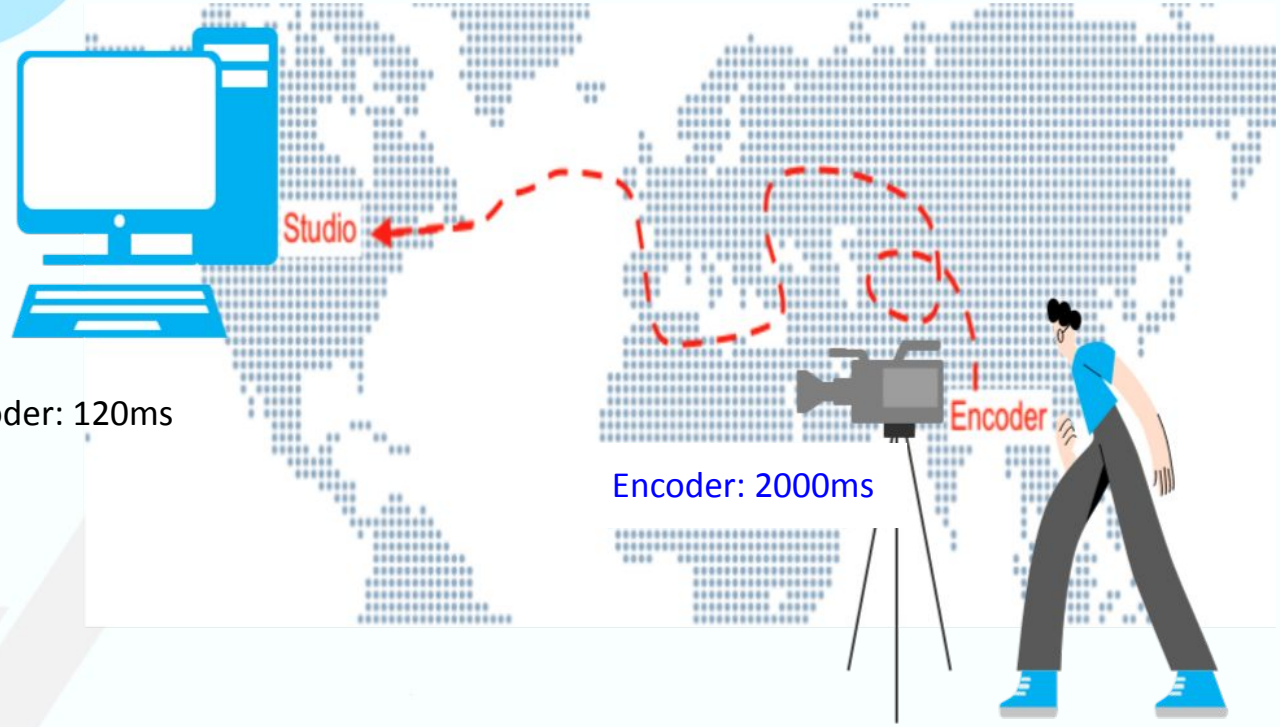



"Lost" (потерянные) означают больные пакеты
"Drops"(отбрасывание) означает мертвые пакеты.
"Latency"(задержка) — это время лечения каждого больного пакета.
Высокое значение означает больше шансов восстановить больной пакет.
2*RTT

SRTMiniServer.ru

# Про “квадратики”

Где ставить Latency?




Decoder: 120ms

Encoder: 2000ms

SRTMiniServer.ru

# Популярные вопросы

- Что там с реальной задержкой от стекла до стекла.
- SRT и Firewall.  Ура,  революция?
- SRT vs NDI
- SRT и Бондинг (свой, поверх MPTCP)

SRTMiniServer.ru

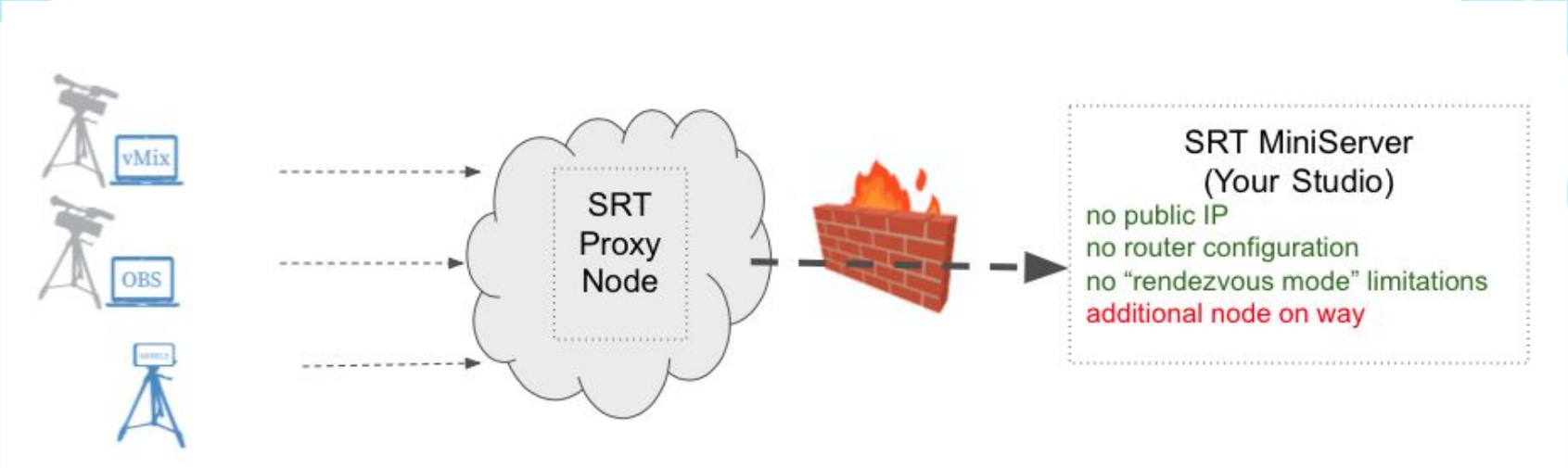
# SRT и Firewall

Специальный режим “Рандеву (Rendezvous).
Маркетинговая обёртка для старого метода “UDP Hole punching” (Skype, WebRTC etc)

- Не тривиален в настройке.
- Серебряная пуля (НЕТ): Не все файрволы пробиваются.
- Нужен “план Б”. Но его нет в SRT.

SRTMiniServer.ru

# SRT и Firewall (Plan B)




SRTMiniServer.ru

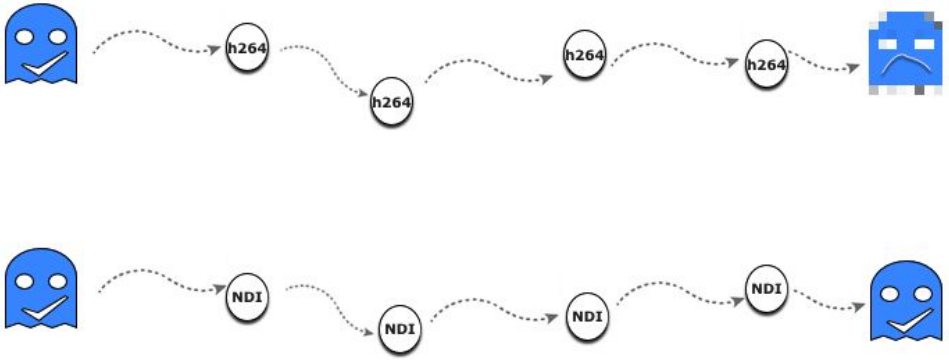
# Популярные вопросы

- Что там с реальной задержкой от стекла до стекла.
- SRT и Firewall.  Ура,  революция?
- SRT vs NDI
- SRT и Бондинг (свой, поверх MPTCP)

SRTMiniServer.ru

# SRT vs NDI

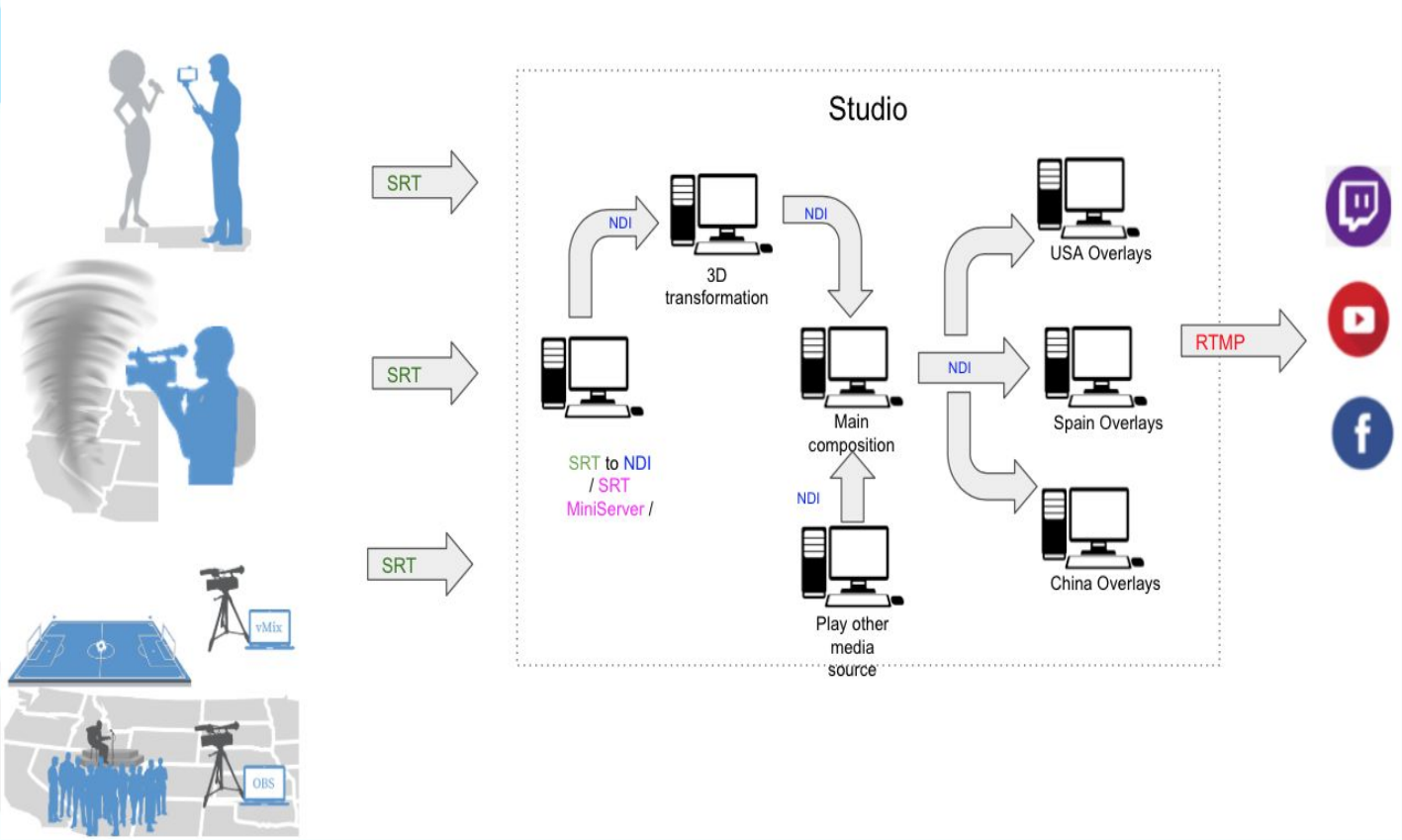
What happens if we do many encode-decode….




NDI is the first ever codec to provide multi-generational stability

| Example NDI video stream | Approximate bandwidth required |
|---|---|
| 1 x UHDp60 video stream | 250 Mbps |
| 1 x UHDp30 video stream | 200 Mbps |
| 1 x 1080p60 video stream | 125 Mbps |
| 1 x 1080i60 video stream | 100 Mbps |
| 1 x 720p60 video stream | 90 Mbps |
| 1 x SD video stream | 20 Mbps |

# SRT vs NDI



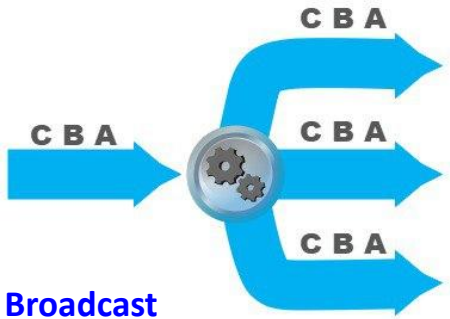

SRTMiniServer.ru

# Популярные вопросы

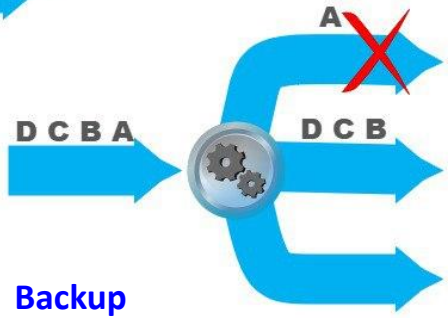
- Что там с реальной задержкой от стекла до стекла.
- SRT и Firewall.  Ура,  революция?
- SRT vs NDI
- SRT и Бондинг (нативный, поверх MPTCP, что-то своё)

SRTMiniServer.ru

# Нативный: SRT 1.5




Broadcast

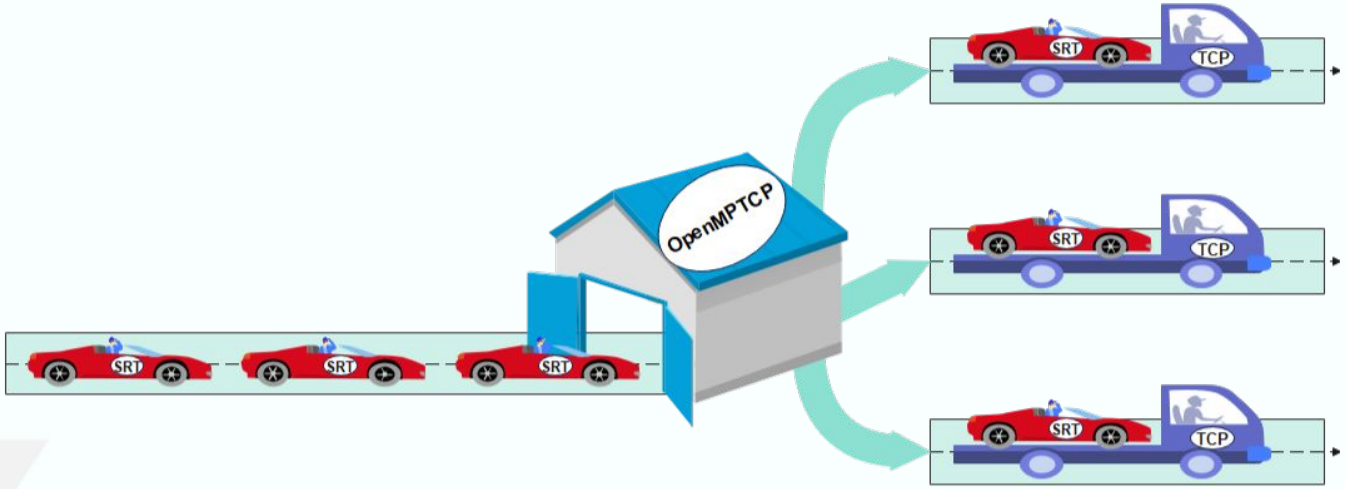



Backup

Balancing
(суммирование)

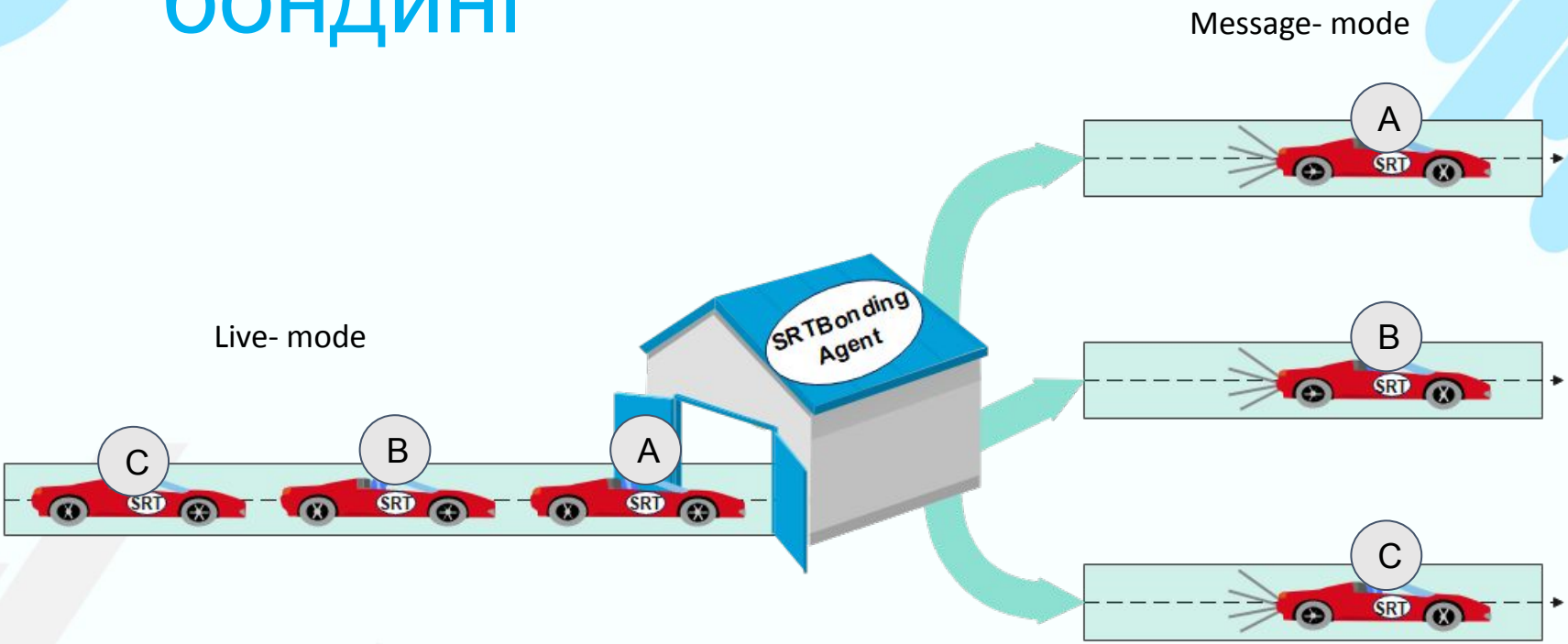
TMiniServer.ru

# SRT и MPTCP




SRTMiniServer.ru

# Поэтому сделали свой бондинг




SRTMiniServer.ru

# Популярные вопросы

- Что там с реальной задержкой от стекла до стекла.
- SRT и Firewall.  Ура,  революция?
- SRT vs NDI
- SRT и Бондинг (нативный, поверх MPTCP, кастомный)

SRTMiniServer.ru

# WebRTC vs SRT

В стандарте WebRTC

- фиксированы кодеки
- громоздкий

SRT более лёгок;
как бы подстраивается в имеющиеся MPEG-TS потоки.

SRTMiniServer.ru

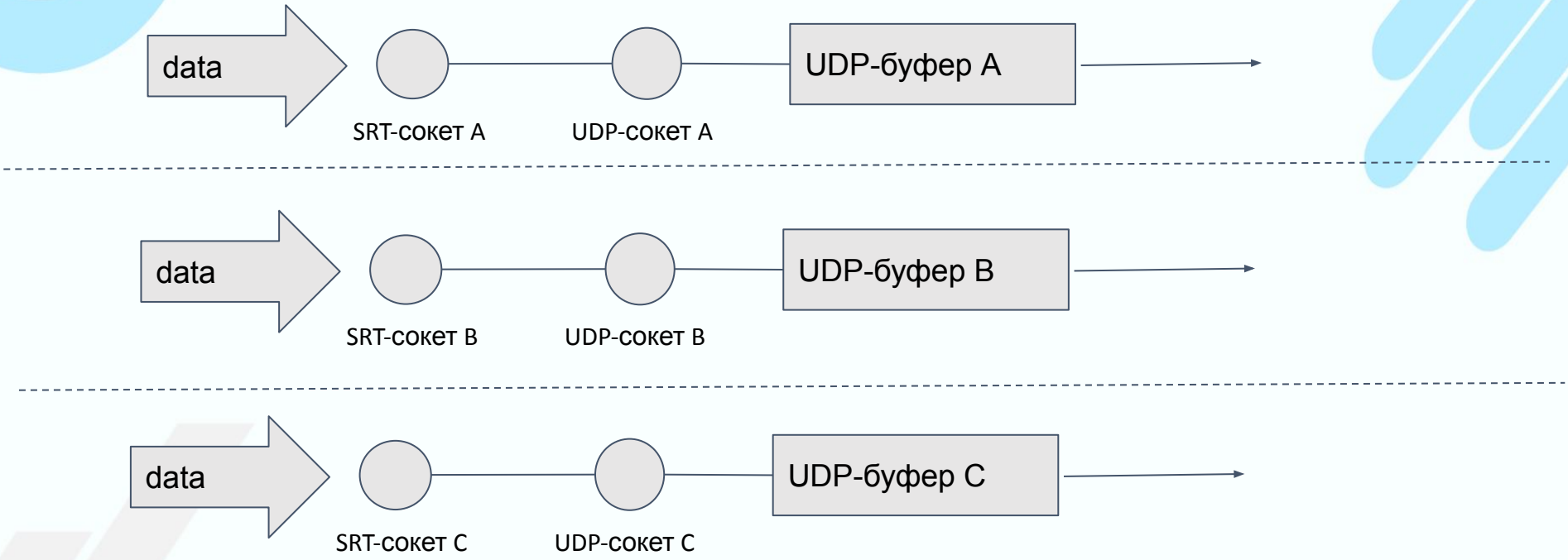
# Грабли на которые наступили

На примере бондинг решения

Шаринг UDP-сокетов

SRTMiniServer.ru

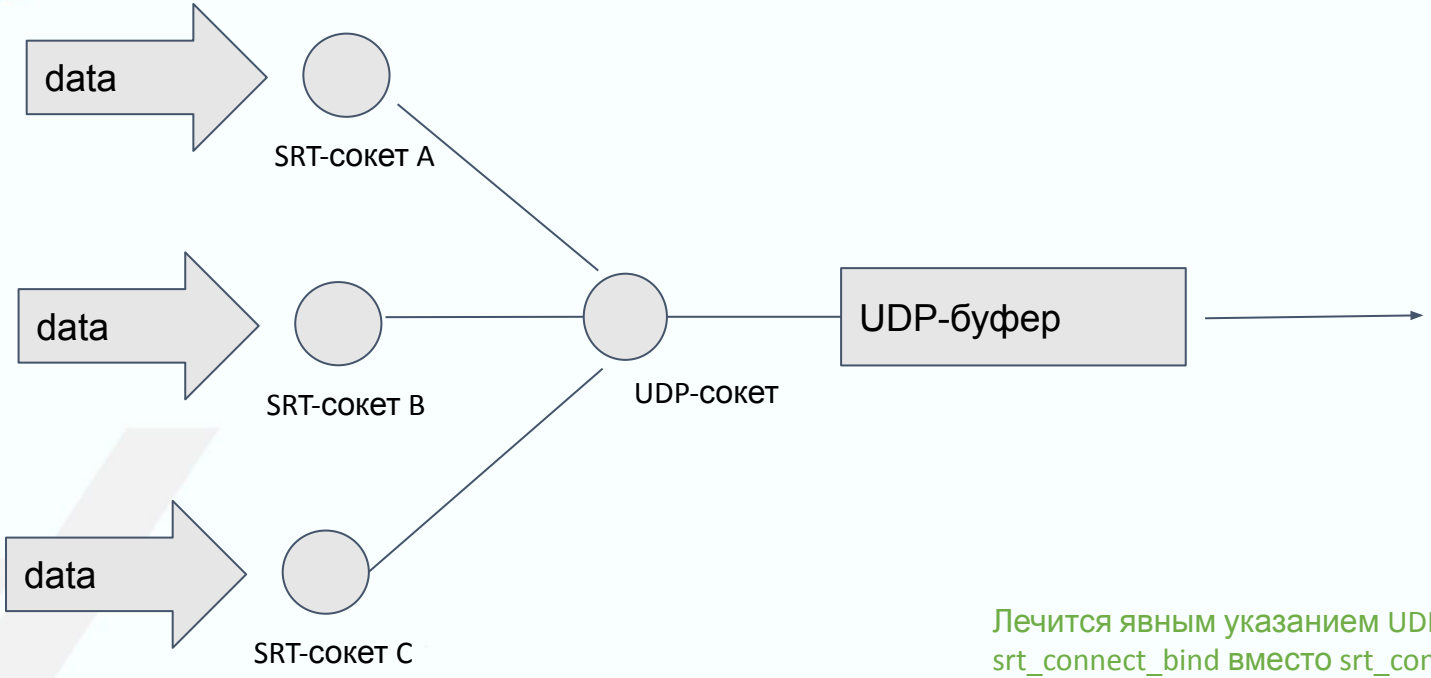
Предполагали что будет работать так:




SRTMiniServer.ru

# Шаринг UDP-сокета

А работало так (по умолчанию):




Лечится явным указанием UDP-порта
srt_connect_bind вместо srt_connect

SRTMiniServer.ru

# Спасибо за внимание!

SRTMiniServer.RU

Michael@GaraninApps.com

SRTMiniServer.ru