

SRT — предпосылки к появлению и его возможности

GaraninApps.com

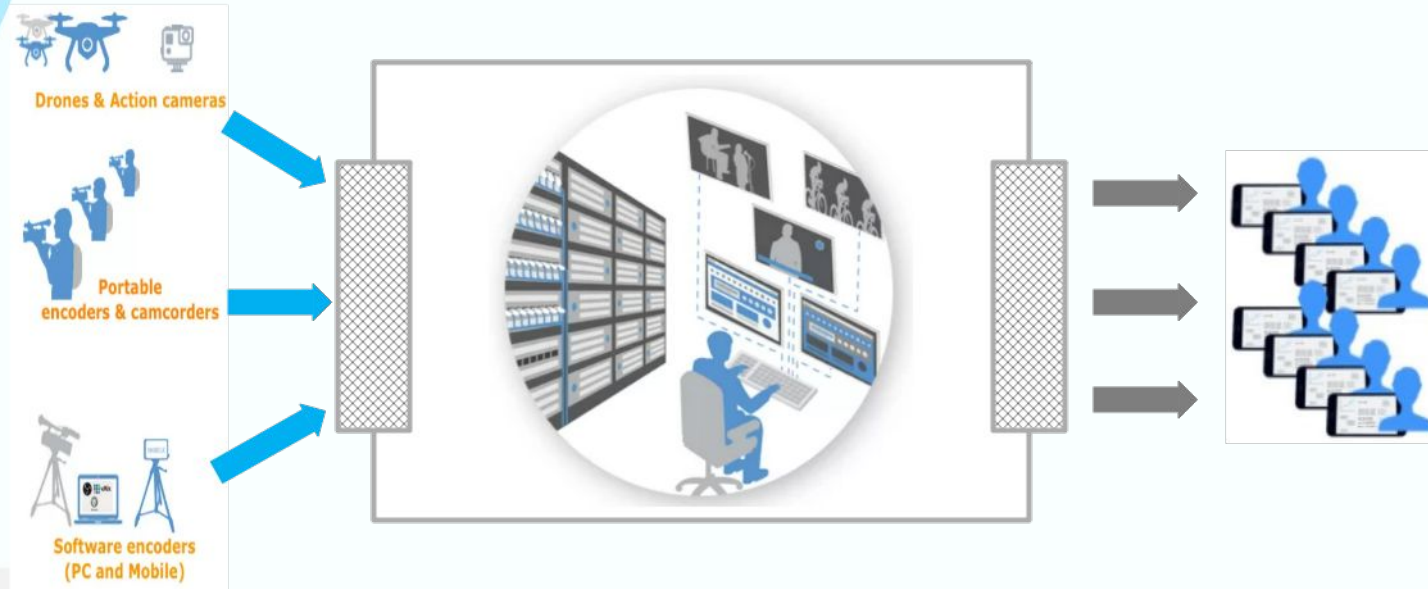
Откуда узнал про SRT



Сообщество
стримеров

AVStream

Что делаем мы

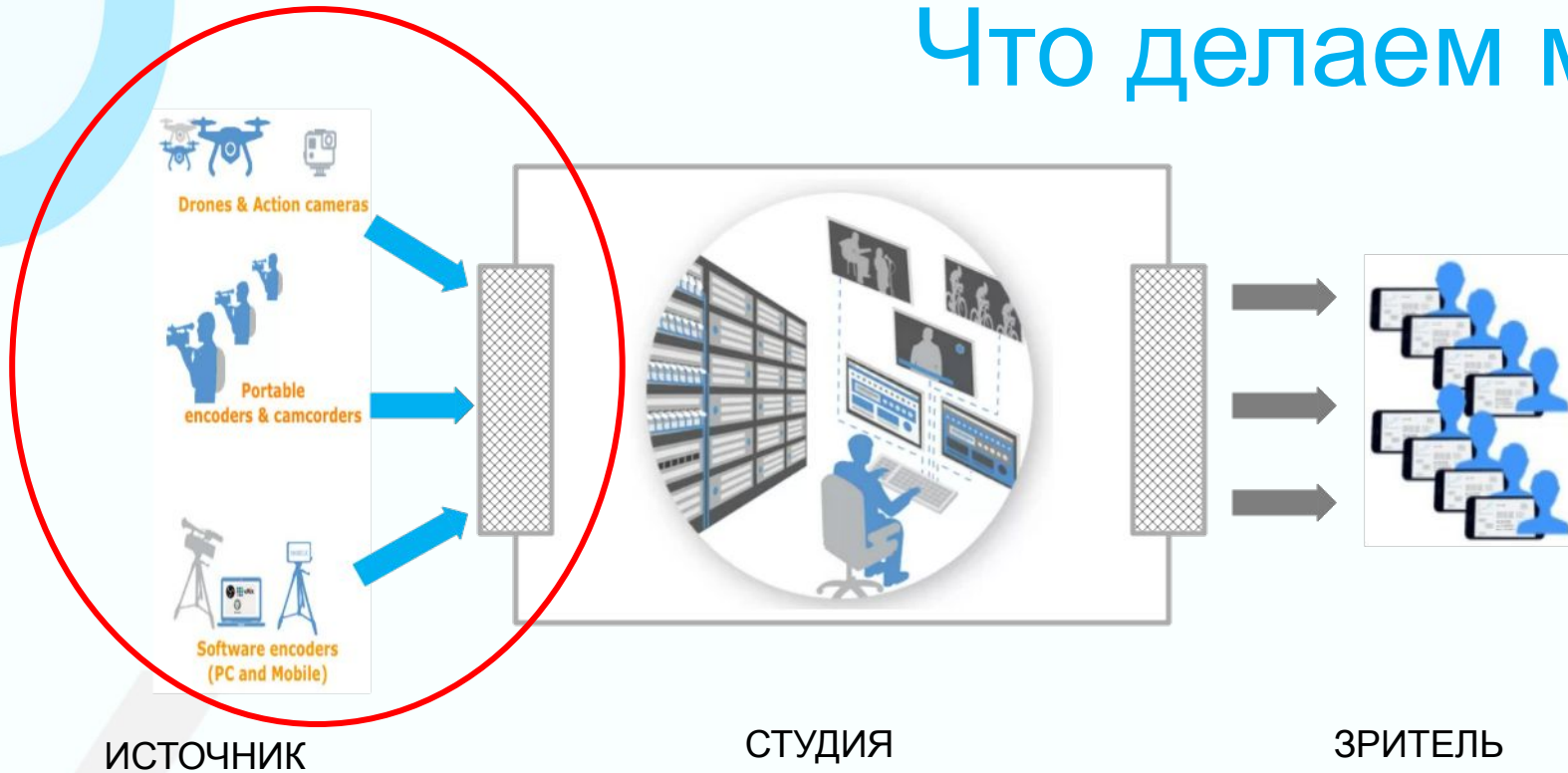


ИСТОЧНИК

СТУДИЯ

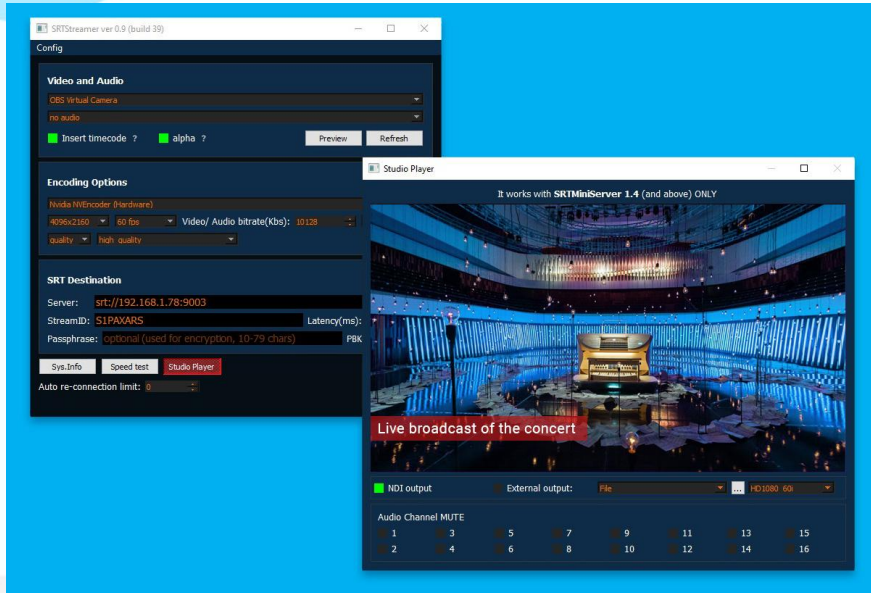
ЗРИТЕЛЬ

Что делаем мы

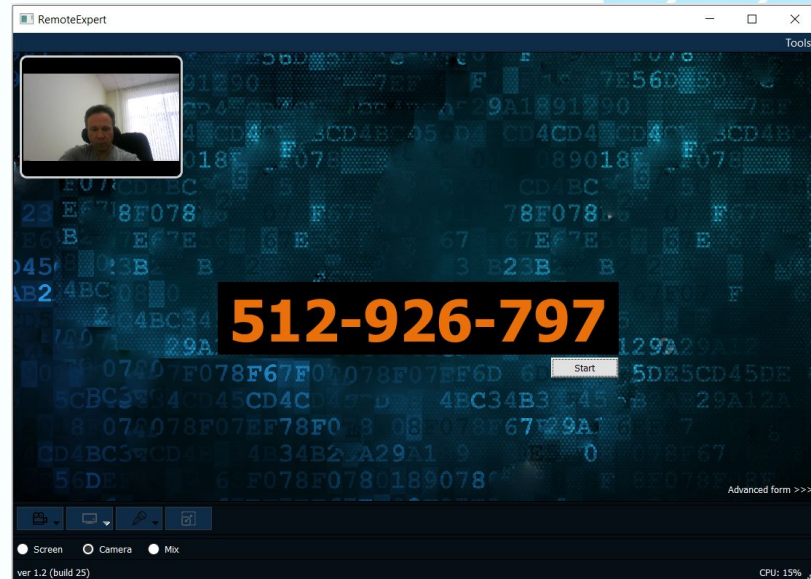


Пример 1: free SRT-энкодеры

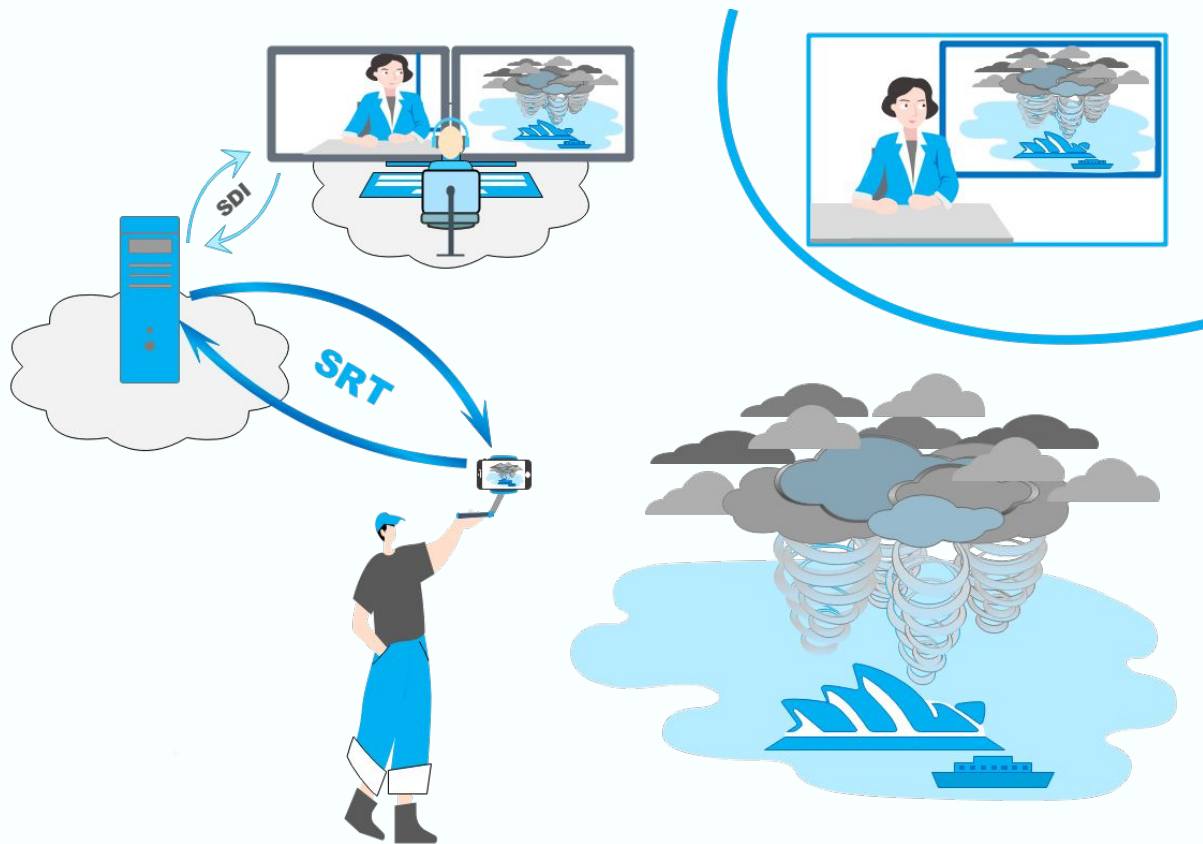
SRT Streamer



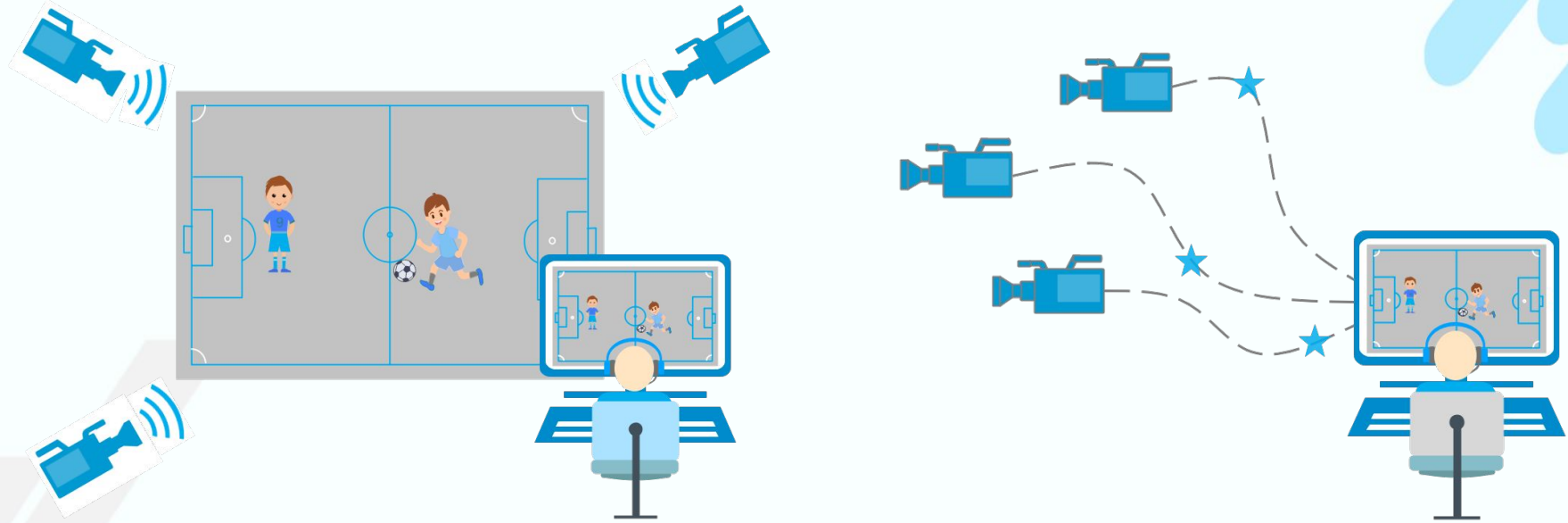
RemoteExpert



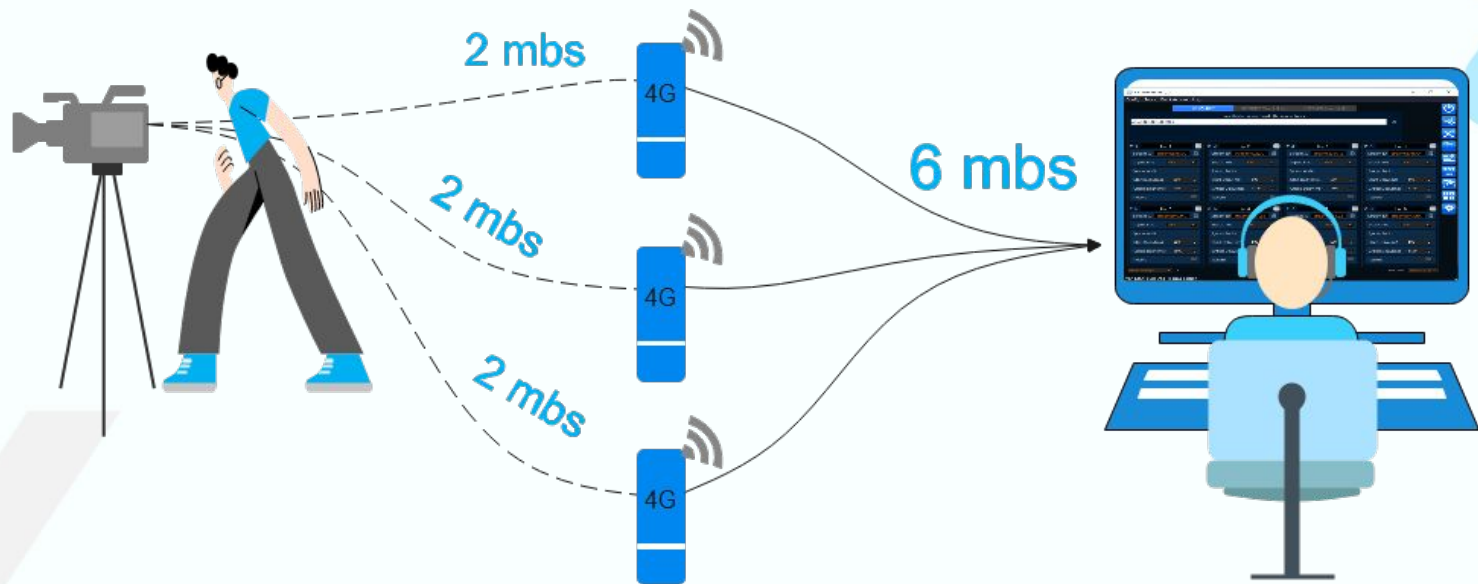
Пример 2: мобильные репортеры



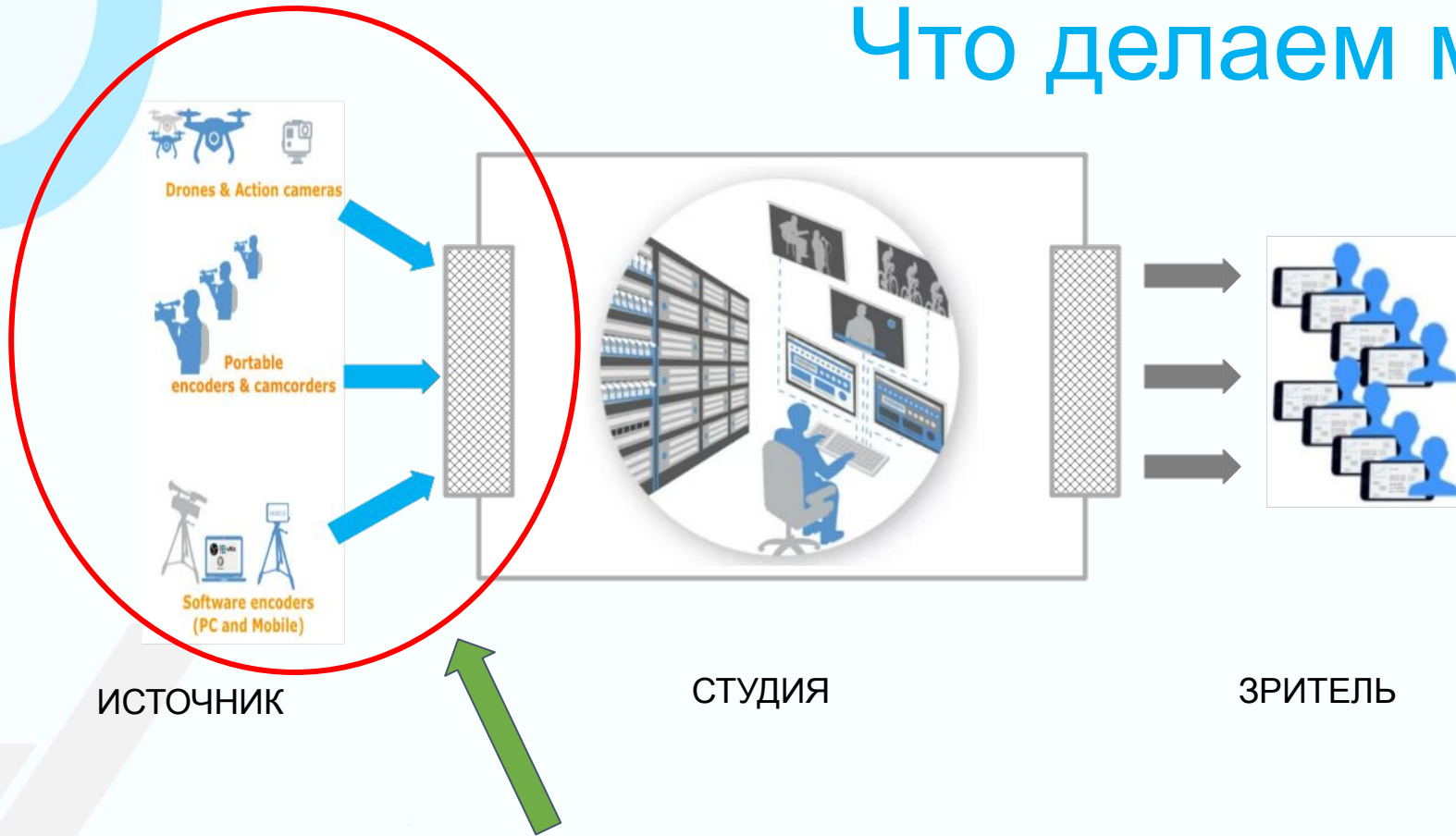
Пример 3: Тайм синхронизация



Пример 4: бондинг



Что делаем мы



ИСТОЧНИК

СТУДИЯ

ЗРИТЕЛЬ



ПРЕДПОСЫЛКИ К ПОЯВЛЕНИЮ SRT



// не много истории

Расклад в начале 2000

Desktop, Flash, RTMP



ИСТОЧНИК

СТУДИЯ

ЗРИТЕЛЬ

Расклад на недавний момент

Mobile, HLS, RTMP



Претензии к RTMP

* проприетарный. игнор разработчиков.
Adobe его не развивает.

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

* базируется на TCP

SRT: ответ от Haivision

- * Open-source протокол. Ориентация на коммунити разработчиков. Активное развитие.
- * Codec-agnostic: можно передавать что угодно
- * Базируется на UDP

SRT: Codec Agnostic



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

SRT построен на UDP ?!!!

Какого чёрта....

TCP



UDP



Типовые возражения:

- UDP это потеря данных
- UDP блокируется провайдерами

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

UDP - залог скорости

TCP



UDP*

* протоколы на базе UDP



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

Позволяют добиться значительно большей скорости передачи данных без потери

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.

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_INVALID, 0);  
  
    const int st = srt_bind(u, local_name, local_namelen);  
    if (st != 0)  
        return st;  
  
    return srt_connect(u, remote_name, remote_namelen);  
}
```


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

Немного деталей

- * Режимы соединения
- * Режимы передачи данных (Buffer, Message, Live)

Детали: режимы соединения

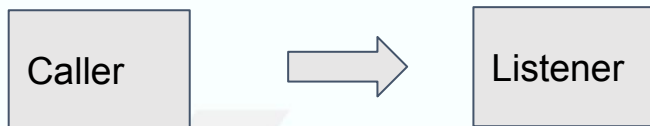
- * Caller & Listener
- * режим Рандеву (rendezvous)

Детали: режимы соединения

Listener - слушает UDP-порт

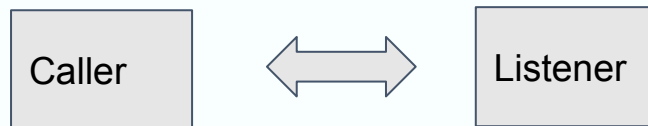
Caller - инициирует коннект на IP:PORT Listener'a

Направление данных может быть любым



1) Соединение
(нужно знать IP:PORT листенера)

Нужен "белый IP"



2) Передача данных

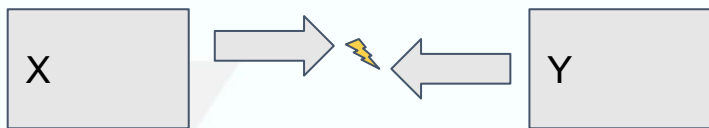
Детали: режимы соединения

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

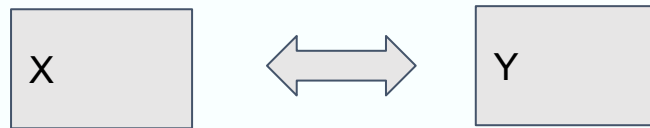
базируется на старом методе "UDP Hole punching".

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.

Немного деталей

- * Режимы соединения
- * Режимы передачи данных (Buffer, Message, Live)

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) );
```

режим Buffer

- `SRT0_TSBPDMODE` = false
- `SRT0_RCVLATENCY` = 0
- `SRT0_PEERLATENCY` = 0
- `SRT0_TLPKTDROP` = false
- `SRT0_LINGER` = 180 s
- `SRT0_MESSAGEAPI` = false
- `SRT0_NAKREPORT` = false
- `SRT0_RETRANSMIT_ALGO` = 0
- `SRT0_PAYLOADSIZE` = 0
- `SRT0_CONGESTION` = "file"

Аналог TCP.

Посланные 10.000
могут придти по частям.

Но придут все и в
нужном порядке.

`srt_send` / `srt_recv`

`srt_sendfile` / `srt_recvfile`

режим Message

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

Сообщение: всё или ничего.

`srt_sendmsg2` / `srt_recvmsg2`

```
typedef struct SRT_MsgCtrl_  
{  
    int flags;  
    int msgttl;  
    int inorder;  
    int boundary;  
    int64_t srctime;  
    int32_t pktseq;  
    int32_t msgno;  
    SRT_SOCKETGROUPDATA* grpdata;  
    size_t grpdata_size;  
} SRT_MSGCTRL;
```

режим Live

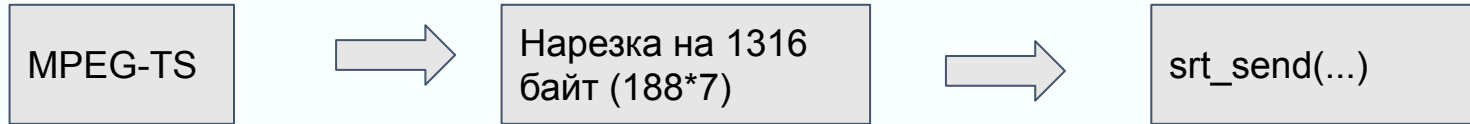
- `SRTO_TSBPDMODE` = true
- `SRTO_RCVLATENCY` = 120
- `SRTO_PEERLATENCY` = 0
- `SRTO_TLPKTDROP` = true
- `SRTO_LINGER` = 0
- `SRTO_MESSAGEAPI` = true
- `SRTO_NAKREPORT` = true
- `SRTO_RETRANSMITALGO` = 1
- `SRTO_PAYLOADSIZE` = 1316
- `SRTO_CONGESTION` = "live"



режим Live

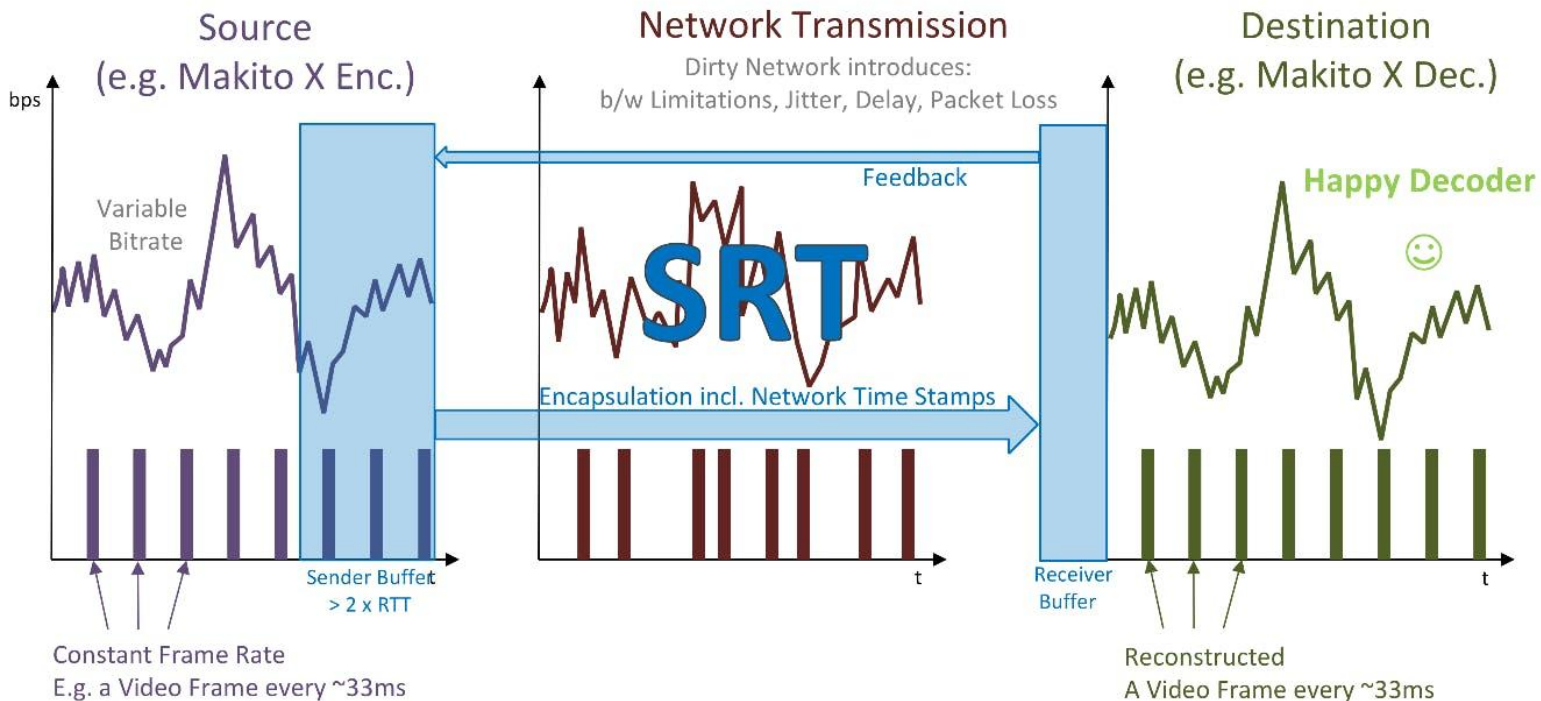
Live - для видео-стриминга

Общий Workflow:



режим Live

Live - самый медленный из-за стабилизирующего буфера.



Немного деталей

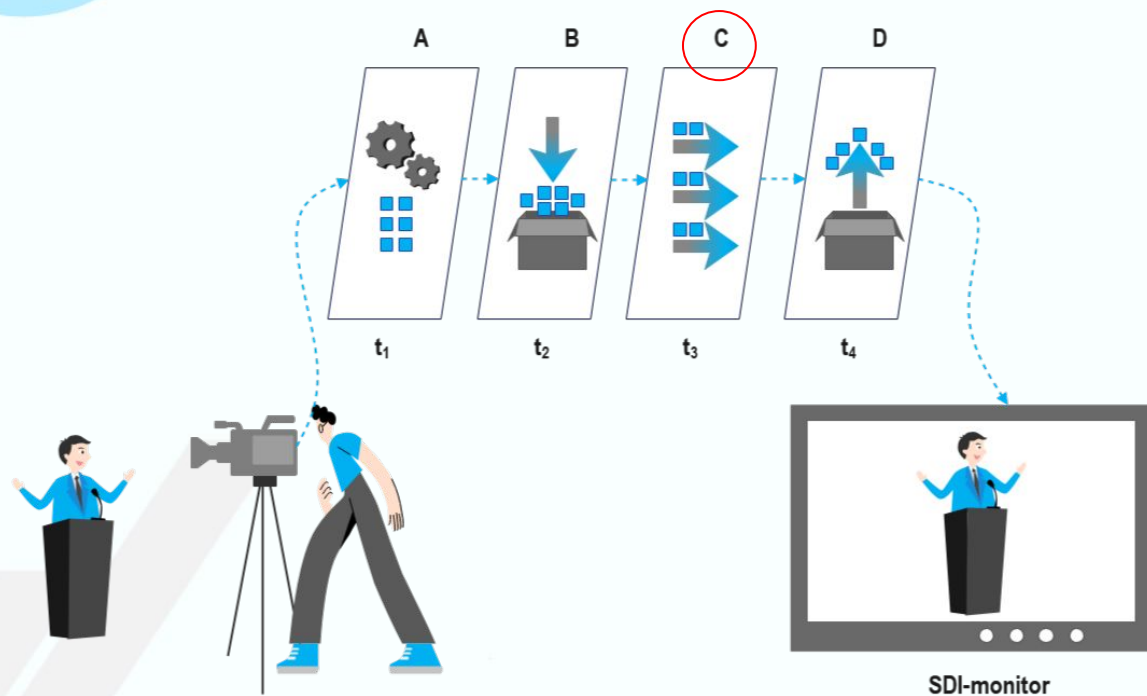
* Режимы соединения
// Caller, Listener, Rendezvous

* Режимы передачи данных
// Buffer, Message, Live

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)

SRT: что там с задержкой



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

SRT: что там с задержкой

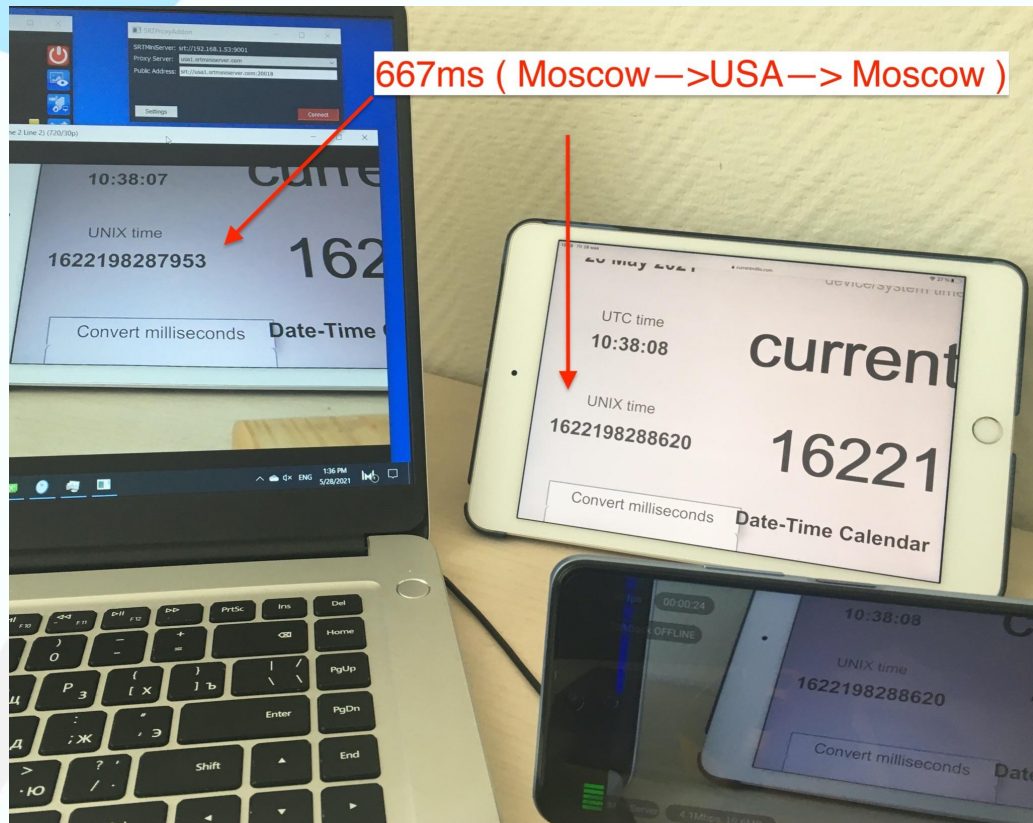


Схема:
Larix по LTE (Megafon)
стримит на SRT
MiniServer (офисный
провайдер) через USA-
прокси.

Задержка: 667 ms

SRT: что там с задержкой

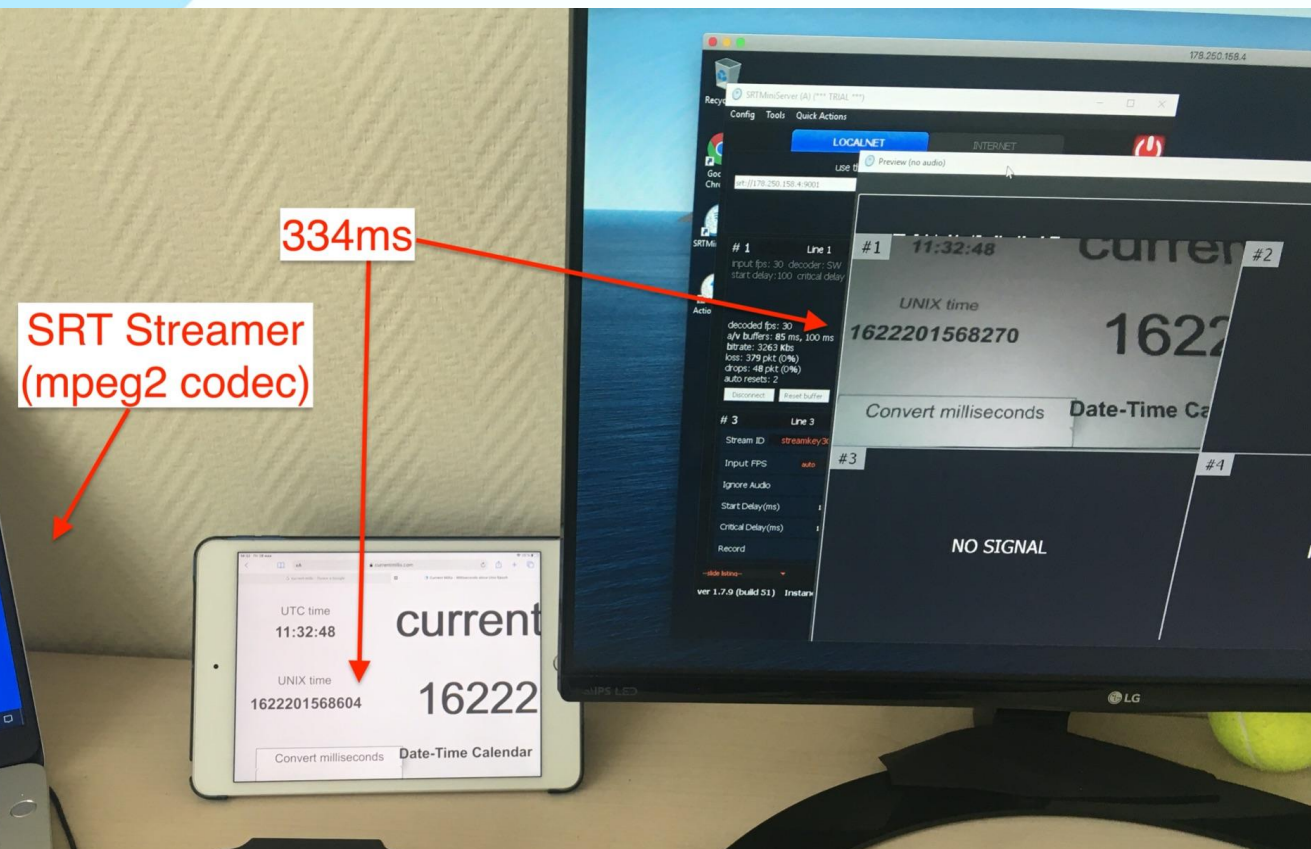


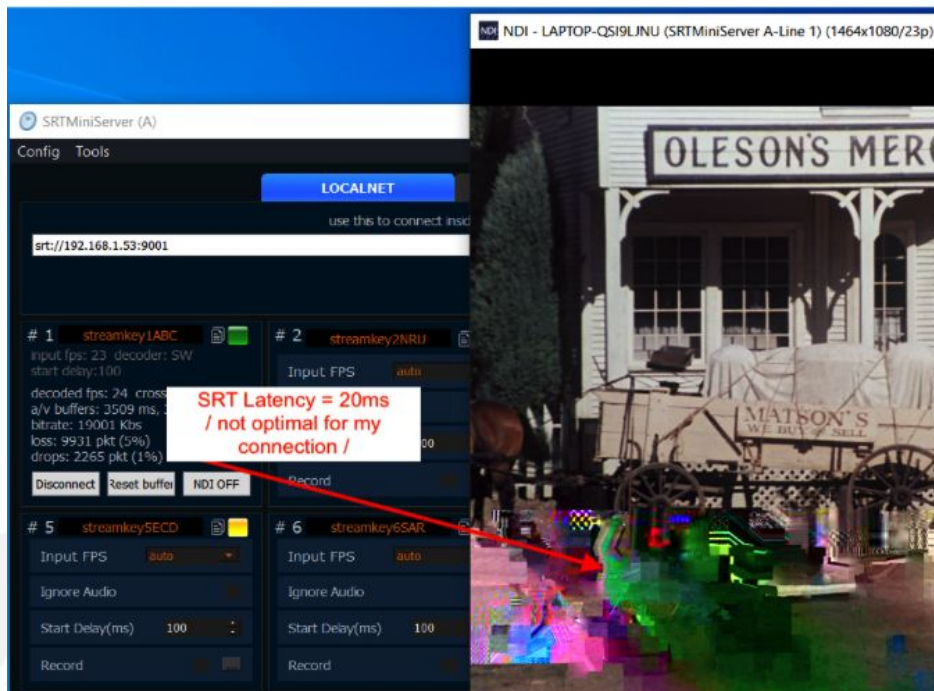
Схема:

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

Задержка: 334ms

Про “квадратики” (скорость vs качество)

SRT Latency 20 ms



NDI - LAPTOP-QS9LJINU (SRTMiniServer A-Line 1) (1464x1080/23p)

SRTMiniServer (A)

Config Tools

LOCALNET

use this to connect inside

srt://192.168.1.53:9001

1 streamkey1ABC
input fps: 23 decoder: SW
start delay: 100
decoded fps: 24 cross
a/v buffers: 3509 ms
bitrate: 19001 Kbs
loss: 9931 pkt (5%)
drops: 2265 pkt (1%)

2 streamkey2NRU
Input FPS auto

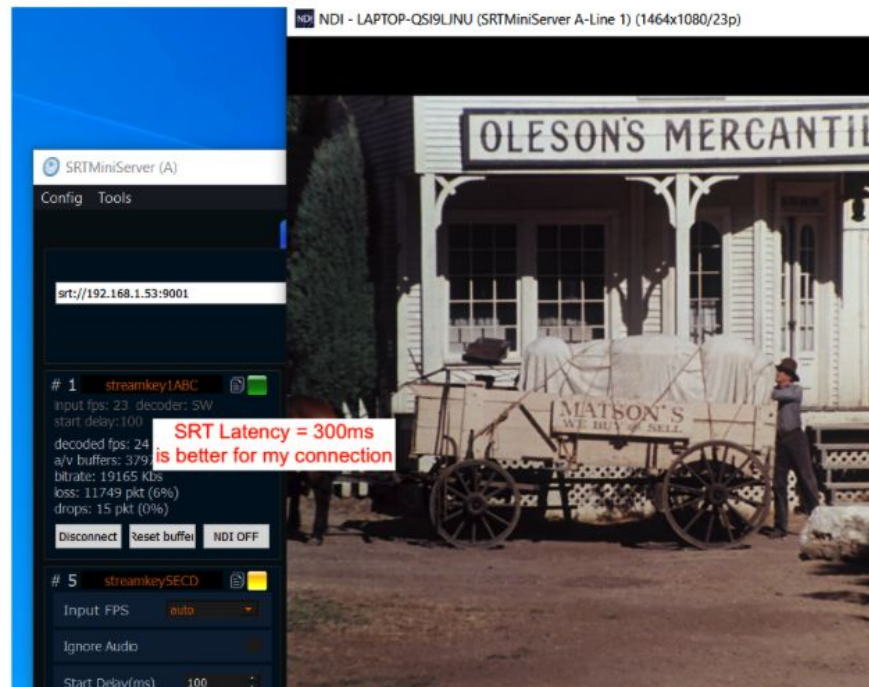
5 streamkey5ECD
Input FPS auto
Ignore Audio
Start Delay(ms) 100
Record

6 streamkey6SAR
Input FPS auto
Ignore Audio
Start Delay(ms) 100
Record

SRT Latency = 20ms / not optimal for my connection /

The video feed shows a scene with a building labeled "OLESON'S MERCANTILE" and a cart labeled "MATSON'S WE BUY & SELL". The feed is heavily distorted with colorful artifacts, and a red arrow points from the latency text to the distorted area.

SRT Latency 300 ms



NDI - LAPTOP-QS9LJINU (SRTMiniServer A-Line 1) (1464x1080/23p)

SRTMiniServer (A)

Config Tools

srt://192.168.1.53:9001

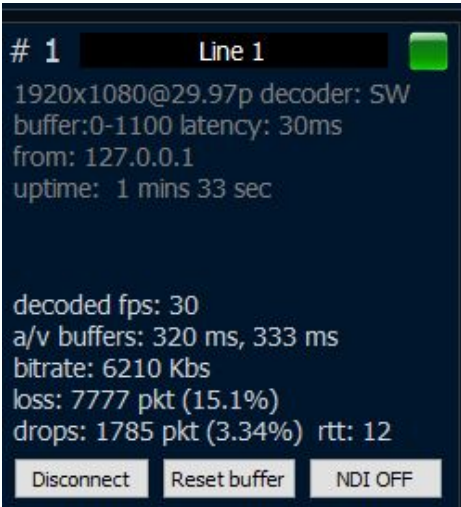
1 streamkey1ABC
input fps: 23 decoder: SW
start delay: 100
decoded fps: 24
a/v buffers: 379
bitrate: 19165 Kbs
loss: 11749 pkt (6%)
drops: 15 pkt (0%)

5 streamkey5ECD
Input FPS auto
Ignore Audio
Start Delay(ms) 100

SRT Latency = 300ms is better for my connection

The video feed shows the same scene as the left image, but it is clear and free of artifacts.

Про “квадратики”



The screenshot displays the following information:

- # 1 Line 1** (with a green status indicator)
- 1920x1080@29.97p decoder: SW
- buffer:0-1100 latency: 30ms
- from: 127.0.0.1
- uptime: 1 mins 33 sec
- decoded fps: 30
- a/v buffers: 320 ms, 333 ms
- bitrate: 6210 Kbs
- loss: 7777 pkt (15.1%)
- drops: 1785 pkt (3.34%) rtt: 12

At the bottom, there are three buttons: **Disconnect**, **Reset buffer**, and **NDI OFF**.

Annotations with arrows point to specific parts of the screenshot:

- Loss** and **Drops** point to the lines: `loss: 7777 pkt (15.1%)` and `drops: 1785 pkt (3.34%)`.
- Latency** points to the line: `latency: 30ms`.
- RTT** points to the line: `rtt: 12`.

Про “квадратики”

Latency 30ms

```
# 1 Line 1
1920x1080@29.97p decoder: SW
buffer:0-1100 latency: 30ms
from: 127.0.0.1
uptime: 1 mins 33 sec

decoded fps: 30
a/v buffers: 320 ms, 333 ms
bitrate: 6210 Kbs
loss: 7777 pkt (15.1%)
drops: 1785 pkt (3.34%) rtt: 12

Disconnect Reset buffer NDI OFF
```

Latency 500ms

```
# 1 Line 1
1920x1080@29.97p decoder: SW
buffer:0-1100 latency: 500ms
from: 127.0.0.1
uptime: 1 mins 3 sec

decoded fps: 30
a/v buffers: 42 ms, 33 ms
bitrate: 4750 Kbs
loss: 5611 pkt (16%)
drops: 0 pkt (0%) rtt: 34
auto resets: 2

Disconnect Reset buffer NDI OFF
```

Loss - ПОЧТИ ОДИНАКОВЫЙ
Drops - РАЗНЫЙ

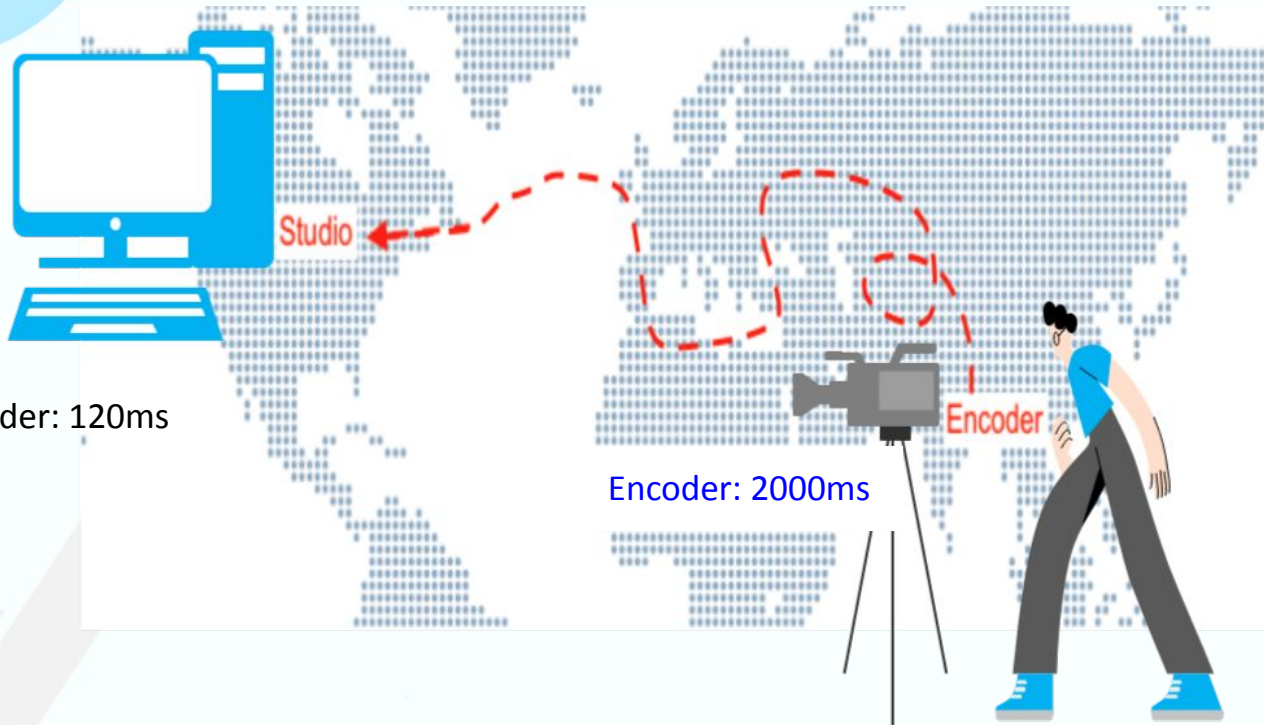
"Lost" (потерянные) означают больные пакеты. Это показатель “качества” канала.

"Drops"(отбрасывание) означает мертвые пакеты.

"Latency"(задержка) — это время лечения каждого больного пакета. Высокое значение означает больше шансов восстановить больной пакет. $3 \times RTT$

Про “квадратики”

Где ставить Latency?



Популярные вопросы

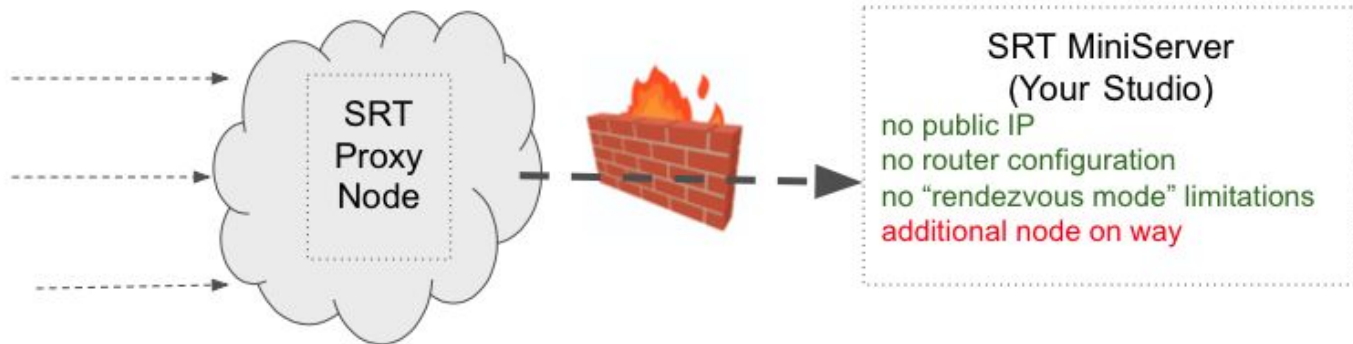
- Что там с реальной задержкой от стекла до стекла.
- SRT и Firewall. Ура, революция?
- SRT vs NDI
- SRT и Бондинг (свой, поверх MPTCP)

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.

SRT и Firewall (Plan B)

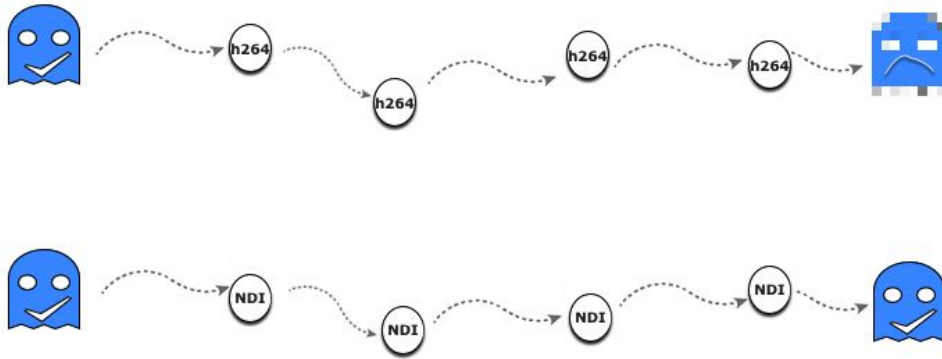


Популярные вопросы

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

SRT vs NDI

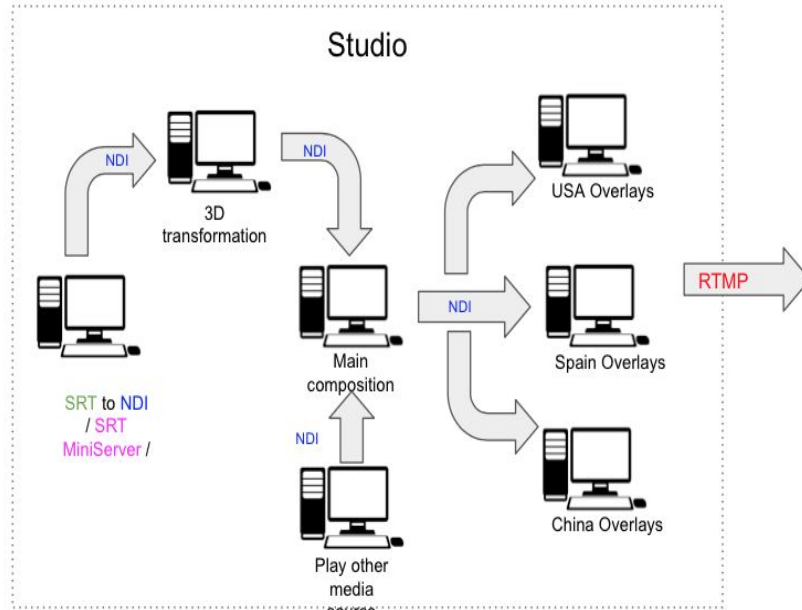
What happens if we do many encode-decode....



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

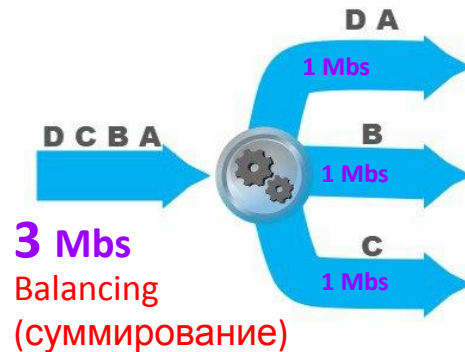
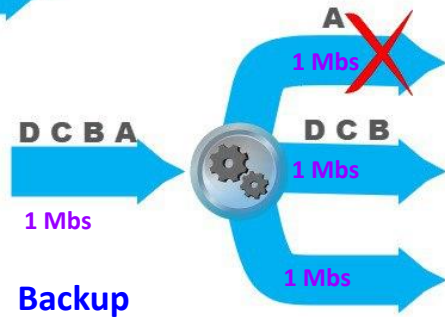
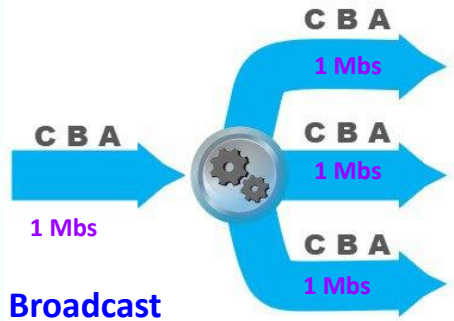
SRT vs NDI



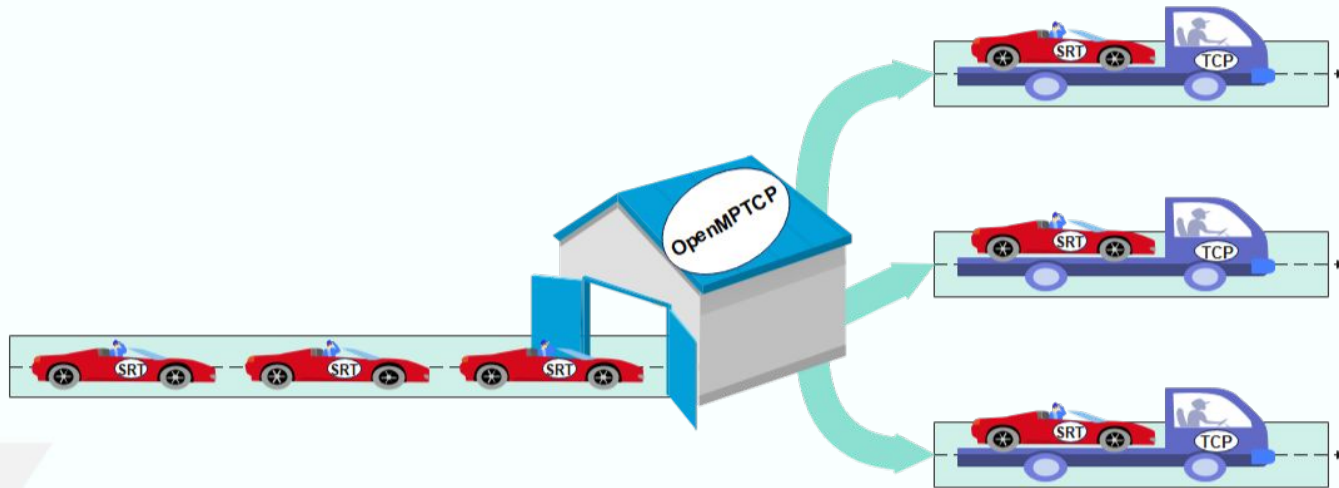
Популярные вопросы

- 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)

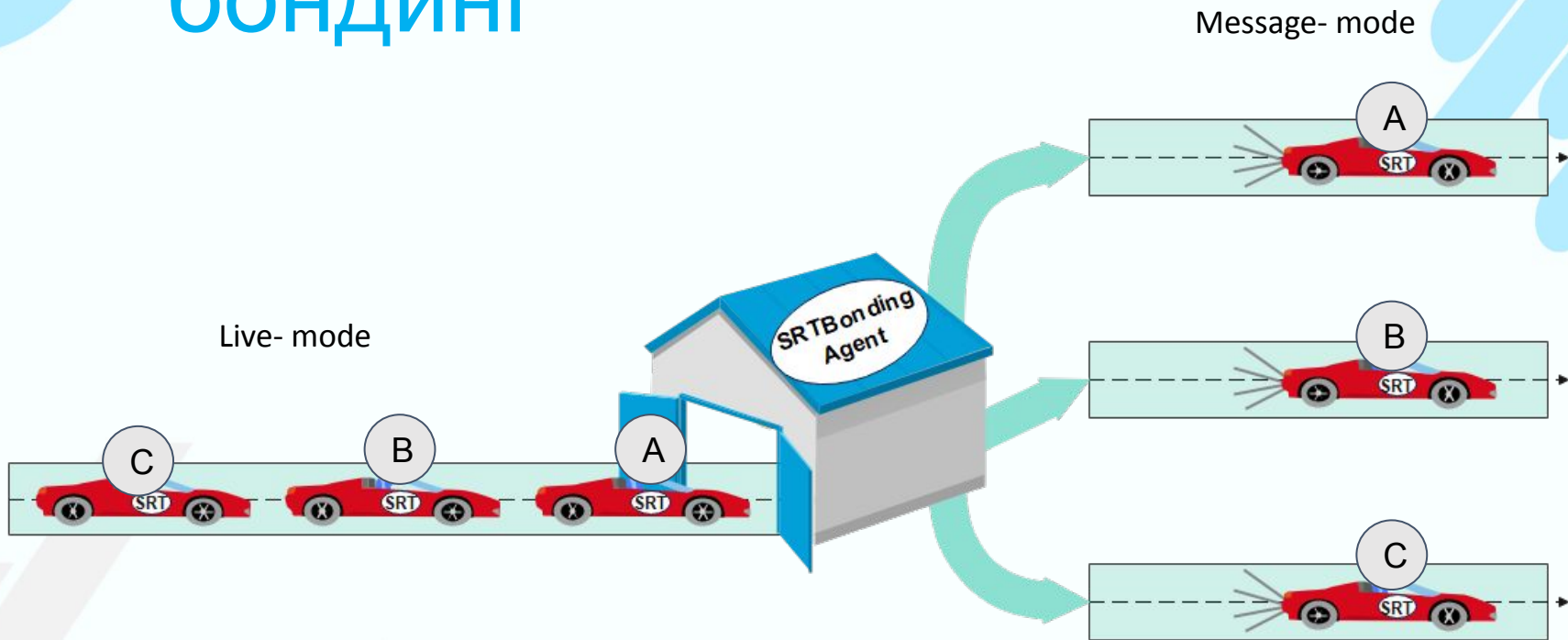
Нативный: SRT 1.5



SRT и MPTCP



Поэтому сделали свой бондинг



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)



WebRTC vs SRT

В стандарте WebRTC

- фиксированы кодеки
- громоздкий

SRT более лёгок;

как бы подстраивается в имеющиеся MPEG-TS потоки.



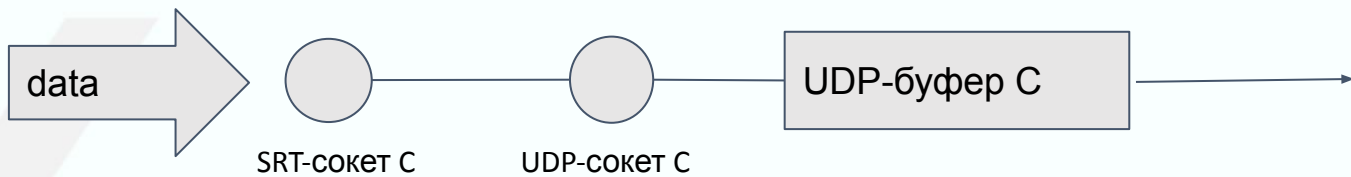
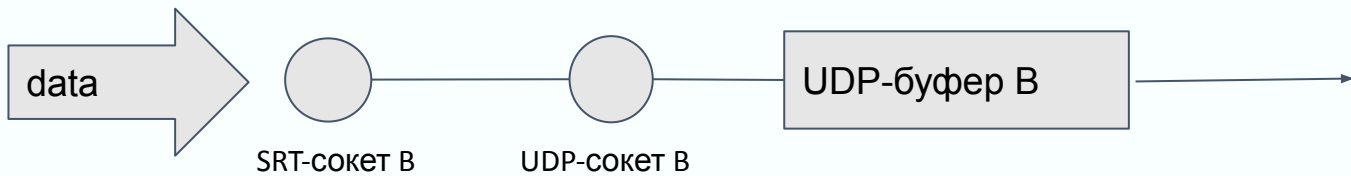
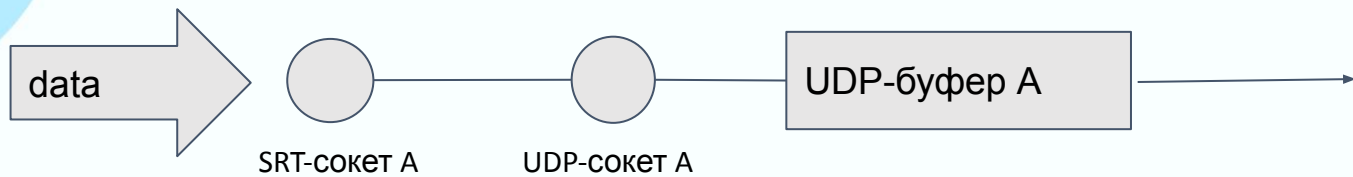
Грабли на которые наступили

На примере бондинг решения

Шаринг UDP-сокетов

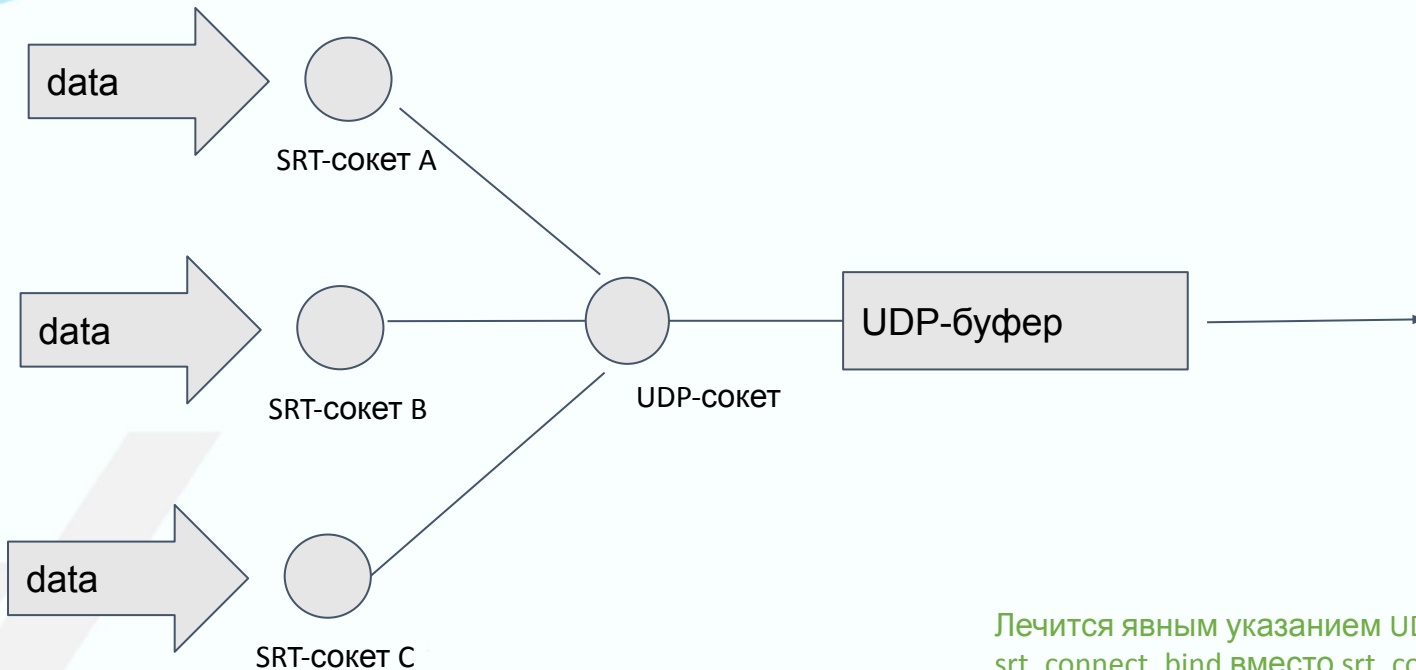


Предполагали что будет работать так:



Шаринг UDP-сокета

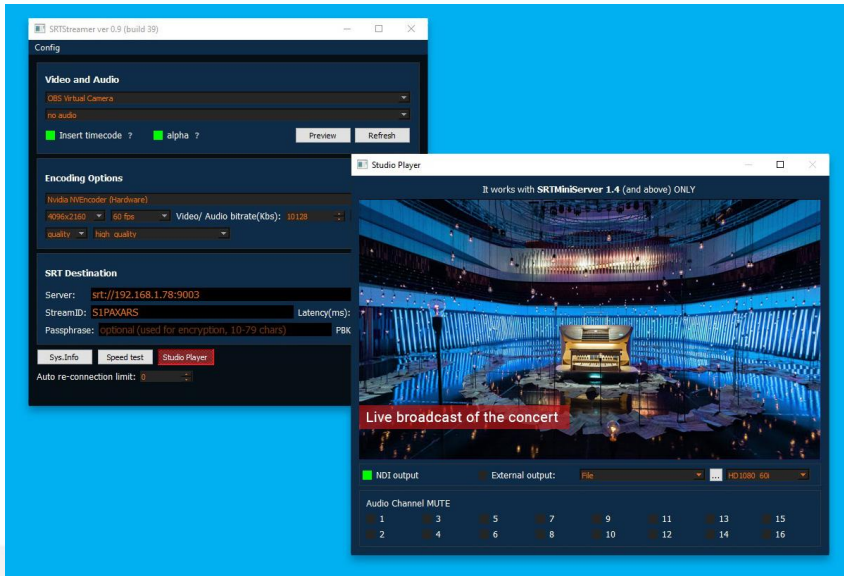
А работало так (по умолчанию):



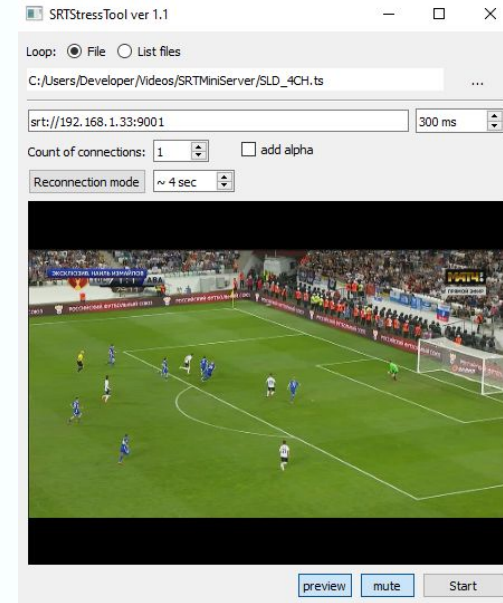
Лечится явным указанием UDP-порта
`srt_connect_bind` вместо `srt_connect`

Free утилиты

SRT Streamer



SRT Stress Tool





Спасибо за внимание!

SRTMiniServer.RU



Michael@GaraninApps.com