



Perfect Streamer

Выпуск 2.0.0.206

Perfect Soft

июл. 29, 2026

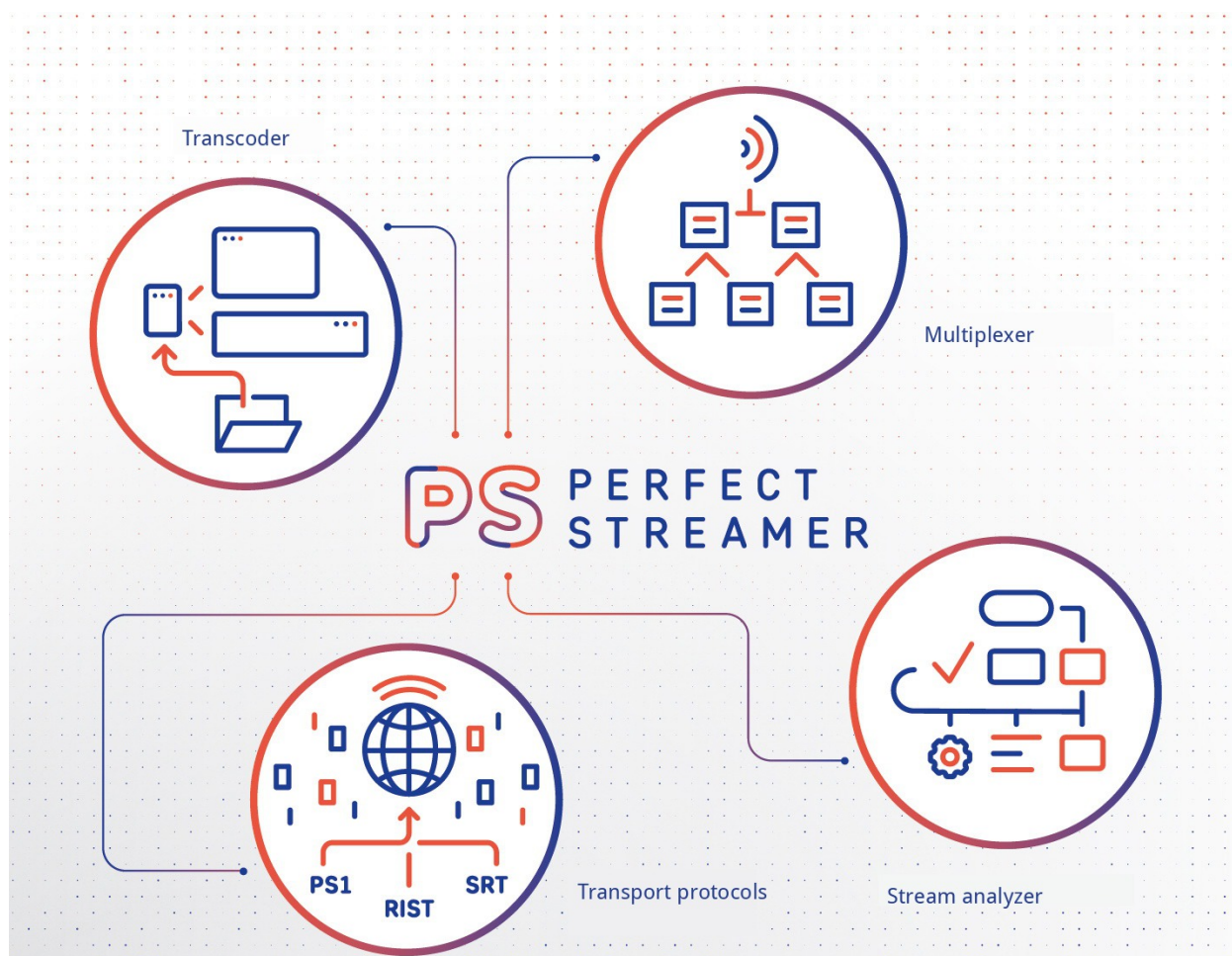
1	Perfect Streamer — руководство пользователя	1
1.1	Назначение	2
1.1.1	Передача потоков между узлами	3
1.1.2	Основные функции	3
1.1.3	Для DVB-вещателей	4
1.1.4	Для OTT-вещателей	5
1.2	Установка	5
1.2.1	Требования к системе	5
1.2.2	Установка на системы семейства RHEL	6
1.2.3	Установка на системы семейства Debian	6
1.2.4	Файлы и службы	7
1.2.5	Транскодеры	8
1.3	Первый старт и активация	13
1.3.1	Временная активация и запуск	13
1.3.2	Начальные настройки	14
1.3.3	Постоянная активация	14
1.3.4	При истечении срока годовой или пробной лицензии	15
1.4	Планирование и протоколы передачи данных	15
1.4.1	Модель передачи: Stream, Sender, Receiver, Peer	15
1.4.2	Peer-протоколы надёжной передачи	15
1.4.3	Выбор протокола передачи	17
1.4.4	Планирование полосы, задержки и портов	17
1.4.5	Шифрование потоков	18
1.4.6	Другие входы и выходы	18
1.4.7	Резервирование источников и раздача	18
1.4.8	Требования к входному потоку	19
1.5	Быстрый старт	19
1.5.1	Шаг 1. Создать поток	19
1.5.2	Шаг 2. Добавить вход	19
1.5.3	Шаг 3. Запустить и проверить	20
1.5.4	Шаг 4. Раздать поток	20
1.5.5	Что дальше	20
1.6	Meshwork	20
1.6.1	Ключевые понятия	21
1.7	Streamer: особенности реализации	31
1.7.1	SPTS-потоки	31

1.7.2	MPTS-потoki	37
1.7.3	Анализатор	39
1.7.4	Демультимплексор	42
1.7.5	Мультимплексор	43
1.7.6	Тестовые потоки	45
1.7.7	OTT и DVR	46
1.7.8	Транскодеры	51
1.7.9	DVB-приёмник	54
1.7.10	Публикация и приём RTMP	57
1.7.11	Прочие входы и выходы	58
1.7.12	EPG/EIT	60
1.8	Веб-интерфейс	62
1.8.1	Доступ и роли	62
1.8.2	Устройство интерфейса	63
1.8.3	Как работает контекстная справка	64
1.9	Дополнительные материалы	100
1.9.1	Кеширование и CDN для OTT-раздачи	100
1.9.2	Подключение внешних систем мониторинга	111
1.9.3	Оптимизация работы узла	116
1.9.4	Транскодер Intel VPL: поддерживаемое оборудование	117
1.10	FAQ	120
1.10.1	SRT: авторизация по логину и паролю в стороннем ПО	120
1.10.2	Рекомендации по работе с UDP-мультикастом	121
1.10.3	Рекомендации по настройке сети для мультикаста	121
1.10.4	Flussonic и SRT	122
1.11	Инструменты	122
1.11.1	TS Analyze Perfect Streamer Toolkit v2.3 — TR 101 290	122
1.11.2	MPTS Migrate Perfect Streamer Toolkit v1.1 — миграция идентичности MPTS	130
1.12	Справочные материалы	137
1.13	История версий	137
1.13.1	версия 2.0.0.206 Beta	137
1.14	Roadmap	141
1.14.1	Дескремблирование отдельных потоков через CAM	142
1.14.2	Промышленные системы DRM для OTT-раздачи	142
1.14.3	Сигнализация точек врезки рекламы в OTT-раздаче	142

ГЛАВА 1

Perfect Streamer — руководство пользователя

1.1 Назначение



Perfect Streamer — серверное программное обеспечение для надёжной доставки потоков MPEG-TS через публичную сеть Интернет с потерями пакетов и задержками, а также для приёма, обработки, мультиплексирования, транскодирования и OTT-раздачи телевизионных каналов. Продукт ориентирован на вещателей и операторов связи: магистральная передача каналов между операторами, головные станции DVB-to-IP, IPTV/OTT-платформы и сети распространения.

Perfect Streamer поставляется как единое приложение для Linux x86-64 со встроенным веб-интерфейсом и HTTP API. Одна установка выполняет все этапы тракта: приём, обработку и ремультимплексирование, транскодирование, передачу между узлами, OTT-раздачу и архив (DVR), мониторинг и многое другое.

1.1.1 Передача потоков между узлами

Для каждого потока настраивается сервер-передатчик (**Sender**) и один или несколько приёмников (**Receiver**). Для передачи потоков между передатчиком и приёмником, а также для связи со сторонним оборудованием доступны четыре **Peer**-протокола:

- **PS1** (Perfect Stream) — протокол собственной разработки на базе UDP, работает по принципу Automatic Repeat reQuest (ARQ) с выборочной ретрансмиссией. Низкая ресурсоёмкость; передача потоков высокого битрейта, включая полноразмерный MPTS. Особенно эффективен в конфигурациях точка-многоточка, один отправитель и много получателей.
- **SRT** — открытый протокол, имеет широкое распространение и хорошие характеристики компенсации потерь пакетов; режимы listener/caller; совместим со сторонним оборудованием и облачными сервисами.
- **RIST** — открытый протокол на базе RTP/RTCP. Работает по принципу ARQ без ACK, только NACK, что обеспечивает высокую эффективность; профили Simple и Main.
- **Pro-MPEG / RTP+FEC** (Pro-MPEG COP3, он же SMPTE 2022-1/2) — RTP с упреждающей коррекцией ошибок (FEC) без обратного канала. Достоинства — низкая задержка и совместимость с профессиональным вещательным оборудованием; недостатки — постоянный дополнительный трафик и слабое восстановление при больших потерях пакетов.

Все Peer-протоколы поддерживают шифрование AES. Подробное описание протоколов и планирование каналов передачи — в разделе [Планирование и протоколы передачи данных](#).

Помимо Peer-протоколов поддерживаются стандартные входы и выходы: UDP (unicast/multicast), RTP, HLS/HTTP, RTMP/RTMPS, файл/устройство, именованный канал (pipe) и внешнее приложение (std); дополнительно только в качестве входов — TCP и RTSP.

1.1.2 Основные функции

- [Обработка без перекодирования](#) — списки разрешённых и запрещённых PID, переназначение PID и автоматическая раскладка PID программы, правка SDT и языков аудио, смена PNR, удаление ненужных таблиц (SDT/CAT/NIT/TDT, телетекст, субтитры), приведение интервалов PAT/PMT к TR 101 290; режимы битрейта, включая CBR со стаффингом до DVB-совместимого постоянного битрейта.
- [Мультиплексор MPTS](#) — сборка мультиплекса из отдельных программ с генерацией PSI/SI и выходом, соответствующим TR 101 290.
- [Демультимплексор](#) — извлечение отдельных программ из MPTS в самостоятельные SPTS.
- [Транскодер](#) — аппаратное транскодирование на GPU NVIDIA и Intel VPL, программное на CPU; несколько энкодеров с одного декодера (например, для адаптивного битрейта).

- **Анализатор** — контроль соответствия TR 101 290 (DVB) для SPTS и MPTS, детектор дрейфа PCR, модель буфера T-STD, глубокий анализ потока, анализатор SCTE-35, детектор CBR/VBR.
- **EPG/EIT** — захват EIT с эфира в базу EPG, импорт XMLTV, генерация корректного EIT (present/following и schedule) в исходящий поток, форс-генерация SDT.
- **EPG-сервер** — отдача полного XMLTV на /xmltv с ограничением набора каналов по логину, для OTT middleware.
- **Тестовые потоки** — встроенный генератор тестового сигнала.
- **Meshwork** — объединение серверов в домены на основе gossip-протокола децентрализованных сетей: живой статус узлов, общий каталог используемых адресов и портов (библиотека ресурсов), репликация алертов по домену.
- Учёт сессий и контроль доступа — лимиты соединений на пользователя, ACL по каналам, prefix/suffix-логин для интеграции с middleware. Авторизация сессий клиентов во внешнем биллинге — проверка каждой сессии при подключении и периодическая реавторизация с отзывом доступа посреди сессии (жизненный цикл в стиле RADIUS: Start / Interim / Stop); универсальный HTTP-коннектор для подключения биллинга оператора. Учёт сделан отдельно по каждому протоколу — HLS/DASH для OTT и PS1/SRT для пиринга потоков.
- **Веб-интерфейс** — управление всеми функциями, роли администраторов, адаптивные профили для рабочей станции, планшета и телефона.
- Автоматические TLS-сертификаты (ACME) — автовыпуск и обновление сертификатов, горячая перезагрузка TLS; HTTPS и HTTP/3 «из коробки».
- **Анализатор** и мониторинг — контроль TR 101 290 (DVB) на входе, единый алертинг с доставкой по Email, Telegram или во внешний скрипт, мозаика каналов.

Также доступны: резервирование источников с автопереключением; интеграция с системами мониторинга (Zabbix, Grafana, Prometheus, InfluxDB, Nagios); интеграция с внешним биллингом для авторизации сессий клиентов; резервное копирование и экспорт/импорт настроек.

1.1.3 Для DVB-вещателей

Головным станциям спутникового, кабельного и эфирного вещания (DVB-to-IP, ремультимплекс, контрибуция между площадками) доступны:

- **Приём с DVB-карт** — DVB-S, DVB-S2, DVB-T, DVB-T2 и DVB-C с полным набором параметров настройки; управление LNB и DiSEqC 1.0, разделение LNB между тюнерами.
- Сканер транспондеров и «слепое» сканирование — обход reference-списков (satellites.xml, cables.xml, terrestrial.xml) со сбором PSI/SI, построением структуры мультимплекс → программа и измерением SNR, уровня и BER; поиск нестандартных транспондеров без reference-файла.
- Дескремблинг — аппаратный CI/CAM (EN 50221) с CAM, совмещённым с DVB-приёмником, и программный BISS-1 / BISS-E (ключи по PNR или PLP, горячее обновление). Очистка условного доступа на выходе — удаление ECM/EMM, CAT и CA-дескрипторов из PMT для получения чистого FTA-потока.
- Декапсуляция T2-MI (ETSI TS 102 773) — одновременная отдача внешнего DVB-S/S2-мультимплекса и всех вложенных T2-MI-несущих: приём регионального DVB-T2 по спутнику без отдельного шлюза.
- Монитор сигнала (femon) — непрерывное измерение lock, уровня, SNR, BER и UCB с построением графиков во времени.

- **Демультимплексор** — извлечение отдельной программы (по PNR) из принятого транспондера в самостоятельный SPTS.

1.1.4 Для OTT-вещателей

IPTV/OTT-операторам и платформам распространения доступны:

- **OTT-доставка** — HLS, Low-Latency HLS и MPEG-DASH на CMAF; доставка по HTTPS и HTTP/3 (QUIC); адаптивный мультибитрейт (ABR) в едином master-плейлисте; GOP/IDR-выровненная сегментация; сырой MPEG-TS over HTTP для TS-клиентов.
- Субтитры WebVTT — декодирование DVB-телетекста и DVB-субтитров в WebVTT для OTT-плееров, в том числе в архиве.
- **Сетевой DVR** — запись каждого OTT-канала в дисковый архив параллельно с раздачей; воспроизведение архива по тем же HLS/DASH-URL (таймшифт и VOD), бесшовный переход архив → live, catch-up по EPG, адаптивный VOD и CMAF-архив, многодисковые хранилища с ретеншеном и автоматической очисткой.
- **Транскодер и ABR-лесенка** — один декодер питает несколько энкодеров с собственными разрешениями, битрейтом и кодеком (1toN); аппаратно на GPU (NVIDIA, Intel VPL) или программно на CPU.
- Готовность к CDN — контент-адресуемые сегменты с immutable-заголовками, нормализованный cache-key и CORS для масштабирования за обратным прокси или CDN.

1.2 Установка

Perfect Streamer устанавливается из пакетов; для семейств RHEL и Debian доступны репозитории. После установки *произведите активацию и начальную настройку сервиса*.

1.2.1 Требования к системе

- **Perfect Streamer** работает на ОС Linux. Главное требование — версия GLIBC ≥ 2.17 .
- Сетевые интерфейсы, с которыми работает сервис стримера, должны иметь статические настройки.
- Скрипты инсталлятора используют утилиту **sudo**, т.е. она должна быть установлена в системе.

Для семейств Red Hat и Debian имеются пакеты инсталляции и репозитории. Поддерживается версия RHEL 7 и выше (CentOS и др.), а также RPM-совместимые дистрибутивы — например, openSUSE Leap (см. замечание в конце раздела *Установка на системы семейства RHEL*). Системы на базе Debian (Ubuntu и др.) должны иметь службу systemd.

Ориентировочные требования к аппаратному обеспечению: 1 ядро 2.4 ГГц и 1 ГБ ОЗУ на каждые 200 Мбит/с трафика. Оценка приблизительная и зависит от используемых протоколов и настроек сервиса.

1.2.2 Установка на системы семейства RHEL

Установить репозиторий для RHEL 7:

```
$ sudo yum install yum-utils
$ sudo yum-config-manager --add-repo=http://repo.pstreamer.tv/pub/pstreamer/pstreamer.
↪ repo
```

Или для RHEL 8+:

```
$ sudo yum config-manager --add-repo=http://repo.pstreamer.tv/pub/pstreamer/pstreamer.
↪ repo
```

Установить пакет:

```
$ sudo yum -y install pstreamer
```

Обновить пакет:

```
$ sudo yum -y update pstreamer
```

Удалить все пакеты:

```
$ sudo yum -y remove pstreamer aksusbd
```

Примечание: **openSUSE** (Leap, последняя стабильная версия) — система на базе RPM. Применяется тот же репозиторий, что и для RHEL, но операции выполняются пакетным менеджером **zypper**:

```
$ sudo zypper addrepo http://repo.pstreamer.tv/pub/pstreamer/pstreamer.repo
$ sudo zypper --gpg-auto-import-keys refresh
$ sudo zypper install pstreamer
```

Обновление — `sudo zypper update pstreamer`, удаление — `sudo zypper remove pstreamer aksusbd`.

1.2.3 Установка на системы семейства Debian

Установить репозиторий:

```
$ sudo wget http://repo.pstreamer.tv/pub/deb/dists/pstreamer/pstreamer.list -O /etc/
↪ apt/sources.list.d/pstreamer.list
$ sudo apt-get update
```

Установить пакет:

```
$ sudo apt-get install pstreamer
```

Обновить пакет:

```
$ sudo apt install pstreamer
```

Удалить все пакеты:

```
$ sudo apt-get remove pstreamer aksusbd
```

1.2.4 Файлы и службы

/usr/local/bin/pss

Исполняемый файл.

/opt/pss/config/pss.properties

Глобальные настройки, логи, пути к папкам и др. При внесении изменений перезапустить сервис.

/opt/pss/config/pss.json

Основной файл настроек. Создаётся и обновляется автоматически. При старте сервис пытается загрузить именно его.

/opt/pss/config/pss_back.json

Резервная копия предыдущей рабочей конфигурации. Сохраняется автоматически при действии **Восстановить настройки** в веб-интерфейсе и используется как первый запасной вариант, если основной *pss.json* не удалось разобрать.

/opt/pss/config/pss_default.json

Файл настроек по умолчанию. Поставляется вместе с пакетом и применяется как последний запасной вариант, если ни основной *pss.json*, ни *pss_back.json* не удалось загрузить. Из него же создаётся рабочий *pss.json* при самом первом запуске: файл задаёт порт веб-интерфейса 8808 и учётную запись *admin / admin*.

/opt/pss/config/bad/

Архив повреждённых файлов *pss.json*. Если при старте основной файл настроек не удалось разобрать, он перемещается сюда с именем вида *pss_YYYYMMDD_HHMMSS.json*. Каталог создаётся автоматически. Подробнее см. раздел [Поведение при старте и ошибках конфигурации](#).

/opt/pss/data

Папка размещения данных. Создаётся и обновляется автоматически. Может быть изменена в файле глобальных настроек.

/usr/lib/systemd/system/pss.service

systemd-файл службы.

/var/log/pss

Папка записи логов. Может быть изменена в файле глобальных настроек.

Имя службы — **pss**. Запускается от пользователя **pss**.

В процессе установки ставится сопутствующий пакет **aksusbd** от системы защиты; он включает службы **hasplmd** и **aksusbd**.

Поведение при старте и ошибках конфигурации

При запуске сервис последовательно пытается загрузить файлы настроек из папки `/opt/pss/config`:

1. `pss.json` — основной файл настроек.
2. `pss_back.json` — резервная копия предыдущей рабочей конфигурации.
3. `pss_default.json` — настройки по умолчанию, поставляемые вместе с пакетом.

Используется первый успешно загруженный файл. Если все три файла отсутствуют или повреждены, сервис стартует с пустыми настройками; в этом случае требуется ручная правка административного пароля и перезапуск.

Структурно повреждённый файл настроек. Если `pss.json` содержит синтаксическую ошибку JSON, неизвестные ключи, неверный тип значения или нарушение уникальности (например, два потока с одинаковым `id`), сервис помещает повреждённый файл в архив `/opt/pss/config/bad/` с именем `pss_YYYYMMDD_HHMMSS.json` (дата и время старта). После этого сервис продолжает попытки загрузки в обычном порядке и пересохраняет рабочую конфигурацию в `pss.json` из `pss_back.json` или `pss_default.json`. Подробности (имя ключа, описание ошибки, имя файла в архиве) записываются в журнал работы.

В архив попадает только основной `pss.json`. Файлы `pss_back.json` и `pss_default.json` при повреждении не архивируются — записей в журнале достаточно для диагностики, а сами файлы остаются на месте и могут быть исправлены вручную.

Если в момент следующей загрузки в `/opt/pss/config/bad/` уже существует файл с той же меткой времени (например, при быстрых перезапусках в одну и ту же секунду), он перезаписывается.

Числовые значения за пределами допустимого диапазона. Если в файле настроек встречается числовое значение меньше минимально допустимого или больше максимально допустимого для этого параметра, сервис не отбрасывает файл целиком. Вместо этого в журнал записывается предупреждение с указанием имени параметра, прочитанного значения и применённой границы, а само значение приводится к ближайшей допустимой границе диапазона (минимуму или максимуму). После завершения загрузки сервис автоматически пересохраняет `pss.json` с уже скорректированными значениями, поэтому при повторном запуске эти предупреждения уже не появляются.

Такое поведение применяется только при первоначальной загрузке файла настроек. При изменении настроек через веб-интерфейс или внешний API значения вне допустимого диапазона по-прежнему отклоняются с ошибкой — без автоматической коррекции.

1.2.5 Транскодеры

Установка транскодеров для Perfect Streamer.

Доступны пакеты:

- **pstreamer-tcsw** — транскодирование на CPU (Software).
- **pstreamer-tcnv** — транскодирование на GPU NVIDIA.
- **pstreamer-tcivpl** — транскодирование на GPU Intel (Intel VPL).

Главное требование — версия GLIBC ≥ 2.28 .

Транскодер работает на AlmaLinux 8.9. Транскодер Intel VPL работает на системах AlmaLinux 10 (RHEL 10) и Ubuntu 24.04.

Какое оборудование Intel подходит для транскодирования, какие рантаймы и кодеки доступны на разных поколениях графики и как проверить, что оборудование распознано, — в разделе [Транскодер Intel VPL: поддерживаемое оборудование](#).

RHEL 8+

1. Установить pstreamer.
2. Для NVIDIA установить репозитории и обновить систему (если они ещё не были установлены):

```
sudo dnf config-manager --add-repo https://developer.download.nvidia.com/compute/cuda/
↪repos/rhel9/x86_64/cuda-rhel9.repo
sudo dnf clean all
sudo dnf update -y
reboot
```

3. NVIDIA Encoder.

- Установить CUDA:

```
sudo dnf -y install cuda-toolkit-12-5
```

- Установить драйвер (выберите вариант):

Legacy

```
sudo dnf -y module install nvidia-driver:latest-dkms
```

New

```
sudo dnf -y module install nvidia-driver:open-dkms
```

После установки обязательно перезагрузить машину:

```
reboot
```

После перезагрузки проверить работу драйвера:

```
nvidia-smi
modprobe nvidia
sudo lsmod | grep nvidia
```

или

```
modprobe nouveau
sudo lsmod | grep nouveau
```

4. Intel VPL Encoder.

- Установка драйвера Intel и Intel VPL (RHEL 10):

```
dnf install -y intel-gpu-firmware
dnf install -y https://mirrors.rpmfusion.org/free/el/rpmfusion-free-release-$(rpm -E
↪%rhel).noarch.rpm https://mirrors.rpmfusion.org/nonfree/el/rpmfusion-nonfree-
↪release-$(rpm -E %rhel).noarch.rpm
dnf install -y intel-media-driver
dnf install -y intel-vpl-gpu-rt
```

5. Установить пакеты транскодера.

- CPU Software метод:

```
sudo dnf install -y pstreamer-tcsw
```

- NVIDIA GPU:

```
sudo dnf install -y pstreamer-tcnv
```

- Intel VPL GPU:

```
sudo dnf install -y pstreamer-tcivpl
```

Ubuntu 22/24

1. Установить pstreamer.

2. NVIDIA Encoder.

- Установить CUDA toolkit версии 12.5:

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-  
keyring_1.1-1_all.deb  
sudo dpkg -i cuda-keyring_1.1-1_all.deb  
sudo apt-get update  
sudo apt-get -y install cuda-toolkit-12-5
```

- Установить драйвер (выберите вариант):

legacy kernel module flavor:

```
sudo apt-get install -y cuda-drivers
```

или

open kernel module flavor:

```
sudo apt-get install -y nvidia-driver-555-open  
sudo apt-get install -y cuda-drivers-555
```

После установки обязательно перезагрузить машину:

```
reboot
```

После перезагрузки проверить работу драйвера:

```
nvidia-smi
```

Для работы транскодера требуется версия CUDA 12.5. В системе уже может быть установлена CUDA другой версии. Также при обновлении системы возможно обновление CUDA на более новую версию. Это не мешает работе, их можно не удалять, разные версии CUDA устанавливаются в отдельные папки.

3. Intel VPL Encoder (Ubuntu 24.04).

Ключевые компоненты Intel VPL берутся из официального репозитория Intel Graphics — в стандартных репозиториях Ubuntu они старше и не поддерживают современные GPU.

- Подключить репозиторий Intel Graphics:

```
sudo apt-get update
sudo apt-get install -y ca-certificates gnupg wget
wget -q0 - https://repositories.intel.com/gpu/intel-graphics.key | sudo gpg --yes --
↳dearmor --output /usr/share/keyrings/intel-graphics.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/intel-graphics.gpg] https://
↳repositories.intel.com/gpu/ubuntu noble unified" | sudo tee /etc/apt/sources.list.d/
↳intel-gpu-noble.list
sudo apt-get update
```

- Установить рантайм Intel VPL и драйвер:

```
sudo apt-get install -y libvpl2 libmfxgen1 intel-media-va-driver-non-free
```

- Дополнительно можно установить инструменты диагностики:

```
sudo apt-get install -y libvpl-tools
```

- Пользователь **pss**, от имени которого работает служба, должен входить в группы **render** и **video** (доступ к устройствам `/dev/dri/`):

```
sudo usermod -aG render,video pss
sudo systemctl restart pss
```

- Проверить, что GPU виден (при установленных `libvpl-tools`):

```
vainfo --display drm --device /dev/dri/renderD128
vpl-inspect
```

В выводе `vainfo` должна присутствовать строка «Driver version: Intel iHD ...», а `vpl-inspect` должен перечислять аппаратную реализацию (AccelerationMode VAAPL).

4. Установить пакеты транскодера.

- CPU Software метод:

```
sudo apt-get install -y pstreamer-tcsw
```

- NVIDIA GPU:

```
sudo apt-get install -y pstreamer-tcnv
```

- Intel VPL GPU (Ubuntu 24.04):

```
sudo apt-get install -y pstreamer-tcivpl
```

Файлы и проверка установки

Пакеты транскодеров установят файлы:

- `/usr/local/bin/tcsw` — исполняемый файл SW (CPU) транскодера.
- `/usr/local/bin/tcnv` — исполняемый файл NVIDIA (GPU) транскодера.
- `/usr/local/bin/tcivpl` — исполняемый файл Intel VPL (GPU) транскодера.
- `/opt/pss/config/pss_tc_sw.properties` — файл стартовой конфигурации для SW (CPU) транскодера.

- `/opt/pss/config/pss_tc_nv.properties` — файл стартовой конфигурации для NVIDIA (GPU) транскодера.
- `/opt/pss/config/pss_tc_ivpl.properties` — файл стартовой конфигурации для Intel VPL (GPU) транскодера.

Установка пакетов транскодера перезапустит службу **pss**. Проверить установку транскодеров можно в разделе *About* веб-интерфейса: будет отображена версия транскодеров или ошибка.

Другие ОС на основе Debian и RHEL

Для установки транскодера `pstreamer-tcnv` на ОС, отличные от Ubuntu 22/24 и RHEL 8+, используйте конфигуратор для подбора необходимой версии CUDA и драйвера на сайте NVIDIA:

<https://developer.nvidia.com/cuda-toolkit-archive>

При выборе каждой из версий CUDA доступна информация, для какой версии ОС она подходит. Поддерживаемая архитектура — `x86_64`. Если вам требуется более старая версия ОС, то проверьте её поддержку в более старых версиях CUDA. Главное требование к ОС — это поддержка версии GLIBC `>= 2.28`.

Драйвер NVIDIA необходимо устанавливать из репозитория, предлагаемого для версии вашей ОС на странице выбранной версии CUDA.

Поддержку видеокарт NVIDIA для функционала транскодера `pstreamer-tcnv` можно проверить на сайте NVIDIA в [Video Encode and Decode Support Matrix](#). На данной странице доступна матрица поддержки видеоформатов для декодера и энкодера, а также другие характеристики.

Удаление старой версии CUDA и драйвера

В случае установки некорректной версии CUDA и драйвера может потребоваться установить другую версию, предварительно удалив установленную.

<https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html?highlight=uninstall#removing-cuda-toolkit>

Удаление CUDA

RHEL:

```
dnf remove "cuda*" "*cublas*" "*cufft*" "*cufile*" "*curand*" "*cusolver*" "*cusparse*"
↳ "*gds-tools*" "*npp*" "*nvjpeg*" "nsight*" "*nvvm*" "*nvptx*"
```

Debian/Ubuntu:

```
apt remove --autoremove --purge "cuda*" "cublas*" "cufft*" "cufile*" "curand*"
↳ "cusolver*" "cusparse*" "gds-tools*" "npp*" "nvjpeg*" "nsight*" "nvvm*"
↳ "nvptx*"
```

Удаление драйвера

Ubuntu 22/24:

```
apt remove --autoremove --purge -V \
  cuda-compat\* \
  cuda-drivers\* \
  libnvidia-cfgl\* \
  libnvidia-compute\* \
  libnvidia-decode\* \
  libnvidia-encode\* \
  libnvidia-extra\* \
  libnvidia-fbc1\* \
  libnvidia-gl\* \
  libnvidia-gpucomp\* \
  libnvidia-nscq\* \
  libnvdm\* \
  libxnvctrl\* \
  nvidia-dkms\* \
  nvidia-driver\* \
  nvidia-fabricmanager\* \
  nvidia-firmware\* \
  nvidia-headless\* \
  nvidia-imex\* \
  nvidia-kernel\* \
  nvidia-modprobe\* \
  nvidia-open\* \
  nvidia-persistenced\* \
  nvidia-settings\* \
  nvidia-xconfig\* \
  xserver-xorg-video-nvidia\*
```

RHEL 9:

```
dnf module remove --all nvidia-driver
dnf module reset nvidia-driver
```

RHEL 10:

```
dnf remove nvidia-driver\*
```

1.3 Первый старт и активация

1.3.1 Временная активация и запуск

Временная версия — без ограничений по количеству потоков. После установки служба **pss** неактивна, так как для запуска требуется активация. Для временной активации запустить скрипт:

```
$ /opt/pss/tools/activate.sh
```

Служба **pss** будет запущена и настроена на автостарт. Можно проверить успешность запуска:

```
$ systemctl status pss
```


При проблемах см. логи:

```
$ tail -n 20 /var/log/pss/main.log
$ journalctl -u pss -n 20
```

1.3.2 Начальные настройки

Подключиться через браузер к веб-интерфейсу:

`http://host:8808`

Логин: `admin`

Пароль: `admin`

Доступны два веб-интерфейса: новый (основной) — по пути `/admin`, классический — по пути `/classic`.

Порты служб по умолчанию:

Служба	HTTP	HTTPS
Веб-интерфейс и HTTP API	8808	43981
ОТТ-раздача (HTTP-сервер)	43972	43982
EPG-сервер	43973	43983

Сразу после установки доступен только HTTP-порт веб-интерфейса — 8808. Он задан в поставляемом файле `pss_default.json`, из которого при первом запуске создаётся рабочий `pss.json`; других служб этот файл не описывает, поэтому остальные порты таблицы открываются только после настройки соответствующей службы, а HTTPS — после включения TLS и установки сертификата. Если ключ `web-server.bind-port` в настройках отсутствует, служба использует встроенное запасное значение 43971.

Обязательно сменить логин и пароль администратора — в списке администраторов веб-интерфейса. Учётным записям назначаются роли: **Admin** (полный доступ), **RestrictedAdmin** (просмотр и постановка потоков на паузу), **Viewer** (только просмотр).

Сменить, если надо, порт веб-интерфейса — в настройках веб-сервера. Изменение вступит в силу после перезапуска службы; перезапуск можно выполнить из веб-интерфейса.

Проверить данные активации можно в разделе *About* веб-интерфейса.

1.3.3 Постоянная активация

Система защиты поддерживает программные (SL) и USB (HL) ключи. Для постоянной активации:

1. В разделе *About* веб-интерфейса получить C2V-данные для запроса на активацию, выслать их поставщику.
2. Ввести полученные от поставщика V2C-данные из файла или из буфера обмена в том же разделе веб-интерфейса.
3. В том же разделе сверить данные активации.

Чтобы Perfect Streamer увидел USB-ключ при активной пробной лицензии, надо перезапустить службу **pss** — через функцию перезапуска в веб-интерфейсе или командой `sudo systemctl restart pss`. Если же срок пробной лицензии истечёт, служба сама переключится на постоянный USB-ключ.

1.3.4 При истечении срока годовой или пробной лицензии

При невозможности запуска службы **pss** ввести команду:

```
$ journalctl -u pss -n 20
```

Следующая ошибка означает, что срок лицензии Perfect Streamer закончился:

```
LDK Protection System: Feature has expired (H0041)
```

1.4 Планирование и протоколы передачи данных

Ядро **Perfect Streamer** — надёжная доставка потоков MPEG-TS между узлами через публичную сеть с потерями пакетов и задержками. В этом разделе описана модель передачи, транспортные протоколы и то, как спланировать канал: выбрать протокол, рассчитать полосу, задержку и порты. Детальные настройки отдельных полей вынесены в контекстное описание веб-интерфейса (*Веб-интерфейс*), а обработка потока на узле — в раздел *Streamer: особенности реализации*.

1.4.1 Модель передачи: Stream, Sender, Receiver, Peer

Единица работы — поток (**Stream**), один канал MPEG-TS. Для каждого потока настраивается сервер-передатчик (**Sender**) и один или несколько приёмников (**Receiver**); связка «передатчик — приёмник» обозначается **Peer**.

Настройка сводится к спискам входов (**input**) и выходов (**output**) для каждого потока:

- на передатчике **input** — источники MPEG-TS, **output** — передача на приёмники;
- на приёмнике **input** — приём потока от передатчика.

Несколько входов в списке дают резервирование источников, несколько выходов — одновременную раздачу одного канала многим получателям и по разным протоколам.

Для протоколов, где соединение инициирует приёмник, приёмник за NAT подключается к передатчику сам — проброс входящих портов на его стороне не требуется.

1.4.2 Peer-протоколы надёжной передачи

Для передачи между передатчиком и приёмником доступны четыре Peer-протокола. Они делятся на два класса по принципу компенсации потерь:

- **ARQ** (Automatic Repeat reQuest) — потерянные пакеты передаются повторно по запросу приёмника. Требуется обратный канал и буфер на приёмнике; глубина восстановления настраивается. К этому классу относятся PS1, SRT и RIST.
- **FEC** (Forward Error Correction) — к потоку постоянно добавляются избыточные данные, позволяющие восстановить потери без обратного канала. К этому классу относится Pro-MPEG / RTP+FEC.

PS1 (Perfect Stream)

Протокол собственной разработки на базе UDP. Работает по принципу ARQ с выборочной ретрансмиссией. Отличается низкой ресурсоёмкостью и переносит потоки высокого битрейта, включая полноразмерный MPTS.

Соединение инициирует приёмник, поэтому передатчик требует аутентификации: приёмники прописываются как Peer с логином и паролем либо авторизуются по IP. Задержка определяется буфером приёмника (окно устранения джиттера и ретрансмиссии); буфер передатчика должен быть больше буфера приёмников. При смене активного источника на передатчике приёмники не переподключаются — разрыв восстанавливается штатной ретрансмиссией без видимого переподключения.

PS1 — проприетарный протокол и работает только между узлами Perfect Streamer. Особенно эффективен в конфигурации точка-многоточка: один передатчик и много приёмников.

SRT

Открытый протокол на базе UDP (UDT). Широко распространён и хорошо компенсирует потери пакетов. Поддерживает режимы listener и caller, что даёт совместимость со сторонним SRT-оборудованием и облачными сервисами.

В режиме listener на передатчике к одному потоку подключается несколько приёмников; для авторизации приёмники прописываются как Peer, доступна авторизация по IP. Иницилирующая сторона (caller) может находиться как на приёмнике, так и на передатчике. Запас полосы на ретрансмиссию задаётся в процентах сверх битрейта потока.

На плотных приёмных узлах с большим числом SRT-входов в режиме caller можно отключить встроенную буферизацию приёмника SRT (TSBPD): синхронизацию берёт на себя джиттер-буфер узла ([Синхронизация](#)), что заметно снижает нагрузку на CPU.

RIST

Открытый протокол на базе RTP/RTCP. Работает по принципу ARQ без ACK, только по NACK, что обеспечивает высокую эффективность. Использует unicast и multicast.

Реализованы профили Simple и Main: Simple занимает два UDP-порта подряд (базовый порт должен быть чётным), Main мультиплексирует данные в один порт. Поддерживается несколько путей (пиров) с взвешенной балансировкой — для резервирования и агрегации. При multicast приёмников может быть много, с авторизацией по IP.

Pro-MPEG / RTP+FEC (SMPTE 2022-1/2)

Доставка MPEG-TS поверх RTP с упреждающей коррекцией ошибок (FEC), без обратного канала. Протокол известен также как Pro-MPEG COP3. В PSS реализована двумерная матрица FEC (строки и колонки, SMPTE 2022-2).

Поток RTP-пакетов группируется в матрицу заданного размера, по строкам и колонкам которой формируются FEC-пакеты. Чем меньше матрица, тем лучше восстановление, но выше постоянные накладные расходы. FEC-каналы занимают два дополнительных порта (port+2 и port+4), что нужно учитывать при размещении нескольких потоков на одном хосте или в одной multicast-группе.

Достоинства — низкая фиксированная задержка и совместимость с профессиональным вещательным оборудованием. Недостатки — постоянный дополнительный трафик и слабое

восстановление при больших потерях (свыше $\sim 0,2\%$), поэтому протокол рассчитан на управляемые каналы с низкими потерями, а не на «тяжёлый» публичный интернет.

1.4.3 Выбор протокола передачи

Протокол	Принцип	Обратный канал	Задержка	Совместимость
PS1	ARQ поверх UDP	требуется	настраиваемая	только между узлами Perfect Streamer
SRT	ARQ поверх UDP (UDT)	требуется	настраиваемая	открытый; стороннее SRT-оборудование и облако
RIST	ARQ (NACK) поверх RTP/RTCP	требуется	настраиваемая	открытый; стороннее RIST-оборудование; unicast и multicast
Pro-MPEG / RTP+FEC	FEC поверх RTP	не требуется	низкая, фиксированная	открытый (SMPTE 2022-1/2); профессиональное вещательное оборудование

Ориентиры выбора:

- канал между узлами Perfect Streamer, высокий битрейт или полноразмерный MPTS, раздача точка-многоточка — **PS1**;
- совместимость со сторонним оборудованием или облаком, гибкая настройка запаса на ретрансмиссию — **SRT**;
- открытый транспорт на базе RTP, multicast, балансировка по нескольким путям — **RIST**;
- управляемый канал с низкими потерями, минимальная задержка без обратного канала, профессиональное вещательное оборудование — **Pro-MPEG / RTP+FEC**.

1.4.4 Планирование полосы, задержки и портов

ARQ-протоколы (PS1, SRT, RIST). Потери компенсируются повторной передачей, поэтому нужно предусмотреть:

- обратный канал от приёмника к передатчику;
- запас полосы сверх номинального битрейта — всплески потерь вызывают всплески ретрансмиссионного трафика;
- буфер на приёмнике под задержку канала. Чем больше буфер, тем глубже восстановление потерь, но выше сквозная задержка; ориентир — несколько значений RTT. Потоки низкого битрейта требуют большего по времени буфера, чтобы в окне восстановления помещалось достаточно пакетов.

FEC-протокол (Pro-MPEG / RTP+FEC). Избыточные пакеты передаются постоянно, независимо от фактических потерь, поэтому накладные расходы фиксированы и зависят от размера матрицы FEC. Обратный канал не нужен, задержка низкая и фиксированная, но восстановление ограничено — этот транспорт для каналов с низкими потерями.

Порты. Каждой слушающей точке нужен уникальный UDP-порт в пределах хоста или группы. Дополнительно учитывайте, что профиль RIST Simple занимает два порта подряд (чётный базовый и следующий), а Pro-MPEG — базовый порт плюс два FEC-порта (port+2 и port+4). В сети Meshwork общий каталог занятых адресов и портов помогает избегать коллизий (см. [Meshwork](#)).

1.4.5 Шифрование потоков

Все Peer-протоколы поддерживают шифрование AES по общей парольной фразе, которая вводится на обеих сторонах. Для SRT дополнительно выбирается длина ключа (128, 192 или 256 бит). Шифрование не изменяет битрейт потока.

Для Pro-MPEG шифрование является нестандартным расширением, поэтому при его включении совместимость со сторонним ПО и оборудованием не гарантируется.

1.4.6 Другие входы и выходы

Помимо Peer-протоколов доступны стандартные транспорты для приёма и передачи:

Протокол	Вход	Выход	Примечание
UDP	да	да	unicast/multicast, SSM, выбор интерфейса; до 7 TS-пакетов на датаграмму
RTP	да	да	восстановление порядка переставленных пакетов
TCP	да	—	режим клиента; приём там, где UDP заблокирован
HLS / HTTP	да	да	на приёме выбирается вариант адаптивного плейлиста по настраиваемому номеру (по умолчанию первый); вход — только для SPTS
RTSP	да	—	IP-камеры и RTSP-серверы; авто-ремукс в MPEG-TS; только для SPTS
RTMP / RTMPS	да	да	публикация на сторонние RTMP-приёмники; H.264 и HEVC; только для SPTS
file / device	да	да	запись в TS-файл и вывод на устройство (в т.ч. SDI); цикличное воспроизведение из файла
pipe (FIFO)	да	да	мост к внешнему процессу через именованный канал
std (внешнее приложение)	да	да	мост к нестандартным протоколам через консольное приложение

Публикация и приём RTMP описаны в [Публикация и приём RTMP](#); RTSP, HLS/HTTP-вход, файлы и устройства, именованные каналы и внешние приложения — в [Прочие входы и выходы](#). Детальные настройки каждого транспорта — в разделе [Веб-интерфейс](#). OTT-раздача (HLS, LL-HLS, DASH) описана отдельно в [OTT и DVR](#).

1.4.7 Резервирование источников и раздача

Резервирование. Для потока можно задать несколько входов; активен всегда только один. При аварии активного входа поток автоматически переключается на следующий по списку, а при восстановлении может возвращаться к приоритетному источнику. Это позволяет заранее спланировать основной и резервные источники канала; детальные настройки переключения входов описаны в [SPTS-потоки](#).

Раздача. Несколько выходов на один поток позволяют раздавать один канал сразу многим получателям и по разным протоколам одновременно.

1.4.8 Требования к входному потоку

- Соответствие ISO/IEC 13818-1, Single Program (SPTS) или Multi Program Transport Stream (MPTS).
- Состав дорожек — по выбранному типу контента SPTS-потока, см. [SPTS-потоки](#).
- Скремблированные (закодированные) потоки поддерживаются, см. [SPTS-потоки](#).
- Для синхронизации в потоке должны быть валидные метки PCR.

Фильтрация, модификация и режимы битрейта входного потока описаны в [SPTS-потоки](#), особенности мультиплекса — в [MPTS-потоки](#).

1.5 Быстрый старт

Сквозной сценарий: от установленного узла до работающего потока с раздачей. Предполагается, что Perfect Streamer уже установлен ([Установка](#)), служба запущена и активирована, а вы вошли в веб-интерфейс ([Первый старт и активация](#), доступ и роли — [Доступ и роли](#)).

1.5.1 Шаг 1. Создать поток

1. Откройте экран «Потоки» ([Потоки](#)) и нажмите «+ Новый поток».
2. В окне «Новый поток» ([Новый поток](#)) оставьте тип SPTS — одиночная программа — и задайте «Имя потока» (2–32 символа: латинские буквы, цифры, дефис, подчёркивание). Тип нельзя изменить после создания.
3. Нажмите «Создать» — откроется редактор потока ([Настроить поток](#)).

Новый поток создаётся приостановленным — это нормально: сначала настройка, затем запуск.

1.5.2 Шаг 2. Добавить вход

На вкладке «Входы» редактора нажмите «Добавить вход». Для типового приёма UDP/multicast выберите тип `udp` и укажите адрес группы и порт — поля Address (multicast group) и Port; названия полей — как в интерфейсе. Полный список протоколов приёма и их выбор описаны в разделе [Планирование и протоколы передачи данных](#).

Если реального источника под рукой нет, используйте тестовый генератор — вход типа `test-stream` ([Тестовые потоки](#); требуется установленный пакет программного транскодера, [Транскодеры](#)): он формирует испытательную таблицу со звуком без внешнего сигнала.

Входов может быть несколько — первый в списке основной, остальные резервные ([Резервирование источников](#)).

1.5.3 Шаг 3. Запустить и проверить

Новые объекты создаются приостановленными — снимите их с паузы: кнопкой паузы в строке входа на вкладке «Входы» («Возобновить вход») и кнопкой паузы потока в его строке списка или в заголовке редактора («Возобновить поток»). Поток перейдёт в состояние «В эфире».

Проверьте приём в окне статистики потока (*Статистика потока*):

- статус и активный вход — в шапке окна;
- секция «Проверка MPEG-TS» — все признаки должны быть отмечены (*Проверка валидности*);
- входной битрейт и графики;
- картинку потока видно в секции «Снимок» окна статистики и на экране «Мозаика» (*Мозаика*).

1.5.4 Шаг 4. Раздать поток

Вариант А — OTT (HLS). На вкладке «OTT» редактора потока установите «Сервис HLS» в значение OTT / HLS и сохраните. Шаблон URL появится в окне статистики (секция «URL доставки») — скопируйте его и подставьте вместо частей login/password реквизиты логина клиента, созданного на экране «Пользователи / Логины» (*Пользователи / Логины*), после чего URL откроется в плеере. Режимы раздачи, DASH и Low-Latency HLS описаны в *OTT и DVR*.

Вариант Б — транспортный выход. На вкладке «Выходы» добавьте выход нужного протокола (UDP, SRT, PS1, RIST и другие — *Планирование и протоколы передачи данных*), укажите адрес получателя и снимите выход с паузы. Выходов может быть несколько — поток уйдёт всем получателям одновременно.

1.5.5 Что дальше

- Выбор протоколов передачи и планирование схемы доставки — *Планирование и протоколы передачи данных*.
- Обработка потока на узле: фильтрация, битрейт, анализ, транскодирование — *Streamer: особенности реализации*.
- Запись архива и воспроизведение (DVR) — *DVR: архив потока*.
- Объединение узлов в сеть и наблюдение за площадкой — *Meshwork*.
- Справочник по всем экранам веб-интерфейса — *Веб-интерфейс*.

1.6 Meshwork

Meshwork — механизм объединения серверов Perfect Streamer в единую сеть узлов. Узлы одного домена автоматически находят друг друга и обмениваются служебной информацией по gossip-протоколу, характерному для децентрализованных сетей. Объединение узлов даёт три возможности:

- **Живой статус узлов.** На любом узле в админке видно состояние всех остальных узлов сети: кто на связи, кто молчит, кто за NAT, когда был последний контакт. Открывать каждый сервер отдельно не нужно. Узел, переставший обмениваться данными, переходит в состояние offline — это сразу видно на всех узлах.

- **Общий каталог ресурсов (библиотека).** Запущенные на узлах входы и выходы UDP / RTP / Pro-MPEG / RIST / PS1 / SRT попадают в общий каталог, видимый на всех узлах домена: удобно подобрать свободный адрес и порт, не пересечься по мультикасту, найти, где уже отдаётся или принимается нужный поток. В пределах домена пир между узлами поднимается по автологу — без ручного ввода логинов и паролей с обеих сторон.
- **Разнос оповещений по домену.** Алерты одного узла видны на всех узлах того же домена: дежурному достаточно одного окна, чтобы следить за всей площадкой. Внешние оповещения достаточно настроить на одном узле домена — уведомления будут приходить обо всех узлах этого домена.

Meshwork — это слой видимости и согласования: он не переносит потоки между серверами и не меняет конфигурацию вещания автоматически. Схему доставки вы по-прежнему строите сами, а Meshwork помогает её планировать и наблюдать.

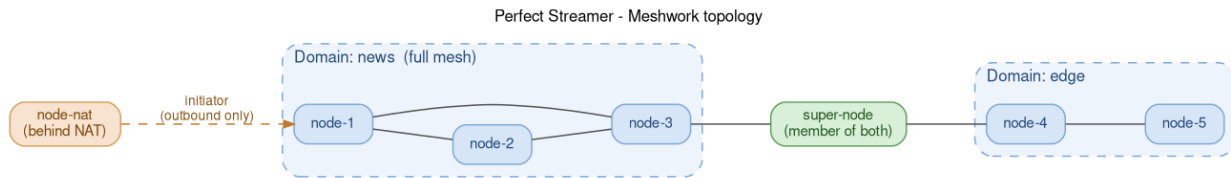


Рис. 1: Топология Meshwork: узлы одного домена образуют полную сетку прямых связей; узел за NAT подключается как инициатор; super-node состоит в двух доменах, но не соединяет их между собой.

1.6.1 Ключевые понятия

Узел

Один сервер Perfect Streamer в сети. У каждого узла есть короткое уникальное имя, по которому его узнают остальные. Без имени и секрета подписи узел в сети не участвует.

Домен

Логическая группа узлов и одновременно **граница изоляции**. Внутри одного домена узлы автоматически находят друг друга, делят общий каталог ресурсов и разносят оповещения. Между разными доменами обмена нет — домены независимы.

Автообнаружение (mesh)

Узлам достаточно задать одинаковый домен и хотя бы одну связь — дальше они находят друг друга сами, без ручного перечисления каждого соседа. В пределах домена выстраивается полная сетка прямых связей: все узлы домена становятся прямыми соседями, промежуточных цепочек не строится.

Связь прописывается по HTTP, но дальнейший обмен идёт по HTTPS. Для публичной сети настроенный HTTPS обязателен; для внутренней сети (intranet address) HTTPS не обязателен.

Явная связь

Соседа можно указать вручную — адресом, именем и общим секретом. Такая связь работает всегда, в том числе по обычному HTTP, и не требует общего домена. Подходит для точечного соединения двух площадок или подключения узла за NAT.

Super-node

Один узел может одновременно состоять в нескольких доменах. Он видит ресурсы и узлы каждого своего домена, но не соединяет домены между собой — изоляция сохраняется. В карте сети такой узел помечается как super.

Оповещения (алерты)

Встроенная служба оповещений следит за потоками, оборудованием узла, приёмом DVB и записью DVR и поднимает алерты, когда что-то выходит за заданные пределы. В сети с включённым разносом на каждом узле виден сводный набор алертов всего домена (см. [Оповещения \(алерты\)](#)).

Построение сети Meshwork

Настройка Meshwork сводится к тому, чтобы каждый узел знал своё имя и домен и имел хотя бы одного достижимого соседа. Дальше домен достраивает остальные связи сам. Все действия выполняются в веб-админке; изменение настроек Meshwork требует роли администратора. Точные поля и их расположение приведены в разделе [Веб-интерфейс](#).

Требования и достижимость

- **Сетевая достижимость.** Узел должен уметь установить TCP-соединение с административным веб-сервером соседа. Порт берётся из адреса соседа («хост:порт») либо из публичного порта, который сосед сообщает о себе ([Идентичность узла](#)), — это всегда тот порт, на котором настроен его веб-сервер, а при включённом у соседа HTTPS — его TLS-порт. Значения для свежей установки приведены в [Начальные настройки](#); нужный порт соседа должен быть открыт со стороны других узлов.
- **HTTPS для автообнаружения через интернет.** Если автообнаруженные узлы домена связываются по публичным (интернет-)адресам, им нужно установить между собой защищённое соединение (HTTPS), чтобы безопасно подтвердить подлинность друг друга по открытой сети. Поэтому включите HTTPS хотя бы на узлах, через которые идёт обнаружение. Включение HTTPS вступает в силу после **перезапуска** сервера. Если узлы доступны только по публичным адресам и ни один не поднят по HTTPS, прямые автообнаруженные связи не аутентифицируются: узел всё равно виден (через ретрансляцию от соседа), но его прямая связь помечается причиной (см. [Карта сети и каталог ресурсов](#)).
- **Локальная сеть — HTTPS не требуется.** Если узлы домена общаются по частным адресам одной доверенной сети (10.x.x.x, 172.16–31.x.x, 192.168.x.x, а также 127.x.x.x и 169.254.x.x), защищённое соединение для автообнаружения не нужно — узлы подтверждают подлинность и по обычному HTTP. Узел распознаётся как локальный по IP-адресу, поэтому в локальной сети задавайте соседей именно по IP-адресам, а не по доменным именам. HTTPS при этом можно включить — он будет использован, если доступен.
- **Явная связь работает и по HTTP** — на неё требование HTTPS не распространяется в любом случае.
- **Имя и секрет обязательны.** Узел участвует в сети, только если у него заданы имя и секрет подписи. Без имени или без секрета узел в Meshwork не участвует.
- **Ограничения на имена.** Имя узла — от 2 до 16 символов, имя домена — не длиннее 16 символов; разрешены латинские буквы, цифры и знаки `_`, `-`, `.`. Пробелы и другие символы недопустимы (например, домен «Alex Home» задать нельзя — используйте Alex_Home).

Идентичность узла

В настройках сервера задаётся, кем узел представляется в сети:

- **Имя узла** — короткое уникальное имя, по которому узел узнают остальные. Пустое имя означает, что узел в сети не участвует.
- **Заметка** — произвольная подпись узла (например, «Москва, стойка 3»), показывается в админке для удобства и на работу сети не влияет.
- **Домен** — узлы с одинаковым доменом автоматически находят друг друга, делят общий каталог ресурсов и разносят оповещения. Пустое значение отключает автообнаружение (узел может работать только по явному списку соседей).
- **Публичные порты** (HTTP и HTTPS) — порты, которые узел сообщает соседям для обратной достижимости при трансляции адресов (NAT). Если не заданы, узел сообщает собственные порты веб-сервера.
- **Дополнительные домены** (режим super-node) — узел входит в автообнаружение каждого перечисленного домена наравне с основным. Ресурсы остаются изолированными по доменам — узел их не смешивает.

Соседи и секреты

Здесь задаётся, с какими соседями узел поддерживает прямую связь и как стороны проверяют друг друга. Запросы между узлами подписываются общим секретом, поэтому без совпадающих секретов связь не установится. Длина каждого секрета — от 16 до 128 символов.

Список соседей — это записи вида «адрес — имя — секрет»:

- **Адрес** соседа — «хост:порт» его админки. Запись может быть без адреса: такой сосед никогда не вызывается этим узлом (он только сам дозванивается) и опознаётся по имени и секрету — это применяют для узла за NAT, который умеет только инициировать соединение.
- **Имя** соседа должно в точности совпадать с именем, заданным у самого соседа.
- **Общий секрет** связи с этим соседом должен быть одинаковым на обоих концах.

Кроме списка соседей, у узла есть **собственный секрет подписи**: им узел подписывает свои исходящие запросы. Сосед проверяет подпись тем секретом, который хранит для этого узла, поэтому собственный секрет узла должен совпадать с секретом, записанным о нём у каждого соседа.

Примечание: Практическое правило: для простой площадки задайте один общий секрет и используйте его и как собственный секрет подписи на каждом узле, и как секрет в каждой записи о соседе — тогда любая пара узлов аутентифицируется одинаково. Для более строгого разграничения можно держать отдельный секрет на каждую связь.

Быстрый старт

Вариант А. Два узла напрямую (явная связь)

Самый простой Meshwork — два сервера, соединённых вручную.

1. На каждом узле задайте уникальное короткое имя (например, node1 и node2).
2. На каждом узле добавьте запись о соседе: адрес его админки, его имя и общий секрет.
3. На каждом узле задайте собственный секрет подписи. Проще всего держать единый общий секрет на оба узла.
4. Откройте карту сети в админке. В течение нескольких секунд оба узла увидят друг друга в состоянии «на связи».

Домен при этом можно не задавать, но без общего домена вы получите только взаимную видимость узлов. Общий каталог ресурсов и разнос оповещений работают лишь между узлами с одинаковым доменом — если они нужны, задайте узлам общий домен (вариант В).

Вариант В. Домен с автообнаружением

Когда узлов больше двух, удобнее задать им общий домен и не перечислять каждую связь вручную.

1. На каждом узле задайте уникальное имя и общий домен площадки.
2. Если узлы домена связываются через интернет (по публичным адресам), включите HTTPS хотя бы на одном-двух узлах домена и перезапустите их (см. [Требования и достижимость](#)). В локальной сети этот шаг можно пропустить.
3. Свяжите узлы «точкой входа»: на каждом узле пропишите хотя бы одного уже работающего соседа (адрес, имя, общий секрет), как в варианте А. Остальные узлы домена узел найдёт сам.
4. Откройте карту сети. Через несколько секунд в ней появятся все узлы домена: явно прописанные — как постоянные, найденные автоматически — как автообнаруженные.

Узлы за NAT

Узел в частной сети, до которого извне не достучаться напрямую, может полноценно состоять в сети как **инициатор**. Он сам устанавливает соединение к доступному соседу, и через это одно соединение данные идут в обе стороны — узел и сообщает о себе, и получает сведения о других. Такому узлу достаточно прописать хотя бы одного достижимого соседа.

Сведения о NAT-узле распространяются по домену автоматически: сосед, который с ним на связи, ретранслирует его состояние остальным. У дальних соседей такой узел будет показан как полученный через ретрансляцию, но в состоянии «на связи».

Чтобы узел за NAT был достижим и для входа (например, служил точкой входа домена), настройте публикацию публичного порта (см. [Идентичность узла](#)) и пробросьте этот порт на маршрутизаторе/файрволе на административный веб-сервер узла. Со стороны соседей такой узел держит запись «имя + секрет без адреса»: сосед не будет тщетно дозваниваться до частного адреса, а опознает узел по имени и секрету, когда тот подключится сам.

Типичные проблемы

- **Узлы не видят друг друга.** Проверьте: у обоих задано имя узла; у каждого есть секрет подписи и он совпадает с тем, что записан о нём у соседа; адрес соседа верен и его админ-порт открыт со стороны другого узла.
- **Сосед в состоянии «нет связи».** Узел вышел за окно ожидания: проверьте сеть и работоспособность его веб-сервера. Для узла за NAT убедитесь, что он сам инициирует соединение к достижимому соседу либо что настроены публикация и проброс порта.
- **Автообнаруженный сосед помечен причиной, прямая связь не работает.** В домене не удаётся подтвердить подлинность узлов. Если они связываются через интернет, включите HTTPS хотя бы на одном узле домена и перезапустите его; в локальной сети проверьте, что соседи заданы по IP-адресам, а не по доменным именам. По явной связи (постоянные узлы) связь работает и без HTTPS.
- **Ложный признак NAT.** Если узлы заданы по доменным именам, а не по IP, признак NAT может не определяться по адресу — это не мешает работе. Для надёжной диагностики NAT используйте IP-адреса.
- **Смена домена «сбросила» карту.** Изменение домена или набора дополнительных доменов пересобирает членство: автообнаруженные связи и общий каталог временно сбрасываются и собираются заново. Это нормально.

Карта сети и каталог ресурсов

Meshwork даёт два сводных экрана в админке: карту узлов сети и общий каталог занятых сетевых ресурсов. Оба обновляются автоматически и сводят по домену информацию, которую иначе пришлось бы собирать с каждого сервера вручную.

Карта сети

Карта показывает состав сети — по одной строке на каждый известный узел, включая тот, с которого вы смотрите. Узлы проверяют связь друг с другом примерно каждые 5 секунд, и картина обновляется сама. Если Meshwork не настроен, список пуст.

Что показывает строка узла:

- **Имя узла** и адрес его админки.
- **Свой узел** — строка того узла, на котором вы смотрите карту; помечена особо и всегда показана «на связи».
- **Тип связи** — постоянный (узел из явного списка соседей) или автообнаруженный (найден автоматически через общий домен).
- **Состояние** — «на связи» или «нет связи». Если подтверждений от узла нет около 15 секунд, он помечается «нет связи». Перезапуск соседа распознаётся автоматически, поэтому короткое мерцание состояния сразу после его перезапуска — это нормально.
- **Последний контакт** — когда последний раз была успешная связь.
- **Домен** — домен или домены узла. У super-node их несколько, и узел дополнительно помечен как super.
- **NAT** — признак того, что на пути к узлу обнаружена трансляция адресов.

- **Ретрансляция / Через** — если сведения о доступности и NAT узла получены не напрямую, а через соседний узел, строка помечается «ретрансляция», а в поле «через» показано имя этого соседа. Так отображаются, например, узлы за NAT, до которых данный узел не может достучаться напрямую. Как только устанавливается прямая связь, она имеет приоритет над ретранслированными сведениями.
- **Причина** — появляется, если автообнаруженный сосед достижим напрямую, но его не удаётся аутентифицировать. Обычно это значит, что узлы связываются по публичным адресам, а в домене нет узла с включённым HTTPS, через который узлы подтверждают подлинность друг друга (см. [Требования и достижимость](#)). Пометка означает, что не работает **прямая** связь; сам узел при этом может оставаться «на связи» через ретрансляцию. В локальной сети (узлы по частным IP-адресам) такой пометки не возникает.

Общий каталог ресурсов

Отдельный экран админки показывает распределённый каталог занятых сетевых ресурсов — все запущенные на узлах домена входы и выходы UDP / RTP / Pro-MPEG / RIST. Это сводный по домену список занятых адресов и портов; по нему удобно выбирать свободный ресурс и видеть, где какой поток уже задействован.

Для каждой записи каталог показывает протокол и направление (вход/выход), узел-источник, домен, адрес/группу/порт, VLAN, при необходимости источник для source-specific multicast и заметку.

Тот же каталог отвечает и на обратный вопрос — кто ещё работает с адресом конкретного входа или выхода: рядом с адресом стоит кнопка «Показать в библиотеке» ([Библиотека — этот поток](#)).

Публикацией управляют два дополнительных поля у самого входа/выхода (показываются только у транспортов, попадающих в каталог, — UDP / RTP / Pro-MPEG / RIST / PS1 / SRT; названия — как в интерфейсе): Peer note (meshwork library) — публичная заметка для каталога (если пуста, используется обычная заметка записи) и Peer private — исключение записи из публикации.

Тип связи PS1 / SRT

Помимо UDP / RTP / Pro-MPEG / RIST, в каталог попадают и соединения [PS1](#) и [SRT](#). У такого соединения две стороны: вызывающая (дозванивается до удалённого адреса) и слушающая (принимает подключение). Для вызывающей стороны каталог показывает тип связи:

- **внутренняя** — вход/выход настроен подключаться как участник домена (доменный автологин включён): он соединяется с другим узлом сети и подтверждает свою принадлежность домену;
- **внешняя** — подтверждения участника домена нет: это обычное подключение к произвольной точке. Даже если по указанному адресу стоит узел сети, без доменного автологина связь считается внешней — тип отражает, как настроено подключение, а не то, входит ли получатель в сеть.

Это пометка-подсказка: на саму передачу она не влияет, а помогает отличить подтверждённые внутридоменные связи от внешних. У слушающих сторон и у семейства UDP / RTP тип не показывается.

Оповещения (алертер)

Алертер — встроенная служба оповещений. Она следит за состоянием потоков, оборудования узла, приёма DVB и записи DVR и поднимает **алерты** (инциденты), когда что-то выходит за заданные пределы. Список активных алертов виден в админке; в сети с включённым разносом по домену на каждом узле виден сводный набор алертов всего домена.

Алертер работает и на одиночном узле — Meshwork нужен только для разноса алертов по домену.

Как устроен алерт

- У каждого инцидента есть **устойчивый идентификатор**: повторное появление той же проблемы — это тот же инцидент, а не новый.
- Инцидент проходит состояния: **поднят** (впервые замечен), **обновлён** (замечен снова с существенным изменением), **снят** (проблема ушла). Активный набор — это инциденты, которые ещё не сняты.
- У каждого алерта есть **уровень серьёзности** (по нарастающей: информация, предупреждение, ошибка, критический, требует действий администратора), **код** (числовой вид инцидента — см. каталог ниже), **источник** (поток, узел, хранилище), **узел-источник** (кто поднял алерт) и человекочитаемое имя объекта.
- Большинство алертов **уровневые**: служба периодически переоценивает условие и сама поднимает или снимает алерт. Часть алертов, требующих действий администратора, **липкие** — они не снимаются сами, пока причина не устранена (их снимают вручную, см. [Просмотр и снятие алертов](#)).

Список активных алертов хранится в памяти и очищается при перезапуске сервиса.

Управление на уровне потока. У каждого потока есть выключатель, полностью заглушающий его алерты (вместе с алертами его входов и выходов), и задержка алерта: источник, восстановившийся в течение заданного времени, алерт не поднимает — это сглаживает кратковременные мерцания.

Настройки алертера

Настройки находятся в разделе оповещений и требуют роли администратора. Точные поля описаны в разделе [Веб-интерфейс](#); ниже — что можно настроить.

- **Главный выключатель** службы оповещений (по умолчанию включена).
- **Разнос по домену** (по умолчанию включён): узел делится своими алертами с узлами того же домена и принимает их — на каждом узле виден сводный набор. Действует только при заданном домене.
- **Разнос системных алертов** (по умолчанию выключен): алерты об оборудовании узла (CPU и память хоста, процесса стримера и транскодеров, загрузка сети и GPU, рабочие потоки и фронтенд DVB — коды 16–23 и 40–45) по умолчанию локальны; включите, чтобы они тоже разносились по домену. Алерты хранилищ (коды 24–25) остаются локальными всегда.
- **Пороги потоков**: тайм-аут отсутствия данных (по умолчанию 15 с), минимально допустимый битрейт (по умолчанию проверка выключена), порог ошибок счётчика непрерывности за окно 10 с (уровни предупреждения и ошибки).
- **Пороги ресурсов узла** — пары «предупреждение / ошибка» в процентах для CPU и памяти хоста, процесса стримера и транскодеров, загрузки сетевого интерфейса и GPU, а также заполнения хранилища DVR. Пороги хоста заданы выше, чем у процесса стримера и транскодеров,

чтобы отдельный компонент предупреждал раньше, чем насытится вся машина. Значение 0 отключает соответствующий уровень.

- **Пороги DVB-фронтенда** — уровень сигнала, SNR/качество, частота битовых ошибок и неисправленные блоки; применяются к адаптеру с захватом сигнала.
- **Пороги записи DVR** — число неудачных записей подряд, множитель застопоривания записи, доля непокрытия архива и время чтения/записи чанков на диск.

Каталог кодов алертов

Код — числовой вид инцидента, удобный для группировки и локализации. В колонке «Разнос» указано, разносится ли алерт по домену (при включённом разносе) или остаётся локальным для узла.

Потоки: состояние и рабочий цикл (источник — поток)

Код	Событие	Разнос
1	вход или выход перешёл в состояние ошибки	по домену
2	неустраняемая ошибка рабочего цикла, нужен ручной сброс (требуется действий администратора)	по домену
3	запущенный поток не отдаёт данные дольше тайм-аута	по домену
4	битрейт потока держится ниже минимального	по домену
13	поток поставлен на паузу: не осталось пригодного входа (требуется действий администратора)	по домену
14	вход или выход самоприпарковался и не восстанавливается повтором (требуется действий администратора)	по домену
27	поток работает на резервном входе — произошло переключение	по домену
39	высокая загрузка CPU рабочим потоком потока	по домену

Соответствие транспортного потока (анализатор TR 101 290) (источник — поток; все разносятся по домену)

Код	Событие
5	повторяющиеся разрывы PCR (TR 101 290, 2.3)
6	интервал повторения PCR более 40 мс (2.3а)
7	интервал повторения PAT более 500 мс (1.3)
8	интервал повторения PMT более 500 мс (1.5)
9	интервал повторения PTS более 700 мс (2.5)
10	точность PCR хуже 500 нс (2.4)
11	пере- или недополнение буфера T-STD по видео (3.3)
12	дрейф PCR более 30 ppm (историческое: оповещение больше не поднимается, дрейф отображается как метрика — Постоянные измерения)
15	ошибки счётчика непрерывности выше порога за короткое окно (1.4)
46	ошибка CRC таблиц PSI (2.2)
47	установлен признак transport_error_indicator (2.1)
48	превышен интервал повторения SI-таблицы (SDT / EIT / TDT / NIT)
49	сводный вердикт: поток не соответствует TR 101 290

Часть кодов анализатора появляется только при включённых опциях анализа (глубокий анализ, анализ PTS-разрывов, анализ буфера T-STD) и применима лишь к потокам с постоянным битрейтом; сводный вердикт (49) выносится только для уверенно постоянного битрейта. Подробнее — в разделе [Анализатор](#).

Ресурсы и оборудование узла (источник — система)

Код	Событие
16	загрузка CPU хоста выше порога
17	использование памяти хоста выше порога
18	CPU процесса стримера выше порога
19	память процесса стримера выше порога
20	суммарный CPU транскодеров выше порога
21	суммарная память транскодеров выше порога
22	загрузка сетевого интерфейса выше порога (на каждый интерфейс)
23	загрузка GPU выше порога (на каждое устройство)
24	заполнение хранилища DVR выше порога (на каждое хранилище)
25	файловая система хранилища DVR недоступна (требует действий администратора)
40	высокая загрузка CPU рабочим потоком DVB-адаптера
41	DVB-фронтенд потерял сигнал (lock)
42	уровень сигнала DVB ниже порога
43	SNR/качество сигнала DVB ниже порога
44	частота битовых ошибок (BER) DVB выше порога
45	неисправленные блоки DVB выше порога

Все алерты этой группы по умолчанию локальны. Коды 16–23 и 40–45 можно разнести по домену переключателем разнеса системных алертов; коды 24–25 остаются локальными всегда. Пороговые алерты DVB-фронтенда (42–45) оцениваются только на адаптере с захватом сигнала; код 41 поднимается при потере захвата (lock). BER (44) по умолчанию выключен, так как сырая шкала зависит от приёмника (см. [DVB-приёмник](#)).

Meshwork и жизненный цикл (источник — сеть/система; локально)

Код	Событие
26	узел сети замолчал — перестал подтверждать связь
28	процесс стримера завершил запуск (кратковременное уведомление)
37	в одном домене два узла с одинаковым именем (требует действий администратора)

Здоровье записи DVR (источник — поток; локально)

Код	Событие
29	записи сегментов или индекса DVR не проходят — архив не пишется
30	запись DVR застопорилась: есть вход, но новый сегмент долго не сохраняется
31	архив DVR деградировал — пропуски в недавнем покрытии
32	индекс архива DVR повреждён (требует действий администратора)
50	высокое время чтения/записи чанков DVR на диск

Конфигурация, транскодеры, OTT и сертификаты (источник — система/поток)

Код	Событие	Разнос
33	основная конфигурация не загрузилась при старте, сервис поднялся на запасной (требуется действий администратора)	по домену
34	в режиме low-latency HLS/DASH отброшены звуковые дорожки в неподдерживаемом кодеке (поддерживаются AAC и AC-3)	по домену
35	не удалось выпустить или продлить HTTPS-сертификат через встроенный клиент ACME	по домену
36	настроенный транскодер не загрузился при старте — нет исполняемого файла или устройства (требуется действий администратора)	по домену
38	поток требует транскодер (программный, NVIDIA или Intel VPL), недоступный на этом узле (требуется действий администратора)	локально

Модуль условного доступа (CI/CAM, EN 50221) (источник — система; локально)

Код	Событие
51	модуль CAM извлечён (требуется действий администратора)
52	нет подписки: модуль сообщает, что программа не оплачена
53	ошибка CI/CA — неустранимая ошибка канала или условного доступа

Просмотр и снятие алертов

В админке есть раздел оповещений со списком алертов. Его можно фильтровать по набору (только активные / только снятые / все), по уровню серьёзности, по типу и идентификатору источника, по коду, по домену и по времени, а также искать по тексту сообщения. По умолчанию показаны и активные, и недавно снятые записи (журнал переходов), поэтому общее число строк может превышать число активных инцидентов — для текущего активного набора выбирайте режим «только активные».

Состояние не обязательно опрашивать вручную: админка получает изменения активного набора в реальном времени и обновляет список сама, как только алерт поднимается, обновляется или снимается.

Уровневые алерты снимаются автоматически, когда условие уходит. Липкие алерты, требующие действий администратора (например, ошибка загрузки конфигурации, недоступность хранилища, жёсткая ошибка рабочего цикла), сами не снимаются: после устранения причины такой инцидент снимают (подтверждают) вручную. Снятие действует на том узле, который поднял алерт: если в сети вы видите чужой (разнесённый по домену) инцидент, снимать его нужно на узле-источнике (он указан в строке).

Разнос оповещений по домену

При включённом разноте и заданном домене на каждом узле виден сводный активный набор по всему домену: собственные алерты узла плюс алерты соседей того же домена. У каждой строки указан узел-источник, поэтому видно, на каком именно узле возникла проблема.

- **Разносятся по домену:** потоковые алерты и алерты соответствия транспортного потока, переключение на резервный вход, ошибки загрузки конфигурации и транскодера, отбраковка звука OTT, ошибка сертификата ACME.
- **Остаются локальными:** ресурсные алерты об оборудовании и хранилищах, «узел замолчал», коллизия имён узлов, уведомление о старте сервиса, алерты здоровья записи DVR и алерты

модуля условного доступа. Системные алерты об оборудовании (коды 16–23 и 40–45) можно перевести в разнос переключателем разноса системных алертов.

Алерт «узел сети замолчал» каждый узел поднимает самостоятельно о том соседе, который перестал отвечать, — поэтому об одном упавшем узле его соседи поднимут такой алерт независимо друг от друга.

Внешняя доставка алертов

Помимо списка в браузере, алерты можно доставлять во внешнюю команду, в чат Telegram и по электронной почте (SMTP). Каждый канал независим и по умолчанию выключен, а доставку ведёт на своём фоновом потоке, поэтому медленный или недоступный адресат не задерживает остальные каналы и саму службу.

У каждого канала есть минимальный уровень серьёзности для доставки (по умолчанию — предупреждение): доставляется каждый видимый на узле алерт (собственный или разнесённый по домену) не ниже этого уровня, причём и при подъёме, и при снятии. Параметры подключения каждого канала описаны в разделе [Веб-интерфейс](#).

Типичные проблемы

- **Алерты соседей не видны.** Проверьте, что на узлах включён разнос по домену и задан общий домен. Системные алерты об оборудовании по умолчанию локальны — включите разнос системных алертов, если их тоже нужно видеть по домену.
- **Липкий алерт не уходит после устранения причины.** Снимите его вручную на узле-источнике.

1.7 Streamer: особенности реализации

Раздел описывает, как узел обрабатывает потоки: конвейеры SPTS и MPTS, анализатор, демультиплексор и мультиплексор, тестовые потоки, OTT-раздача и DVR, транскодеры, приём DVB, публикация и приём RTMP, работа с файлами и внешними приложениями, EPG. Общая модель передачи и протоколы описаны в разделе [Планирование и протоколы передачи данных](#), управление через веб-интерфейс — в разделе [Веб-интерфейс](#).

1.7.1 SPTS-потоки

SPTS (Single Program Transport Stream) — поток MPEG-TS с одной программой; это основная единица обработки Perfect Streamer. Тип потока — SPTS или MPTS — выбирается при создании и не может быть изменён позже. Для SPTS-потока дополнительно выбирается тип контента (значения — как в интерфейсе): Default (audio + video), Audio only (звук без видео, радиоканалы) или Video only.

Скремблированные (закодированные) потоки поддерживаются: скремблирование определяется автоматически, и проверки элементарных потоков на это время приостанавливаются. Настройка «Скремблированный канал» объявляет поток постоянно скремблированным и отключает автоопределение.

Общие требования к входному потоку перечислены в [Требования к входному потоку](#), протоколы приёма и передачи — в разделе [Планирование и протоколы передачи данных](#). Обработка мультиплексов описана в [MPTS-потоки](#). Создание и настройка потоков выполняются в веб-интерфейсе ([Потоки](#)).

Конвейер обработки

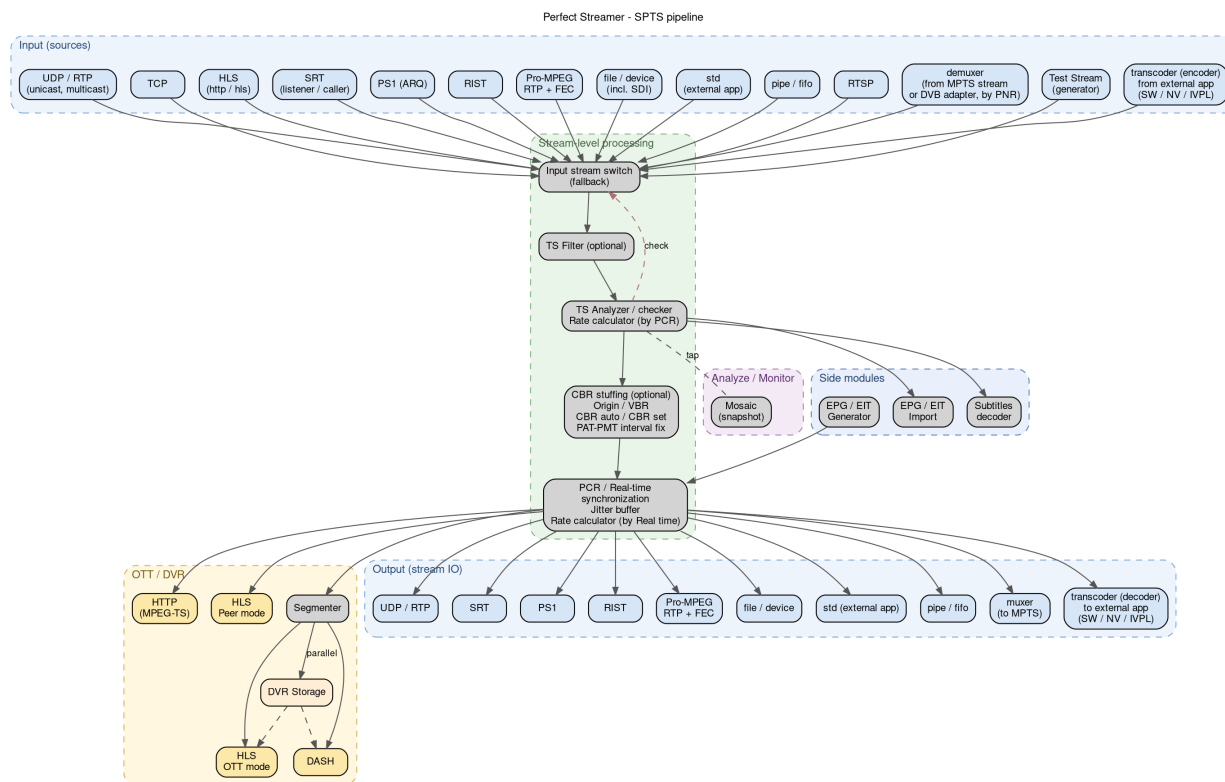


Рис. 2: Конвейер SPTS-потока: входы, обработка на уровне потока (переключатель входов, TS-фильтр, анализатор, выравнивание битрейта, синхронизатор), подсистема OTT / DVR и выходы.

Все входы потока сходятся в переключатель входов, дальше поток проходит последовательные этапы обработки:

1. **Переключатель входов** — выбирает активный источник и переключает поток на резерв при аварии (*Резервирование источников*).
2. **TS-фильтр** — необязательная фильтрация и модификация MPEG-TS (*Фильтрация MPEG-TS, Модификация потока*).
3. **Анализатор** — непрерывная проверка валидности и измерение потока (*Анализатор*); по его вердикту переключатель входов признаёт источник аварийным. Данные анализатора использует также мозаика (*Мозаика*, отключается настройкой «Генерировать мозаику»), импорт EPG и декодер субтитров.
4. **Выравнивание битрейта** — необязательное заполнение до постоянного битрейта и исправление интервала PAT/PMT (*Управление битрейтом*).
5. **Синхронизатор** — выдача пакетов в темпе PCR с компенсацией сетевого джиттера (*Синхронизация*). Сюда же генератор EIT вставляет таблицы программы передач (*Генератор EIT*).

С выхода синхронизатора поток раздается на все выходы: транспортные протоколы (*Peer-протоколы надёжной передачи*), запись в файл, мультиплексор MPTS (*Мультиплексор*), транскодер (*Транскодеры*), а также в подсистему OTT / DVR (*OTT и DVR*). Источником SPTS-потока может служить и отдельная программа мультиплекса — вход-демультиплексор (тип demuxer) выделяет её из MPTS-потока или DVB-приёмника по номеру программы (*Демультиплексор*).

Резервирование источников

У потока может быть несколько входов, но активен всегда только один. Порядок входов в списке задаёт приоритет: первый — основной, остальные — резервные. Если активный вход признан аварийным, поток переключается на следующий по списку; после последнего входа перебор продолжается с первого. Вход, поставленный на паузу, при переключении не учитывается.

Аварийность определяет анализатор: вход считается аварийным, если поток не проходит проверку валидности или данные не поступают дольше заданного таймута. Работа потока на резервном входе фиксируется оповещением (*Оповещения (алерты)*).

Настройки переключения — на вкладке «Поток» редактора потока (*Настроить поток*):

«Таймаут потока (с)»

Время без валидного входного потока, после которого выполняется переключение на следующий вход. По умолчанию 30 секунд.

«Интервал проверки (с)»

Период повторной проверки источников. По умолчанию 60 секунд.

«Повторно проверять основной вход»

Пока поток работает на резервном входе, входы выше по списку повторно проверяются с интервалом проверки. Как только более приоритетный источник снова валиден, поток возвращается на него. Включено по умолчанию.

Фильтрация MPEG-TS

По умолчанию поток передаётся как есть. Фильтрация применяется на двух уровнях: правила PID — отдельно на каждом входе, фильтры таблиц — на уровне потока.

Правила PID настраиваются на вкладке «MPEG-TS PID» окна входа (*Редактор входа и выхода*):

«Принимать PID (белый список)»

Список разрешённых PID. Если список пуст, разрешено всё, кроме запрещённого чёрным списком.

«Отклонять PID (чёрный список)»

Список запрещённых PID. PID, указанный одновременно в обоих списках, отклоняется.

Таблица PMT после фильтрации регенерируется по фактическому составу элементарных потоков. Правила PID применяются до автоматической раскладки PID (*Автоматическая раскладка PID*), если она включена.

Фильтры уровня потока — на вкладке «MPEG-TS» редактора потока, секция «Фильтры» (по умолчанию все выключены):

«Очищать ненужные таблицы»

Удаляет служебные данные, не относящиеся к программе, и элементарные потоки неизвестных типов; потоки известных типов (видео, звук, телетекст, субтитры) и таблицы PAT/PMT сохраняются. Отдельные таблицы (SDT, EIT, NIT, TDT, CAT) управляются собственными фильтрами.

«Удалять CAT / ECM / EMM»

Удаляет таблицы и сообщения систем условного доступа (CAS), включая CA-дескрипторы из PMT — например, чтобы очистить остатки CAS после дескремблирования.

«Удалять EIT»

Удаляет таблицы программы передач источника. По умолчанию EIT проходит насквозь.

«Удалять NIT», «Удалять TDT»

Удаляют таблицы сети и меток времени источника.

«Удалять телетекст», «Удалять субтитры»

Удаляют потоки телетекста и DVB-субтитров вместе с их записями в PMT.

«Удалять исходную SDT» (секция «SDT»)

Удаляет таблицу описания сервиса источника — например, при передаче потока на дальнейшее мультиплексирование. При назначении собственного имени сервиса ([Модификация потока](#)) исходная SDT заменяется автоматически, отдельно включать этот фильтр не нужно.

Примечание: Удаление обязательных таблиц (SDT, NIT, TDT, EIT, CAT) нарушает требования TR 101 290 к их присутствию в потоке — это осознанный выбор оператора, уместный, например, при передаче потока на дальнейшее мультиплексирование. Кроме того, любая фильтрация меняет состав пакетов потока; о влиянии на точность PCR см. [Управление битрейтом](#).

Модификация потока

Собственные данные сервиса задаются в секции «SDT» вкладки «MPEG-TS». Генерацию таблицы включает поле «Имя сервиса» (а также активная генерация EIT из EPG): если в источнике нет SDT, таблица генерируется; если есть — заменяется сгенерированной. Поля «Имя провайдера» и «Язык» заполняют соответствующие поля генерируемой таблицы. Настройка «ID сети» задаёт идентификатор сети для генерируемых таблиц EIT, когда источник не сообщает собственный идентификатор сети; если идентификатор в источнике есть, используется он.

Номер программы переназначается в секции «Номер программы»: включите «Изменить номер программы» и задайте «Новый номер программы». Таблицы PAT и PMT регенерируются с новым номером; SDT и EIT также перегенерируются.

«Фиксировать интервал PAT/PMT» — регенерирует PAT/PMT и вставляет их с интервалом, соответствующим TR 101 290. Применяется к источникам с редкими или нерегулярными таблицами.

Переназначение PID и языков выполняется на входе, вкладка «MPEG-TS PID» окна входа:

«Переназначить PID»

Пары «Из PID» / «В PID» — замена идентификаторов элементарных потоков, например для приведения потока к плану PID сети. При включённой автоматической раскладке PID ([Автоматическая раскладка PID](#)) эти пары не действуют.

«Переназначить язык аудио»

Пары «Аудио PID» / «Новый язык» — замена языка звуковой дорожки в PMT.

Генерация EIT из внешнего источника EPG включается в секции «Генерировать EIT из EPG» вкладки «Поток»: выбираются «Источник EPG» и «Канал EPG», и в поток вставляются таблицы EIT с программой передач ([Генератор EIT](#)). Обратная операция — настройка «Извлекать EIT в базу данных EPG», которая экспортирует программу передач из потока в базу EPG узла ([Импорт EPG/XMLTV](#)).

Для входов, принимающих исходный транспортный поток, доступно программное дескремблирование BISS: на входе задаётся ключ BISS-1 или BISS-E, и поток расшифровывается до анализа и дальнейшей обработки. Приём скремблированных сервисов с DVB-приёмника (CAM, BISS на уровне адаптера) описан в [DVB-приёмник](#).

Автоматическая раскладка PID

По умолчанию идентификаторы элементарных потоков переходят из источника в выходной поток без изменений, поэтому при переключении на резервный вход раскладка PID на выходе меняется вслед за источником. Автоматическая раскладка назначает PID сама, по одному и тому же правилу для любого входа: план PID выходного потока остаётся неизменным независимо от того, какой источник сейчас активен.

Режим доступен только для SPTS и включается в секции «MPEG-TS» вкладки «MPEG-TS» редактора потока:

«Автоматическая раскладка PID»

Включает режим.

«Базовый PID»

Первый PID окна: от 32 до 8127, по умолчанию 32. Нижняя граница выводит окно за пределы диапазона, зарезервированного под таблицы PSI/SI, верхняя оставляет всё окно ниже PID нулевых пакетов.

Программа перенумеровывается в окно из 64 PID, начинающееся с базового:

PID	Назначение
базовый	PMT
базовый + 1	видео
далее	звуковые дорожки
далее	остальные элементарные потоки: телетекст, DVB-субтитры, данные, SCTE-35
далее	PCR, если он не передаётся в одном из перечисленных потоков
далее	ЕСМ: сначала по дескрипторам уровня программы, затем по дескрипторам отдельных потоков
с конца окна вниз	ЕММ

Внутри каждой группы сохраняется порядок исходной таблицы PMT. Если видео в программе нет, раскладка сжимается: сразу за PMT идёт первая звуковая дорожка. ЕММ выделяются с конца окна, поскольку таблица CAT разбирается независимо от PMT и может прийти раньше — их адреса не должны зависеть от числа элементарных потоков.

Таблицы PAT, PMT и CAT пересобираются под новую раскладку. SDT, NIT, EIT и TDT не изменяются: ссылок на PID они не содержат.

Что учесть при включении режима:

- Режим принудительно включает фильтрацию MPEG-TS, поэтому PID, не объявленные в PMT и CAT, наружу больше не проходят. Поток, который полагался на транзит необъявленных PID, их теряет.
- Ручное переназначение PID на входах не действует и игнорируется, о чём в журнал пишется предупреждение. Веб-интерфейс скрывает эти поля, пока режим включён; сохранённые пары не теряются и возвращаются при его выключении.
- Списки приёма и отклонения PID продолжают работать и применяются до перенумерации: слот получают только выжившие элементарные потоки, и пропусков в окне не остаётся (*Фильтрация MPEG-TS*).
- На вход-транскодер (*Транскодеры*) и вход тестового генератора (*Тестовые потоки*) режим не действует: там план PID строит кодер или генератор, поэтому их собственные поля PMT PID остаются доступными.

- Если программа в окно не помещается, перенумерация не выполняется вовсе — раскладка остаётся исходной, а в журнал идёт предупреждение; наполовину перенумерованный поток не выпускается.
- Изменение любой из двух настроек применяется на лету: сам поток не перезапускается, но его входы и выходы переподнимаются — несколько секунд на пересинхронизацию.

В статистике потока блок «Информация о медиа — источник» показывает PID, пришедшие на вход, а «Информация о медиа — результат» и таблица битрейтов по PID — назначенные; при включённом режиме расхождение между ними штатно.

Автоматическая раскладка PID не связана с назначением PID в мультиплексе MPTS ([Назначение PID](#)) — это отдельная настройка тихег-входа.

Управление битрейтом

Режим управления битрейтом задаётся полем «Режим битрейта» на вкладке «MPEG-TS» (значения — как в интерфейсе):

Original (по умолчанию)

Поток передаётся без изменений; метки PCR источника не пересчитываются.

VBR

Удаляет NULL-пакеты источника, что минимизирует битрейт. Подходит, когда поток используется только для OTT-вещания или транзита. Метки PCR при этом не пересчитываются, поэтому точность PCR по TR 101 290 на выходе не гарантируется.

CBR (auto stuffing)

Выравнивает битрейт вставкой NULL-пакетов (stuffing) до постоянного значения. Целевой битрейт определяется автоматически: он фиксируется через несколько секунд после включения, повышается при устойчивом превышении и постепенно снижается, если пик контента долго остаётся ниже цели. Смена цели не прерывает выдачу потока.

CBR (explicit stuffing)

То же, но целевой битрейт задаётся явно полем «Битрейт заполнения (kbps)». Задавайте значение с запасом больше фактического битрейта контента.

В режимах CBR метки PCR пересчитываются по фактически переданным байтам, поэтому выход соответствует требованиям TR 101 290 к точности PCR. Дополнительно при активном заполнении на выходе поддерживается частота повторения PCR в пределах нормы даже на длинных видеокадрах. Анализатор в этом случае измеряет именно выдаваемый, выровненный поток ([Анализатор](#)).

Когда нужен CBR. Любая операция, которая вставляет или удаляет пакеты, — фильтры таблиц и PID, генерация SDT и EIT, исправление интервала PAT/PMT, смена номера программы, а также сам режим VBR — меняет положение контента относительно его меток PCR. Если получатель потока требует соответствия TR 101 290, например при DVB-вещании, включите режим CBR: заполнение маскирует эти изменения пересчётом PCR. Для потока после транскодера включите режим CBR на потоке-приёмнике ([PCR и битрейт выходного потока](#)).

Текущее состояние заполнения — целевой битрейт и скорость вставки NULL-пакетов — отображается в данных активного входа на странице потока ([Потоки](#), секция «Заполнение»; секция видна, только пока заполнение активно).

Синхронизация

Синхронизатор выдаёт пакеты на выходы в темпе меток PCR источника — поэтому валидные PCR обязательны для входного потока. Между приёмом и выдачей поток задерживается в джиттер-буфере, который сглаживает неравномерность прихода пакетов из сети.

Буфер настраивается отдельно на каждом входе (окно входа, [Редактор входа и выхода](#); дополнительные настройки, названия — как в интерфейсе):

Auto jitter buffer

Автоматический подбор задержки по типу входа: для входов TCP и HLS используется увеличенный буфер, для остальных — стандартный. Включено по умолчанию.

Jitter buffer (ms)

Размер буфера вручную, когда автоматический режим выключен (по умолчанию 500 мс). Увеличьте значение, если на нестабильной сети буфер регулярно опустошается.

PCR discontinuity window (ms)

Окно допустимого скачка PCR (по умолчанию 1000 мс). Скачок за пределы окна считается разрывом потока и вызывает жёсткую пересинхронизацию.

Медленное расхождение между темпом PCR источника и часами узла устраняет компенсатор дрейфа: корректировка применяется плавно, микросдвигами, без жёсткой пересинхронизации и без прерывания выдачи. Управление — на вкладке «Анализ» редактора потока: «Компенсация дрейфа синхронизации» (включена по умолчанию) и «Мягкое окно дрейфа синхронизации (ms)» — зона нечувствительности, внутри которой корректировка не применяется (по умолчанию 500 мс). Жёсткая пересинхронизация остаётся резервным механизмом на случай настоящих разрывов потока.

1.7.2 MPTS-потоки

MPTS (Multi Program Transport Stream) — поток MPEG-TS с несколькими программами (сервисами); каждая программа имеет уникальный номер (PNR). Основное применение — DVB-вещание и транзит готовых мультиплексов. Тип потока — SPTS или MPTS — выбирается при создании и не может быть изменён позже ([Новый поток](#)).

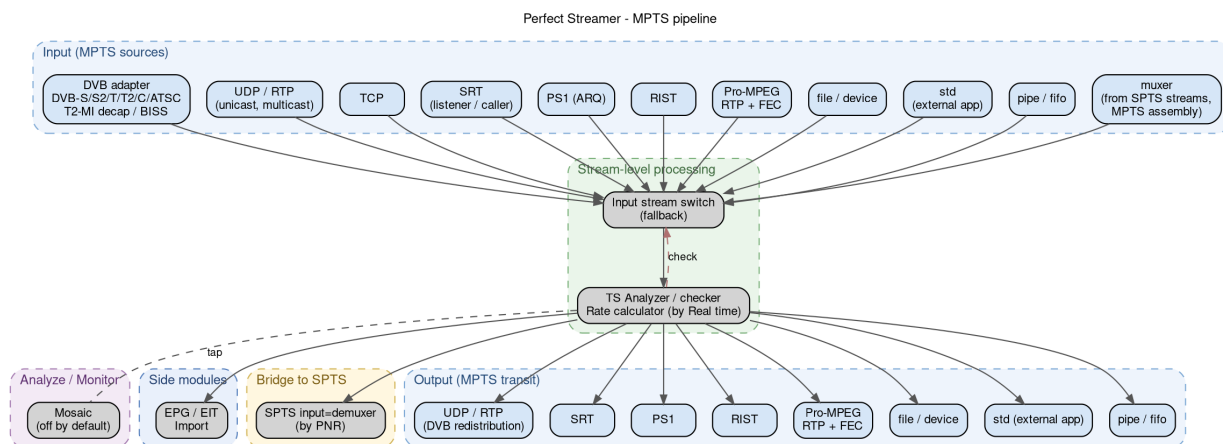


Рис. 3: Конвейер MPTS-потока: транзит мультиплекса без изменения состава; обработка отдельных программ выполняется через демультиплексор в SPTS-потоках.

Транзит мультимплекса

MPTS-поток — транзитный: мультимплекс передаётся как есть. Фильтрация и модификация MPEG-TS (*Фильтрация MPEG-TS*, *Модификация потока*), режимы битрейта (*Управление битрейтом*), OTT-раздача и запись DVR (*OTT и DVR*) к MPTS-потoku не применяются. Постоянный битрейт и состав собственного мультимплекса задаются при его сборке мультимплексором (*Мультимплексор*); обработка отдельной программы выполняется после её выделения в SPTS-поток (*Демультимплексор*).

Источники MPTS-потока — сетевые транспорты, принимающие готовый мультимплекс (UDP / RTP, SRT, PS1, RIST, Pro-MPEG, TCP, файл и другие — раздел *Планирование и протоколы передачи данных*), DVB-приёмник (*DVB-приёмник*) и мультимплексор (*Мультимплексор*). Однопрограммные источники — HLS, RTSP, RTMP, демультимплексор, транскодер, тестовый генератор — для MPTS недоступны. Выходы — транспортные, включая раздачу мультимплекса в сеть DVB-дистрибуции; OTT-выходов у MPTS-потока нет.

Резервирование источников работает так же, как у SPTS-потоков: несколько входов по приоритету, таймаут потока, интервал проверки и возврат на основной вход (*Резервирование источников*).

Анализ по программам

Анализатор MPTS-потока ведёт учёт по каждой программе отдельно и сводно по мультимплексу: битрейт, состояние и ошибки непрерывности, скремблирование, информация о медиа по каждой программе, а также идентификаторы мультимплекса (TS ID и идентификатор сети). Сводные счётчики программ — всего, чистых, скремблированных и с ошибками — выводятся в окне статистики потока (*Статистика потока*); в списке потоков MPTS-потоки отмечены значком MPTS и выделены в отдельную группу. Скремблированная программа считается валидной и в счётчик ошибок не попадает.

Монитор TR 101 290 (вкладка «Анализ», настройка «Монитор TR 101 290») выносит для MPTS один вердикт на весь мультимплекс: ошибки транспорта, непрерывность по всем PID, повторение и разрывы PCR по всем программам, присутствие таблиц PAT/PMT и SI (NIT, SDT, EIT, TDT). Окно допустимого скачка PCR задаётся настройкой входа «PCR discontinuity window (ms)» — как у SPTS-потоков (*Синхронизация*). Углублённые метрики SPTS — точность PCR, анализ видеобуфера T-STD, интервалы PTS — к мультимплексу целиком не применяются: для детального анализа программы выделите её в отдельный SPTS-поток (*Демультимплексор, Анализатор*).

Мозаика для MPTS-потока по умолчанию выключена (настройка «Генерировать мозаику»); при включении формируется миниатюра для каждой программы мультимплекса. Это заметная дополнительная нагрузка — на загруженных серверах включать мозаику мультимплекса не рекомендуется.

Дескремблирование BISS

На сетевых и файловых входах MPTS-потока (UDP / RTP, SRT, RIST, Pro-MPEG, TCP, PS1, файл и подобных) доступно программное дескремблирование BISS по программам: задаются пары «номер программы — ключ» (BISS-1 или BISS-E), и каждая указанная программа расшифровывается до анализа и дальнейшей передачи. Дескремблирование на уровне DVB-адаптера — CAM и BISS, включая ключи для T2-MI, — описано в *DVB-приёмник*.

EPG мультиплекса

Программа передач из состава мультиплекса извлекается в базу EPG узла настройкой «Извлекать EIT в базу данных EPG» ([Импорт EPG/XMLTV](#)). Генерация EIT на транзитном MPTS-потоке не выполняется: таблицы EIT программ переносят мультиплексор из потоков-источников при сборке ([Мультиплексор](#), [Генератор EIT](#)).

1.7.3 Анализатор

Встроенный анализатор непрерывно проверяет и измеряет активный вход каждого потока. Результаты используются в нескольких подсистемах узла: вердикт валидности управляет переключением входов ([Резервирование источников](#)), метрики и графики выводятся в окне статистики потока ([Статистика потока](#)), сводные плашки — в списке потоков, нарушения порождают оповещения ([Оповещения \(алертер\)](#)). Данные декодера анализатора используются мозаикой, импортом EPG и декодером субтитров ([Конвейер обработки](#)).

Базовые проверки и измерения включены всегда. Углублённые анализы включаются отдельными настройками на вкладке «Анализ» редактора потока — каждая добавляет нагрузку на CPU и включается по необходимости. Накопительные счётчики очищаются кнопкой «Сбросить статистику» в окне статистики.

Проверка валидности

Входной поток проверяется по этапам: захват синхронизации TS, приём таблиц PAT и PMT, наличие меток PCR, разбор звуковых и видеодорожек вплоть до декодирования ключевого кадра. Результат отображается в окне статистики в секции «Проверка MPEG-TS» — отдельные признаки TS, PAT, PMT, PCR, Audio, Audio ES, Video, Video ES. Проверка повторяется с интервалом проверки потока; вход признаётся аварийным, если поток не прошёл проверку или данные не поступают дольше таймаута потока — тогда выполняется переключение на резервный вход ([Резервирование источников](#)).

Скремблированный транспорт — валидное состояние ([SPTS-потоки](#)): транспортный и PSI-уровни продолжают проверяться, а после появления открытых данных полный анализ возобновляется автоматически.

Постоянные измерения

Без дополнительных настроек анализатор измеряет:

- **Битрейт** — скорость входного потока (у SPTS — по меткам PCR, у MPTS — по реальному времени) и выходная скорость после синхронизатора; график «Битрейт» в окне статистики.
- **Битрейт по PID** — текущая скорость и число пакетов в секунду по каждому PID активного входа; таблица в блоке «Анализатор» окна статистики. Тип PID берётся из разбора таблиц потока (PAT, PMT, video, aac и подобные); PID NULL-пакетов (8191) обозначен NULL, а поток, тип которого определить не удалось, — прочерком. Измерение только для SPTS: у MPTS-потока таблицы нет ни при каком источнике мультиплекса, а состав и скорости программ показывает таблица мультиплекса ([Анализ по программам](#)).
- **Непрерывность и транспорт** — ошибки непрерывности CC (окно и накопительно), ошибки транспорта (TEI), счётчики скремблированных пакетов; график «Непрерывность и скремблирование».

- **Интервал PCR** — средний и максимальный интервал между метками PCR. При включённом мониторе TR 101 290 действует строгий предел 40 мс, без него — предел ISO/IEC 13818-1 в 100 мс.
- **Джиттер PCR** — отставание планировщика выдачи от темпа PCR; график «Джиттер PCR».
- **Дрейф PCR** — систематический уход тактовой частоты источника относительно реального времени, в ppm. В отличие от джиттера, дрейф характеризует стабильность кварца энкодера и накапливается минутами; достоверный вердикт формируется на окне от 300 секунд. Допуск ISO/IEC 13818-1 — ± 30 ppm; устойчивое превышение подсвечивается в окне статистики («Дрейф PCR»), отдельно от вердикта TR 101 290: дрейф кварца — проблема энкодера на стороне источника, а не транспортной целостности.
- **Точность PCR** — погрешность вставки меток PCR относительно фактического момента передачи; допуск TR 101 290 — ± 500 нс. Метрика осмысленна только для потоков с постоянным битрейтом (см. детектор режима ниже).

Детектор режима мультиплекса. Анализатор автоматически классифицирует источник как CBR или VBR по регулярности PCR относительно позиции пакетов; вердикт фиксируется с гистерезисом, чтобы исключить частые переключения. Уверенный вердикт показывается плашкой CBR или VBR в списке потоков; пока данных недостаточно, плашки нет. VBR — не дефект: это нормальный режим, при котором проверки, осмысленные только для CBR (точность PCR, видеобuffer T-STD), автоматически не оцениваются.

Углублённые анализы

Три независимых анализа включаются на вкладке «Анализ» редактора потока (все выключены по умолчанию):

«Глубокий анализ»

Переводит анализатор в непрерывный режим разбора потока (заметная нагрузка на CPU) и добавляет:

- интервалы таблиц PAT и PMT (рекомендация — не более 500 мс);
- интервалы ключевых кадров (GOP) и IDR-кадров. Для проигрывателей со случайным подключением рекомендуется GOP не более 1 секунды. Если интервал IDR примерно совпадает с интервалом ключевых кадров, поток имеет closed-GOP — каждый GOP открывается полной точкой входа; если метрика IDR отсутствует, поток имеет open-GOP;
- интервал от PAT до ключевого кадра — от него зависит скорость старта воспроизведения при случайном подключении;
- паспорт видеокодека (тип потока, профиль и уровень, разрешение, развёртка, параметры VBV) и параметры звуковых дорожек;
- данные готовности к вставке рекламы: структура GOP, метки случайного доступа (RAI), события SCTE-35 и точки склейки. Порог предварительного уведомления о точке склейки задаётся настройкой «Уведомлять о точке склейки за (кадры)» (дополнительная настройка);
- контроль присутствия таблиц SI (SDT, EIT, NIT, TDT) и ошибки структуры элементарных потоков.

«Анализировать разрыв PCR/PTS»

Разброс между PCR и метками PTS/DTS по каждому элементарному потоку. Слишком большой разброс вызывает проблемы у проигрывателей с малым буфером синхронизации. Дополнительно измеряется интервал повторения видеометок PTS (показатель 2.5 отчёта TR 101 290, предел 700 мс).

«Анализировать видеобуфер T-STD»

Модель эталонного буфера декодера T-STD (ISO/IEC 13818-1): проверка, что видеобуфер не переполняется и не опустошается. Поддерживаются MPEG-2, H.264/AVC и HEVC; скорость слива адаптируется к фактическому битрейту видео. Оценивается только на потоках с постоянным битрейтом.

Результаты выводятся в окне статистики в секциях «Глубокий анализ» и «Анализатор». Настройки «Трассировка» и «Трассировка MPEG-TS» включают подробное журналирование проверки потока — только для диагностики, не оставляйте их включёнными постоянно.

Монитор TR 101 290

Настройка «Монитор TR 101 290» (вкладка «Анализ») включает непрерывный контроль соответствия потока стандарту ETSI TR 101 290. Монитор лёгкий — тяжёлый разбор элементарных потоков он не включает — и работает независимо от углублённых анализов.

Результат — единый вердикт по потоку: «Норма», «Проблема» или «Проверка...» на время накопления данных. Вердикт показывается цветной плашкой TR-290 в списке потоков и в окне статистики; нажатие открывает отчёт «TR 101 290» — таблицу показателей, сгруппированных по приоритетам стандарта (приоритет 1 — необходим для декодирования, приоритет 2 — рекомендуется для мониторинга, приоритет 3 — зависит от применения), с измеренными значениями и пределами.

Особенности оценки:

- Нарушение попадает в вердикт, только когда оно устойчиво (наблюдается не менее 30 из последних 60 секунд) — короткие всплески индикатор не зажигают. Отдельно отслеживается приёмочный уровень: нарушение, непрерывно длящееся 300 секунд, — критерий для технической приёмки канала.
- Часть показателей требует дополнительных анализов. Строки, зависящие от настроек «Анализировать разрыв PCR/PTS» и «Анализировать видеобуфер T-STD», при выключенной настройке помечаются в отчёте как отключённые и на вердикт не влияют. Численные интервалы PAT/PMT и контроль присутствия таблиц SI измеряются только при включённом «Глубоком анализе» — без него соответствующие строки остаются без измеренных значений.
- Точность PCR и видеобуфер T-STD оцениваются только после того, как детектор режима подтвердил постоянный битрейт источника, — и продолжают оцениваться до сброса статистики или изменения настроек выравнивания битрейта, даже если вердикт детектора позже изменится. На потоках, ни разу не подтверждённых как CBR, эти строки не оцениваются.
- При активном выравнивании битрейта монитор измеряет именно выдаваемый, выровненный поток ([Управление битрейтом](#)).
- Для MPTS-потока выносится один вердикт на весь мультиплекс ([Анализ по программам](#)).

Устойчивые нарушения дополнительно порождают оповещения с кодами анализатора — полный список приведён в [Каталог кодов алертов](#). Счётчики монитора и накопленную статистику сбрасывает кнопка «Сбросить статистику».

Помощник для рекламаций

Если у канала есть устойчивые нарушения TR 101 290 или дрейф PCR, оператор может получить готовый текст-задание для AI-чата, по которому тот составит формальное письмо-рекламацию поставщику потока. Запрос выполняется по HTTP API узла:

```
GET /data/stream/<id>/ai-complaint-prompt
```

Ответ — текст с перечнем активных нарушений, измеренными значениями, порогами стандарта и влиянием каждой ошибки на декодер получателя. Название потока, его идентификаторы и адреса источника в текст намеренно не включаются — вместо них подставлен плейсхолдер, который оператор заполняет сам, чтобы служебные данные не передавались стороннему AI-сервису. Функция доступна только для SPTS-потоков.

1.7.4 Демультимплексор

Вход-демультимплексор (тип demuxer) выделяет одну программу из мультимплекса и превращает её в обычный SPTS-поток. После выделения к программе применима вся обработка SPTS-конвейера (*SPTS-потоки*): фильтрация и модификация, транскодирование, OTT-раздача и запись DVR. Это единственный путь обработки принятого мультимплекса по отдельным программам (*MPTS-потоки*).

Чтобы подключить программу, в SPTS-потоке добавляется вход типа demuxer и выбираются (названия полей — как в интерфейсе):

- **Источник** (поля Source kind и Source id) — MPTS-поток этого же узла или DVB-адаптер (*DVB-приёмник*).
- **Программа** (поле PNR) — номер программы. Для источника-MPTS-потока, если он активен и уже проанализирован, номер выбирается из списка обнаруженных сервисов с их именами; для DVB-адаптера и для ещё не проанализированного источника PNR вводится вручную.

Особенности:

- Демультимплексор получает уже дескремблированный мультимплекс: ключи BISS задаются на входе MPTS-потока (*Дескремблирование BISS*), а дескремблирование CAM и BISS на уровне адаптера — в настройках DVB-адаптера (*DVB-приёмник*).
- Источник защищён от удаления: пока на MPTS-поток или DVB-адаптер ссылаются хотя бы один демультимплексор, удалить его нельзя — узел сообщит, какие объекты на него ссылаются.
- Для внутренних мультимплексов T2-MI используется составной номер программы, объединяющий номер несущей и PNR внутри неё: номер несущей × 65536 + PNR (например, несущая 1, PNR 7 → 65543; несущая 0 — внешний мультимплекс, обычный PNR). Составное значение вводится в поле PNR. Приём T2-MI описан в *DVB-приёмник*.

Один и тот же мультимплекс может одновременно раздаваться целиком (MPTS-поток с транспортными выходами) и по частям — отдельными SPTS-потоками через демультимплексоры.

1.7.5 Мультиплексор

Мультиплексор собирает MPTS-поток из отдельных SPTS-потоков узла и формирует полноценный DVB-мультиплекс: генерирует таблицы PSI/SI, ведёт расписание выдачи с моделью T-STD и выравнивает битрейт вставкой NULL-пакетов. При заданном целевом битрейте выход соответствует TR 101 290 и пригоден для подачи на модуляторы и в сети DVB-дистрибуции.

Сборка мультиплекса

1. Создать поток типа MPTS (*Новый поток*).
2. Добавить в него вход типа muxer — он единственный и содержит настройки всего мультиплекса (см. ниже).
3. Привязать программы — любым из двух способов:
 - из редактора программ мультиплекса: кнопка «Редактировать программы...» на muxer-входе, затем «Добавить программу» и выбор исходного потока;
 - со стороны источника: добавить SPTS-поток выход типа muxer и указать целевой MPTS-поток.

Новая привязка создаётся на паузе — программа появляется в списке со значком паузы, после проверки настроек её снимают с паузы.

4. Повторить для всех программ мультиплекса.

Участники мультиплекса — обычные SPTS-потоки: вся обработка (*SPTS-потоки* — фильтрация, модификация, транскодирование) выполняется на потоке-источнике до сборки. Один SPTS-поток может одновременно раздаваться напрямую и входить в мультиплекс.

Параметры мультиплекса

Задаются на muxer-входе MPTS-потока (названия — как в интерфейсе):

Transport stream id, Network id, Network name

Идентификаторы и имя сети транспортного потока — попадают в генерируемые таблицы PAT/SDT/NIT.

Country code (ISO 3166 alpha-3)

Код страны — используется в таблице TOT (часовой пояс, локальное смещение времени) и региональных дескрипторах NIT.

Output bitrate (kbps)

Целевой битрейт мультиплекса. Задаётся с запасом больше суммарного битрейта всех программ. Значение 0 отключает выравнивание — суммарный битрейт просто следует входным потокам, без заполнения NULL-пакетами (stuffing) и без модели T-STD.

SDT provider name

Имя провайдера — общее для всех сервисов мультиплекса (дополнительная настройка).

Параметры программ

Задаются на мuxeг-выходе потока-источника (названия — как в интерфейсе); редактор программ управляет составом, порядком, паузой и — при выключенном автоназначении PID — правкой PID:

SDT service name (empty = stream name)

Имя сервиса в таблице SDT; пустое значение — используется имя потока-источника.

PNR (0 = stream id)

Номер программы в мультиплексе; 0 — используется идентификатор потока-источника.

Logical channel number (0 = PNR)

Логический номер канала (LCN) для приёмников; 0 — совпадает с PNR.

Service type (0 = auto, ETSI)

Тип сервиса по классификации ETSI: 0 — определяется автоматически по составу программы (SD/HD, кодек, радио) либо задаётся вручную.

Порядок программ в таблицах мультиплекса меняется перетаскиванием строк в редакторе программ.

Назначение PID

Режим назначения PID выбирается переключателем «Automatic PID mapping» на мuxeг-входе:

- **Включено** (по умолчанию) — каждой программе назначаются новые PID из автоматического диапазона; коллизии исключены.
- **Выключено** — сохраняются исходные PID потоков-источников. Это нужно, например, при миграции действующего мультиплекса, чтобы приёмники не требовали повторного сканирования. Отдельные PID при необходимости меняются в редакторе PID программы (кнопка «PID» в редакторе программ): PMT PID и целевые PID элементарных потоков; дубликаты внутри программы редактор не пропускает. Уникальность PID между программами в этом режиме обеспечивает оператор — при коллизии узел пишет предупреждение в журнал, но мультиплекс продолжает работать. Зарезервированные PID источника (меньше 32) сохранить нельзя — таким назначается новый PID с предупреждением в журнале.

Если у потока-участника включена автоматическая раскладка PID ([Автоматическая раскладка PID](#)), редактор PID программы показывает уже назначенные ей идентификаторы, а не исходные, и сообщает об этом: пары переназначения, заданные до включения раскладки, перестают совпадать, и такие элементарные потоки получают PID автоматически. Поэтому опасный момент — включение раскладки у потока, уже входящего в мультиплекс, а не настройка с нуля.

Смена режима применяется на лету, без перезапуска потока.

Генерация таблиц

Мультиплексор формирует собственные таблицы PSI/SI: PAT, PMT каждой программы, SDT (имена сервисов и провайдера), NIT (идентификатор и имя сети, список сервисов, LCN) и TDT/TOT (метки времени). Таблица CAT формируется, когда программы содержат данные систем условного доступа.

Таблицы EIT (программа передач) мультиплексор не генерирует, а переносит из потоков-источников — в том числе EIT, сгенерированный на источнике из внешнего EPG ([Генератор EIT](#)).

Контроль мультиплекса

Состояние собранного мультиплекса отображается в окне статистики MPTS-потока ([Статистика потока](#)): выходная, целевая и входная скорость, число программ, таблица программ с битрейтом и состоянием каждой, идентификаторы TS и сети, состояние планировщика и заполнения.

Если суммарный битрейт программ превышает целевой, выводится предупреждение «Перегрузка битрейта — исходные программы превышают целевую скорость мукса». Недоразмеренный мультиплекс теряет пакеты программ — выпадения звука и изображения, ошибки непрерывности у получателей; при длительной перегрузке возможна и потеря таблиц PSI. Поднимите целевой битрейт или уменьшите битрейт программ (например, транскодированием источников, [Транскодеры](#)).

Соответствие мультиплекса TR 101 290 контролируется монитором анализатора ([Анализ по программам](#)).

1.7.6 Тестовые потоки

Генератор тестового потока программно формирует полноценный SPTS-поток с испытательной таблицей и звуковым сигналом — внешний источник не нужен. Применяется как заглушка эфира на время аварии основного источника, для проверки трактов передачи и приёмников, а также для нагрузочных и нагрузочных испытаний.

Тестовый источник — это вход типа test-stream у SPTS-потока (для MPTS недоступен). Генерацию выполняет модуль программного транскодирования — соответствующий пакет должен быть установлен на узле ([Транскодеры](#)). Поскольку это обычный вход, действуют общие правила резервирования: генератор ставят последним в списке входов — он автоматически подхватывает эфир при аварии всех основных источников; возврат на приоритетный вход выполняется по общим правилам повторной проверки ([Резервирование источников](#)).

Сгенерированный поток проходит обычный конвейер обработки ([Конвейер обработки](#)): его можно анализировать, раздавать по транспортным протоколам и в OTT, записывать в DVR, подавать в мультиплексор и на транскодер.

Настройки генератора

Задаются на входе test-stream (названия — как в интерфейсе; часть из них — дополнительные настройки):

Видео

Video encoder — кодек: MPEG-2, H.264 (по умолчанию) или HEVC; Video bitrate (kbps) — битрейт видео (по умолчанию 3000); Width и Height — разрешение (по умолчанию 1920 × 1080); Frame rate — частота кадров (по умолчанию 25); GOP interval — длина GOP в кадрах; Video pattern — вид испытательной таблицы; Width PAR / Height PAR — соотношение сторон пикселя.

Звук

Audio encoder — кодек: MPEG-1 Layer II (по умолчанию) или AAC; Audio waveform — вид звукового сигнала; Audio frequency (Hz) — частота тона; Audio volume — громкость (0...1).

Наложение

Enable overlay — включает наложение поверх таблицы (включено по умолчанию); Overlay text — произвольный текст, например название канала; Time format (strftime) — формат накладываемого времени, по умолчанию часы:минуты:секунды.

Транспортный поток

Total bitrate (kbps) — общий битрейт потока с заполнением NULL-пакетами (stuffing) до постоянного значения, 0 — без заполнения; PNR, PMT PID, Video PID, Audio PID — номер программы и идентификаторы таблицы PMT и элементарных потоков, чтобы вписать тестовый поток в план PID сети.

Экран «Тестовые потоки» веб-интерфейса ([Тестовые потоки](#)) находится в подготовке; тестовые входы настраиваются в редакторе потока ([Редактор входа и выхода](#)).

1.7.7 OTT и DVR

Подсистема OTT раздаёт потоки по протоколам на базе HTTP: HLS (на MPEG-TS), MPEG-DASH и Low-Latency HLS (на CMAF — фрагментированном MP4), а также MPEG-TS over HTTP. Поддерживаются HTTPS и HTTP/3 (QUIC). Раздача включается на вкладке «OTT» редактора потока ([Настроить поток](#)) и доступна только SPTS-потокам; программа мультиплекса выделяется в SPTS-поток демультиплексором ([Демультиплексор](#)).

DVR — постоянный архив потока на диске. Он пишется параллельно с OTT-раздачей — отдельная служба записи не требуется — и воспроизводится через те же URL, что и живое вещание; отличие только в параметрах запроса ([DVR: архив потока](#)).

Режимы раздачи

Прогрессивная раздача MPEG-TS по HTTP включается настройкой «Сервис HTTP» (значение Live): непрерывная передача исходного транспортного потока в одном HTTP-ответе.

Сегментная раздача управляется настройкой «Сервис HLS» (значения — как в интерфейсе):

Peering / HLS

Простая разбивка на сегменты. Режим предназначен для передачи потока между узлами Perfect Streamer: раздающий узел отдаёт HLS на MPEG-TS, принимающий подключается входом типа HLS. Субтитры WebVTT и DRM в этом режиме недоступны, запись DVR не настраивается.

OTT / HLS

Разбивка, оптимизированная для быстрого старта проигрывателей при OTT-вещании (нагрузка на CPU выше). Раздаётся HLS на MPEG-TS; действует выравнивание сегментов по GOP ([Сегментер](#)).

OTT / LL-HLS / DASH

Раздача на CMAF: поток формирует fMP4-сегменты, поверх которых отдаются MPEG-DASH и Low-Latency HLS. При включённой настройке «Устаревшие чанки MPEG-TS HLS» (включена по умолчанию) дополнительно раздаётся обычный HLS на MPEG-TS; её отключение экономит диск и CPU. MPEG-DASH и Low-Latency HLS доступны только в этом режиме.

Low-Latency HLS дробит медиаплейлист на частичные сегменты (parts): плеер начинает воспроизведение, не дожидаясь полного сегмента; применяются блокирующая перезагрузка плейлиста и подсказка предзагрузки. Целевая длительность части задаётся настройкой «Целевая длительность LL-части (ms)» (дополнительная настройка; по умолчанию 500 мс, применяется на лету, должна быть меньше минимального интервала чанка). Для точной низкой задержки манифест DASH содержит привязку медиавремени к UTC, а CMAF-сегменты несут метки времени продюсера.

Примечание: Упаковка CMAF переносит только звуковые дорожки AAC и AC-3; дорожки в других кодеках отбрасываются, о чём поднимается оповещение (*Оповещения (алертер)*).

Сегментер

В режимах OTT поток анализируется по таблицам PAT/PMT и ключевым кадрам, и сегменты нарезаются по критерию быстрого старта проигрывателей. Нарезка управляется настройками «Мин. интервал чанка (с)» и «Макс. интервал чанка (с)»: анализ начинается с минимального интервала, а если точка нарезки не найдена — сегмент принудительно завершается по максимальному.

«Сегменты, выровненные по GOP» (включено по умолчанию; действует в режиме OTT / HLS) — сегментер выравнивает границы сегментов по точкам случайного доступа и различает IDR и обычный I-кадр. Для источников с closed-GOP каждый сегмент гарантированно начинается с SPS/PPS/IDR — полной точки входа; «хвост» предыдущего GOP вырезается. Это устраняет чёрный кадр на старте VOD-сессий и приводит раздачу в соответствие требованиям HLS (RFC 8216). Тип GOP-структуры источника виден по метрикам анализатора (*Углублённые анализы*). Из-за выравнивания фактическая длительность сегмента может превышать минимальный интервал — заявляемая в плейлисте целевая длительность отражает фактический максимум.

URL и авторизация

Раздачу обслуживает потоковый HTTP-сервер узла (*Потоковый HTTP-сервер*). URL строится по схеме:

```
http://host:port/hls/<поток>/<логин>/<пароль>/index.m3u8
http://host:port/hls/<поток>/<логин>           — авторизация по токену
http://host:port/hls/<поток>/                   — авторизация по IP
```

Вместо /hls/ подставляется /dash/, /llhls/ или /http/; для адаптивных наборов — /hls/adaptive/ и аналогичные (*Адаптивные наборы*). <поток> — идентификатор потока либо имя, заданное настройкой «Имя потока в URL». Неавторизованный доступ запрещён: логины и разрешения клиентов настраиваются в веб-интерфейсе (*Пользователи / Логины*).

Готовые шаблоны URL с текущими режимами раздачи показываются в окне статистики потока — секция «URL доставки» с кнопкой копирования (*Статистика потока*). Активные клиентские сессии видны на экране «Клиенты» (*Клиенты*).

HTTP/3 (QUIC)

Потоковый HTTP-сервер умеет отдавать OTT-маршруты (HLS, DASH, LL-HLS, MPEG-TS over HTTP) поверх QUIC. HTTP/3 включается в настройках потокового HTTP-сервера (*Потоковый HTTP-сервер*); сертификат используется тот же, что и для HTTPS, поэтому HTTPS должен быть настроен.

Браузерные клиенты переходят на QUIC по стандартному объявлению Alt-Svc: объявление выдаётся по явному запросу клиента — параметром ?h3 на master-URL. Если QUIC-соединение не устанавливается (закрыт UDP-порт, недоверенный сертификат), клиент прозрачно остаётся на HTTPS — вещание не прерывается. Административные запросы обслуживаются только по TCP. Low-Latency HLS и DASH поверх QUIC отдаются инкрементально — части уходят клиенту по мере готовности.

Субтитры WebVTT

Настройка «Субтитры телетекста в WebVTT» (включена по умолчанию, действует в режимах OTT) преобразует субтитры DVB Teletext / DVB Subtitling входного потока в дорожки WebVTT: в master-плейлист HLS добавляется дорожка субтитров, в манифест DASH — текстовый набор. Наличие субтитров в источнике видно в информации о медиа в окне статистики. Субтитры архивируются в DVR параллельно с медиасегментами и доступны при воспроизведении архива; защитой DRM они не шифруются. Для диагностики служит настройка «Логировать сообщения субтитров» (дополнительная настройка).

Адаптивные наборы

Адаптивный набор объединяет несколько OTT-потоков одного канала с разными битрейтами под одним master-плейлистом HLS или манифестом DASH — плеер сам переключается между вариантами по пропускной способности.

Набор создаётся как поток типа «Адаптивный» (*Новый поток*, редактор набора — *Адаптивный набор*). Участники — SPTS-потоки с включённым режимом OTT: OTT / HLS для адаптивного HLS на MPEG-TS, OTT / LL-HLS / DASH для адаптивных DASH и Low-Latency HLS. Для каждого участника можно задать битрейт вручную; значение 0 — берётся измеренный битрейт.

Master-URL набора строятся по схемам `/hls/adaptive/`, `/dash/adaptive/`, `/llhls/adaptive/` с теми же вариантами авторизации. Доступ, разрешённый клиенту к адаптивному набору, автоматически распространяется на все входящие в него потоки.

Защита контента (DRM)

OTT-раздачу потока можно шифровать. Шифрование выполняется один раз, в момент формирования сегмента: живая раздача и DVR-архив получают одни и те же зашифрованные данные; архив хранится на диске в зашифрованном виде. Настройки собраны в секции «Защита контента» вкладки «OTT»; применение любой из них перезапускает OTT-раздачу потока.

Режимы (поле «DRM», значения — как в интерфейсе):

HLS AES-128

Полносегментное шифрование TS-сегментов; расшифровку выполняет любой стандартный HLS-плеер. Требует «Сервис HLS» = OTT / HLS.

CENC cenc (DASH), CENC cbcs (DASH)

Побитовое шифрование MPEG Common Encryption (ISO/IEC 23001-7) — схема cenc (AES-CTR) либо cbcs (паттерновый AES-CBC, экосистема Apple FairPlay). Требуют режим OTT / LL-HLS / DASH с отключённой раздачей TS-чанков; раздача — только по MPEG-DASH.

Любой режим защиты требует выключенного «Сервиса HTTP» — прогрессивная раздача идёт мимо шифрования. Несовместимую комбинацию настроек сервер отклоняет с указанием причины.

Ключи и доставка:

- «Ключ контента (32 hex)» — ключ шифрования (для статического AES-128 и CENC). «ID ключа (CENC KID)» — идентификатор ключа CENC; пустое поле — идентификатор детерминированно выводится из идентификатора потока.
- «Доставка ключа» (AES-128, значения — как в интерфейсе): Built-in (PSS serves the key) — сервер сам отдаёт ключ по сессионному URL с той же авторизацией, что и сегменты; External key server — тег ключа в плейлисте указывает на «URI ключа (EXT-X-KEY)», встроенная доставка

отключается. Для CENC сервер ключ плеерам не выдаёт — доставка ключей выполняется DRM-инфраструктурой (лицензионные серверы).

- Ротация ключей (только AES-128): «Источник ключа» = Derived (rotating), «Секрет ротации» (не менее 16 символов) и «Окно ключа (минуты)». Ключ каждого окна вычисляется из секрета необратимой функцией; DVR-архив воспроизводится через любое число ротаций — номер окна восстанавливается по времени записи сегмента.

Предупреждение: Смена «ключевой идентичности» потока — ключа контента, секрета или настроек ротации — делает ранее записанный зашифрованный архив нечитаемым; веб-интерфейс запрашивает подтверждение таких изменений. Ключи хранятся в конфигурации сервера в открытом виде — ограничивайте доступ к файлам конфигурации и резервным копиям.

DVR: архив потока

Запись включается на вкладке «OTT», секция «DVR» (видна в режимах OTT):

«Хранилище DVR»

Хранилище, в которое пишется архив ([Хранилища DVR](#)). Значение «Нет» — запись выключена.

«Хранение (часы)»

Глубина архива потока, от 1 часа до 90 суток. Более старые сегменты удаляются регулярной очисткой.

«Защищённый минимум (часы)»

Нижний порог защиты: очистка никогда не удаляет записи моложе этого значения, даже при нехватке места на диске.

Запись начинается автоматически, как только поток переходит в рабочее состояние. Отвязка от хранилища останавливает запись (файлы остаются на диске, воспроизведение архива прекращается); при смене хранилища запись начинается заново, файлы не переносятся. Деструктивные изменения веб-интерфейс подтверждает диалогом «Подтвердите изменения DVR».

Хранилища DVR

Хранилища настраиваются на экране «Хранилища DVR» ([Хранилища DVR](#)). Для каждого задаются путь к каталогу (после создания не меняется), предел использования диска в процентах и — в дополнительных настройках — интервал очистки, отсрочка и порция обрезки при переполнении, аварийный запас места и гистерезис. На один диск рекомендуется одно хранилище: несколько хранилищ на общем диске конкурируют за свободное место. Удаление хранилища из конфигурации не удаляет файлы с диска.

Очистка архива работает на нескольких уровнях:

- по глубине — сегменты старше значения «Хранение (часы)» каждого потока;
- по месту — при устойчивом превышении предела использования диска удаляются самые старые сегменты пропорционально по всем потокам, но не моложе защищённого минимума;
- аварийная — при критической нехватке места запись останавливается до восстановления свободного места;

- очистка осиротевших файлов внутри архивов записываемых потоков выполняется автоматически; каталоги архивов, оставшиеся от удалённых потоков, удаляются только вручную — кнопкой «Очистить осиротевшие архивы» на экране «Монитор DVR» ([Монитор DVR](#)).

Состояние хранилищ и записываемых потоков — заполнение, объём архива, задержки записи и чтения, активные задачи очистки — отображается на том же экране «Монитор DVR». Архив отдельного потока можно безвозвратно удалить кнопкой «Стереть архив DVR» в окне статистики потока; запись при этом продолжается.

Воспроизведение архива (VOD)

Архив воспроизводится через те же URL, что и живое вещание HLS / DASH, — отличается только параметрами запроса:

t=<время>

Начало VOD-окна (Unix-время, секунды). $t=0$ — от начала архива. Наличие параметра t включает VOD-режим.

d=<секунды>

Длительность VOD-окна. Без параметра (или $d=0$) — до текущего момента. Действует только вместе с t .

epg=<время>

VOD по программе передач: сервер находит EPG-событие, активное в указанный момент, и берёт его начало и длительность как границы окна. Требует привязки потока к источнику EPG — поля «Источник EPG» и «Канал EPG» на вкладке «Поток» ([Модификация потока](#), [Генератор EIT](#)); параметры t и d при этом игнорируются. Для каталога передач интерфейсу достаточно времени события — точные границы вычислит сервер.

Особенности воспроизведения:

- Границы окна нормализуются: запрос раньше начала архива подтягивается к первому доступному сегменту; окно целиком вне архива возвращает пустой, но валидный плейлист.
- HLS-плейлист VOD — закрытый (с признаком завершения); DASH-манифест — статический. Разрывы записи в DASH оформляются отдельными периодами — плееры перематывают через границу без специальных настроек.
- Если клиент доходит до правой границы окна, недостающие сегменты отдаются из живой памяти потока — без переадресаций и повторной авторизации.
- Открытая VOD-сессия защищает своё окно от регулярной очистки до закрытия сессии; аварийная очистка и защищённый минимум имеют приоритет.
- Субтитры WebVTT воспроизводятся и в архиве ([Субтитры WebVTT](#)).
- Адаптивные наборы поддерживают те же параметры VOD. В манифест DASH попадают только варианты с привязанным хранилищем DVR; в master-плейлист HLS входят все варианты, но варианты без архива на запрос архива отвечают ошибкой, и плеер их пропускает.

1.7.8 Транскодеры

Транскодеры перекодируют видео и звук потока: снижают битрейт для ОТТ или мультиплекса, меняют кодек и разрешение, готовят варианты для адаптивных наборов. Каждый транскодер работает как отдельный процесс узла; поддерживается схема «один ко многим» — с одного декодера можно получить несколько выходных потоков с разными настройками кодирования. Транскодирование доступно только SPTS-потокам; в исходном потоке должны быть и видео, и звук.

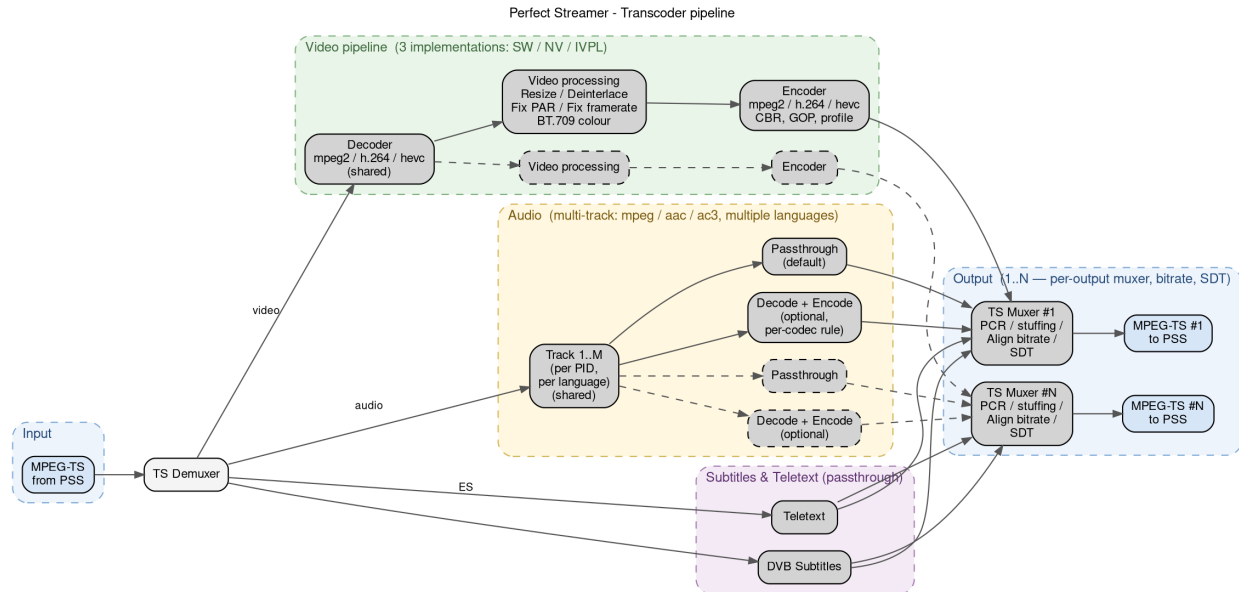


Рис. 4: Архитектура транскодера: один декодер (общий) и N независимых ветвей «обработка + кодер»; звуковые дорожки обрабатываются отдельно от видео, субтитры и телетекст проходят насквозь.

Подключение

Транскодер соединяет два и более потока:

1. На потоке-источнике добавляется **выход типа transcoder** — это декодер. В его настройках выбирается бэкенд транскодирования (см. ниже).
2. На потоке-результате добавляется **вход типа transcoder** — это кодер. В его настройках выбирается работающий декодер-источник (поле Decoder).
3. Шаг 2 повторяется для каждого дополнительного варианта — все они используют общий декодер.

Дальше поток-результат — обычный SPTS-поток: его можно раздавать в ОТТ, записывать в DVR, подавать в мультиплексор.

Бэкенды и кодеки

Бэкенд выбирается на декодере (поле Transcoder backend, значения — как в интерфейсе):

Video pass-through

Видео проходит без перекодирования, транскодируется только звук. Режим выполняется программным транскодером — пакет программного транскодирования должен быть установлен ([Транскодеры](#)).

Software (CPU)

Программное транскодирование на процессоре.

NVidia (GPU)

Аппаратное транскодирование на графических процессорах NVIDIA.

Intel VPL (GPU)

Аппаратное транскодирование на графических процессорах Intel.

Для GPU-бэкендов устройство выбирается полем GPU device selector (дополнительная настройка) — актуально на серверах с несколькими картами. Пакеты транскодеров устанавливаются отдельно ([Транскодеры](#)); установленные бэкенды, их версии и найденные устройства показаны на экране «О программе» ([О системе](#)).

Кодеки декодера — MPEG-2, H.264, HEVC; звук — MPEG (Layer 1–3), AAC, AC-3. Кодеки кодера — H.264 и HEVC (кодер NVIDIA не поддерживает MPEG-2; MPEG-2 кодируют бэкенды Software и Intel VPL); звук — MPEG Layer II и AAC.

Особенности декодера:

- чересстрочные (interlace) потоки поддерживаются на входе и выходе; для H.264 и HEVC формат interlace alternate (отдельные поля) преобразуется в interleaved;
- для HEVC поддерживается профиль Main10 с BT.709 (SDR) и BT.2020 (HDR); кодер HEVC всегда использует профиль Main с BT.709;
- переменная частота кадров (VFR) преобразуется в постоянную;
- поддерживаются потоки без IDR-кадров.

Настройки декодера

Задаются на выходе типа transcoder потока-источника (названия — как в интерфейсе; все перечисленные ниже — дополнительные настройки):

Fix PAR (N/D), Fix framerate (N/D)

Подсказки для исправления данных входного потока — они не преобразуют поток, а корректируют параметры, когда источник их не сообщает или сообщает неверно. Fix PAR — соотношение сторон пикселя дробью (например, 16/9 для Wide SD); Fix framerate — частота кадров дробью, если она отсутствует в потоке (PAL — 25/1, NTSC — 30000/1001, кино — 24/1).

Convert to BT.709

Конвертация цветовых форматов SD (BT.470-2 PAL, SMPTE 170M NTSC) в BT.709.

Watchdog timeout

Таймаут сторожевого перезапуска процесса транскодера.

Процесс транскодера всегда ведёт подробный журнал — отдельная настройка трассировки не требуется ([Контроль работы](#)).

Настройки кодера

Задаются на входе типа transcoder потока-результата:

Encoder

Видеокодек: MPEG-2, H.264 (по умолчанию) или HEVC.

Video profile (H.264)

Профиль кодирования H.264: Main или High.

Video bitrate (kbps)

Битрейт видео. Кодирование всегда выполняется в режиме CBR; суммарный битрейт потока будет больше за счёт звуковых дорожек.

Resize

Изменение размера картинки: пропорциональное уменьшение в 2 или 4 раза; приведение к SD — варианты SD, SD PAL и SD NTSC, все с широкоэкранным соотношением 16:9; масштабирование SD в HD; задание ширины или высоты (вторая сторона пересчитывается пропорционально). Произвольный размер не поддерживается — это исказило бы пропорции.

Deinterlace

Преобразование чересстрочного видео в прогрессивное (включено по умолчанию; при масштабировании SD в HD применяется автоматически).

Настройки качества и структуры потока (дополнительные настройки):

GOP interval

Интервал ключевых кадров в кадрах. По умолчанию 25 — секунда для 25p; рекомендуется, если проигрыватели подключаются к потоку в произвольный момент.

B-frames, Lookahead

Повышают качество кодирования; рекомендуемые значения — 3 и 20–50 кадров соответственно.

Speed preset

Предустановка скорости кодирования от 1 до 7: меньше — выше качество и больше нагрузка. По умолчанию 4.

Framerate resample

Прореживание частоты кадров в 2 или 4 раза.

Encoder PMT PID, PAT/PMT interval (90kHz)

PID таблицы PMT и интервал повторения таблиц PAT/PMT в выходном потоке — при необходимости вписать поток в план сети.

Несовместимые сочетания параметров возможны — ошибки видны в журнале транскодера (*Контроль работы*).

Обработка звука

По умолчанию все звуковые дорожки проходят со входа на выход без перекодирования. Лишние дорожки убираются PID-фильтрами на входе потока (*Фильтрация MPEG-TS*).

Перекодирование звука настраивается правилами отдельно для каждого исходного кодека (дополнительные настройки): дорожку можно пропустить без изменений, перекодировать (MPEG — в AAC; AAC — в MPEG Layer II; AC-3 — в MPEG Layer II или AAC) либо убрать (значение Skip). Дополнительно задаются битрейт звука (Audio bitrate, по умолчанию 128 кбит/с) и громкость понижающего микширования 5.1 (5.1 downmix volume). Если после применения правил в выходном потоке не останется ни одной звуковой дорожки, кодер сообщит об ошибке в журнале.

PCR и битрейт выходного потока

Мультиплексор транскодера формирует новые, согласованные метки PCR. По умолчанию выходной поток — VBR: суммарный битрейт следует фактическому битрейту видео и звука, и точность PCR по TR 101 290 на приёмной стороне не гарантируется. Если транскодированный поток предназначен для DVB-вещания, включите режим CBR на потоке-приёмнике — заполнение выравнивает битрейт и пересчитает PCR ([Управление битрейтом](#)).

Контроль работы

Экран «Транскодеры» ([Транскодеры](#)) показывает все экземпляры: каждая строка — отдельный экземпляр (декодер и подключённые к нему кодеры) с состоянием, типом бэкэнда, временем работы, загрузкой CPU и памятью. Окно «Экземпляр транскодера» ([Экземпляр транскодера](#)) раскрывает источник, список кодеров с параметрами выходного видео, ресурсы процесса и журналы текущего и предыдущего запусков — первый адрес при диагностике проблем.

Загрузка GPU-устройств отображается на экране «Системный монитор» ([Системный монитор](#)). Превышение порогов CPU, памяти и загрузки GPU, а также недоступность настроенного бэкэнда вызывают оповещения ([Каталог кодов алертов](#)).

1.7.9 DVB-приёмник

Perfect Streamer работает с DVB-адаптерами, установленными в системе: поддерживаются стандарты DVB-S, DVB-S2, DVB-T, DVB-T2, DVB-C и ATSC. Дополнительно реализованы декапсуляция T2-MI (ETSI TS 102 773) и дескремблирование BISS-1 / BISS-E и CI/CAM. Основное условие — корректно установленный и работающий драйвер адаптера; экран «DVB-адаптеры» ([DVB-адаптеры](#)) доступен, когда в системе обнаружены DVB-устройства. Устройства DVB-ASI подключаются иначе — файловым входом потока по пути устройства, без записи DVB-адаптера.

Адаптеры

Адаптер — это запись, привязывающая физический тюнер (пару «адаптер/устройство» в системе) к параметрам приёма; при захвате сигнала адаптер отдаёт MPTS-мультиплекс транспондера. Адаптер добавляется в окне «Добавить DVB-адаптер» ([Редактор DVB-адаптера](#)):

- **Режим:** Stream — приём потока; Frontend monitor — только мониторинг сигнала, без чтения потока (устаревшее значение Scanner не используется). Режим и «Система вещания» задаются при создании и не меняются.
- **Оборудование** — тюнер выбирается из списка обнаруженных в системе; занятые помечаются.
- **Параметры настройки** — зависят от системы вещания: частота (для спутника — в МГц, для эфира и кабеля — в кГц), «Символьная скорость, ksym/s», «FEC», «Модуляция», «Спектральная инверсия», «Stream id (ISI/PLP)» — идентификатор потока DVB-S2 multistream или PLP DVB-T2 на уровне тюнера; для эфирного приёма — «Полоса пропускания», «Режим передачи», «Защитный интервал», «Иерархия».

Для спутниковых систем настраивается секция «LNB»: частоты гетеродинов (универсальный Ku — 9750/10600, переключение 11700 МГц; для круговой поляризации и C-диапазона — свои значения), «Поляризация» (адаптер управляет питанием конвертера: 18 В — горизонтальная, 13 В — вертикальная), «Порт DiSEqC» и «Принудительный тон 22 kHz».

«Совместное использование LNB» отключает управление питанием, поляризацией и тоном — их выставляет другой приёмник, владеющий конвертером. Режим используется, когда несколько тюнеров подключены к одному LNB через разветвитель (принимать можно только поляризацию и диапазон, выбранные владельцем), а также чтобы разнести разные PLP одного транспондера по отдельным адаптерам.

Ключи дескремблирования и список программ CAM применяются на лету; изменения параметров настройки перенастраивают фронтенд. Настройка «Экспорт EPG» записывает программу передач мультимплекса в базу EPG узла ([Импорт EPG/XMLTV](#)).

Подключение к потокам

Принятый мультимплекс используется двумя способами:

- **целиком** — вход типа dvb у MPTS-потока: весь транспондер уходит в транзит или на пересборку ([MPTS-потоки](#));
- **отдельными программами** — вход-демультиплексор у SPTS-потоков ([Демультиплексор](#)): программа выбирается по PNR из списка обнаруженных сервисов.

В списке программ адаптера видно, какие программы уже потребляются: пометка «Активна» с уточнением — извлекается как отдельная программа (SPTS) или читается в составе полного мультимплекса (MPTS). Пока на адаптер ссылается хотя бы один поток, удалить его нельзя.

Сканирование

Чтобы не вводить параметры приёма вручную, встроен сканер транспондеров (окно «Сканирование DVB», [Сканирование DVB](#)). Сканер обходит транспондеры выбранного спутника или регионального диапазона, выполняет захват каждого, собирает таблицы PSI/SI и формирует список мультимплексов с программами: PNR, имя сервиса, провайдер, признак скремблирования и основные PID. Любой найденный мультимплекс одной кнопкой превращается в настроенный адаптер; можно выбрать и добавить сразу несколько.

Особенности:

- источник частот — «Каталог» (справочники спутников и региональных диапазонов) либо «Слепое сканирование» по синтезированной сетке частот с необязательным диапазоном и шагом;
- для спутника выбирается «Пресет LNB» (универсальный Ku, круговая поляризация, C-диапазон) или свои частоты гетеродинов;
- сканер занимает физический тюнер целиком — нужен тюнер, не занятый ни системой, ни другими активными записями; приостановленная запись тюнер освобождает, поэтому работающий адаптер можно временно поставить на паузу и просканировать; свободные и занятые тюнеры видны в списке;
- внутрь T2-MI сканер не заходит: параметры декапсуляции задаются уже в настройках созданного адаптера.

Декапсуляция T2-MI

T2-MI (T2-Modulator Interface, ETSI TS 102 773) — формат транспортировки потоков DVB-T2 по распределительным каналам, обычно спутниковым DVB-S/S2 (в том числе multistream): внешний транспондер переносит одну или несколько несущих T2-MI, каждая из которых инкапсулирует BBFRAME с одним или несколькими PLP. После декапсуляции извлекается внутренний MPEG-TS с программами и таблицами PSI/SI.

Perfect Streamer работает в режиме multi-carrier: один адаптер одновременно отдаёт внешний мультиплекс и все декапсулированные несущие T2-MI. Настройки (секция «T2-MI»):

«Режим T2-MI»

Off — декапсуляция выключена, внешний поток передаётся как есть (при обнаружении T2-MI в мультиплексе узел подсказывает об этом в журнале); Manual — декапсуляция всегда включена; Auto — несущие обнаруживаются автоматически по таблицам PMT, без них адаптер работает как обычный мультиплекс.

«T2-MI PID»

Предварительное создание несущей на известном PID до прихода таблиц; остальные несущие всё равно обнаруживаются автоматически.

«T2-MI PLP»

Номер PLP, извлекаемый из каждой несущей; отдельного значения на несущую нет — для разных PLP настраиваются отдельные адаптеры с совместным использованием LNB. Если BBFRAME с заданным PLP не приходят, узел пишет в журнал список наблюдаемых PLP; он же виден в статистике адаптера. Поле «Фильтр T2-MI TSID (-1 = любой)» зарезервировано.

Программы внутренних мультиплексов появляются в общем списке программ адаптера с пометкой PLP. Для подключения такой программы к SPTS-потoku используется составной PNR — номер несущей × 65536 + PNR внутри неё (*Демультимплексор*); несущая 0 — внешний мультиплекс. Служебные дубликаты и тестовые фиды внутри несущих автоматически помечаются «[test]» и не влияют на проверку здоровья.

Дескремблирование

BISS. Программный дескремблер BISS-1 / BISS-E (секция «Дешифрование BISS»). Задаются пары «слот — ключ»; на одном адаптере может быть активно несколько дескремблеров:

- слот — номер программы (PNR) внешнего мультиплекса: расшифровываются все PID программы;
- слот — PLP несущей T2-MI: ключ применяется ко всей несущей до декапсуляции. Это обязательно для зашифрованных multistream-потоков — без ключа декапсулятор отбрасывает каждый зашифрованный пакет.

Формат ключа определяется по длине: 12 шестнадцатеричных цифр — BISS-1 (контрольные байты вычисляются автоматически), 16 — BISS-E. Ключи применяются на лету, без перезапуска адаптера. Работу ключа видно по счётчикам в статистике адаптера: рост «расшифровано» — ключ работает; рост «нет ключа» — ключ не задан или задан не на том уровне; рост «ошибки» — ключ неверен.

CI/CAM. Адаптер с установленным CAM-модулем дескремблирует выбранные программы через Common Interface (секция «Дешифрование CI/CAM»): перечисляются номера программ, модуль расшифровывает их прямо в тракте приёма. Список применяется на лету и не зависит от системы вещания. Состояние модуля, поддерживаемые CA-системы и статус дешифрования каждой программы видны в статистике адаптера и на экране «Оборудование» (*Оборудование*); извлечение модуля, отсутствие подписки и ошибки CA вызывают оповещения (*Каталог кодов алертов*).

«Удалять CA из расшифрованного выхода» (дополнительная настройка) очищает таблицы и сообщения CAS из уже расшифрованных программ — аналогично фильтру потока «Удалять CAT / ECM / EMM» (*Фильтрация MPEG-TS*).

Контроль сигнала

В списке адаптеров видны захват сигнала, уровень, SNR, BER и входной битрейт каждого адаптера. Окно статистики адаптера (*Статистика адаптера*) показывает историю сигнала (уровень, SNR, BER, неисправленные блоки), текущие метрики фронтенда, список программ с медиаданными, счётчики несущих T2-MI и дескремблеров, состояние CAM-модуля и журнал; накопительные счётчики сбрасываются кнопкой «Сбросить статистику». Для контроля антенны без чтения потока адаптер создаётся в режиме Frontend monitor.

Потеря захвата, падение уровня или качества сигнала, рост ошибок BER и неисправленных блоков вызывают оповещения с настраиваемыми порогами (*Оповещения (алертер)*), пороги — в настройках оповещателя).

Буфер приёма. «Буфер демультимплексора (блоки)» (дополнительная настройка) задаёт размер кольцевого буфера приёма в блоках по 64 КБ. Значение по умолчанию 512 (32 МБ) покрывает полнотранспондерные сценарии DVB-S/S2 с несколькими потребителями; для слабонагруженных или встраиваемых систем его можно уменьшить. При переполнениях буфера (счётчик виден в статистике, сообщение — в журнале) увеличьте буфер или сократите объём принимаемых данных — например, откажитесь от чтения полного мультиплекса, если он не нужен. Учитывайте память: каждый адаптер резервирует «блоки × 64 КБ».

Права доступа на устройства

Если DVB-адаптеры в Perfect Streamer не отображаются, а в системе они есть, выдайте службе права на устройства DVB. Создайте файл `/etc/udev/rules.d/99-dvb-permissions.rules` со строкой:

```
SUBSYSTEM=="dvb", GROUP="video", MODE="0660"
```

Затем добавьте пользователя службы в группу video и примените права:

```
sudo usermod -aG video pss
sudo chown -R root:video /dev/dvb/*
sudo systemctl restart pss
```

После перезагрузки системы права выставит правило udev автоматически.

1.7.10 Публикация и приём RTMP

Perfect Streamer работает с RTMP и RTMPS как клиент: публикует SPTS-поток на внешнюю точку приёма (ingest видеоплатформы или любой RTMP-приёмник) и принимает поток с RTMP-источника по запросу (pull). Режим приёма входящих публикаций (push ingest, «как у видеоплатформ») не поддерживается — для таких сценариев используйте мост через внешнее приложение (*Прочие входы и выходы*). RTMP доступен только SPTS-потокам.

Публикация (выход типа rtmp)

Выход добавляется на вкладке «Выходы» потока (названия полей — как в интерфейсе):

URL (rtmp(s)://...)

Адрес точки приёма, включая ключ трансляции — как его выдаёт принимающая платформа.

Bind interface

Локальный интерфейс исходящего соединения.

Client timeout

Таймаут подключения, по умолчанию 10 секунд.

Allow HEVC (Enhanced RTMP)

HEVC публикуется через расширение Enhanced RTMP (включено по умолчанию; дополнительная настройка). Для приёмников без поддержки Enhanced RTMP выключите — публиковаться будет только H.264.

Verify TLS certificate

Для RTMPS: проверка сертификата сервера (по умолчанию выключена; дополнительная настройка).

Формат публикации — одна программа: одно видео H.264 или HEVC и не более одной звуковой дорожки AAC. Многодорожечный источник сужается PID-фильтрами на входе потока ([Фильтрация MPEG-TS](#)); звуковые дорожки в других форматах отбрасываются. Если раскладка дорожек несовместима с RTMP, публикация не выполняется — причина отображается в статистике выхода ([Статистика потока](#)).

Приём (вход типа rtmp)

Вход типа rtmp забирает поток с RTMP-источника по его URL: узел подключается как клиент, принятый поток ремultiplexируется в MPEG-TS и проходит обычный конвейер SPTS ([Конвейер обработки](#)). Настройки те же: URL (rtmp(s)://...), Bind interface, Client timeout и Verify TLS certificate для RTMPS.

1.7.11 Прочие входы и выходы

Подробности настройки входов и выходов, кратко перечисленных в обзоре протоколов ([Другие входы и выходы](#)): приём с RTSP-источников и по HLS/HTTP, работа с файлами и устройствами, именованные каналы и мост к внешним приложениям. Названия полей — как в интерфейсе.

Вход RTSP

Приём видеопотока с RTSP-источников — IP-камер и RTSP-серверов. Узел открывает RTSP-сессию, читает элементарные потоки и ремultiplexирует их во встроенный MPEG-TS; дальше поток проходит обычный конвейер SPTS. Если в источнике нет звуковых дорожек, добавляется тихая дорожка MPEG-1 Layer II — конвейеру нужна хотя бы одна звуковая дорожка. RTSP — однопрограммный источник, доступен только SPTS-потокам.

Настройки:

- URI ([rtsp://...](#)) — полный адрес сессии; Login и Password — учётные данные, если сервер требует авторизацию;

- Transport — транспорт RTP внутри RTSP: UDP (по умолчанию; низкая задержка, чувствительность к потерям), TCP (проходит NAT и межсетевые экраны) или HTTP-туннель (для прохода через HTTP-прокси);
- User-Agent и Cookies (дополнительные настройки) — для серверов с нестандартной авторизацией;
- Client timeout — таймаут открытия и чтения, по умолчанию 10 секунд.

Вход HLS / HTTP

Приём MPEG-TS поверх HTTP: обычный HLS-плейлист либо непрерывная HTTP-передача. Режим по умолчанию определяется автоматически по типу содержимого ответа; при необходимости задаётся явно. Для адаптивного плейлиста по умолчанию берётся первый вариант из master-плейлиста; номер варианта можно задать явно (дополнительная настройка). Поддерживаются авторизация логином и паролем, Cookies и собственный User-Agent, а также плейлисты, перенаправляющие на плейлист новой сессии. Вход доступен только SPTS-потокам; для MPTS используйте UDP / RTP / TCP, файл или DVB-адаптер.

Файлы и устройства

Тип file служит и входом, и выходом:

- **Выход** — запись в TS-файл (например, для последующей диагностики внешними анализаторами) или вывод в устройство: любое устройство в /dev, в том числе платы SDI и ASI. Для устройства включается признак `Is device node`.
- **Вход** — цикличное воспроизведение из TS-файла либо захват с устройства (например, приём DVB-ASI — *DVB-приёмник*).

Путь задаётся полем File path — полный путь к файлу или устройству.

Именованные каналы (pipe)

Чтение из существующего именованного канала (FIFO) и запись в него. В отличие от типа std, узел не запускает дочерний процесс: внешняя программа-источник или получатель должна быть запущена независимо и держать канал открытым со своей стороны. Поле FIFO path (empty = auto) задаёт путь к существующему FIFO (создаётся, например, командой `mkfifo`); при пустом значении узел сам создаёт FIFO и назначает путь автоматически (путь виден в статистике входа или выхода). Доступен для SPTS и MPTS.

Внешние приложения (std)

Тип std — мост к протоколам, не поддерживаемым встроенными средствами: поток MPEG-TS принимается и передаётся через стандартные потоки ввода-вывода стороннего консольного приложения, которое узел запускает и контролирует сам (перезапуская при завершении).

Контракт приложения:

- **вход** — приложение пишет MPEG-TS в stdout; диагностические сообщения выводите только в stderr — stdout зарезервирован для потока;
- **выход** — приложение читает MPEG-TS из stdin; настройка Packets per write задаёт пакетирование записи (1–16 TS-пакетов за операцию, дополнительная настройка).

Настройки: Command (absolute path) — абсолютный путь к приложению; Arguments — командная строка (кавычки сохраняют пробелы внутри аргумента, обратная косая черта экранирует символы); Env names и Env values — пары переменных окружения, передаваемых приложению.

1.7.12 EPG/EIT

Узел ведёт базу данных EPG — программу передач, собираемую из принимаемых потоков и внешних источников XMLTV. Данные базы используются в нескольких подсистемах: генерация таблиц EIT в SPTS-потоках ([Генератор EIT](#)), воспроизведение архива DVR по программе передач ([Воспроизведение архива \(VOD\)](#)) и выдача программы внешним системам — приставкам и OTT middleware ([EPG для OTT middleware](#)).

Импорт EPG/XMLTV

База EPG наполняется из источников трёх видов:

- **EIT из принимаемых потоков** — настройка «Извлекать EIT в базу данных EPG» у потока (SPTS и MPTS, [Модификация потока](#));
- **EIT с DVB-адаптеров** — настройка «Экспорт EPG» адаптера ([DVB-приёмник](#));
- **внешние источники XMLTV** — по ссылке на веб-ресурс или из локального файла.

EIT, извлечённый из потоков и с DVB-адаптеров, накапливается в общем встроенном источнике «поток-импорт»; каждый внешний источник XMLTV ведёт собственную базу. Идентификатор канала уникален в пределах своего источника; названия каналов и тексты событий хранятся на нескольких языках.

Настройка источников

Источники настраиваются на экране «Настройки EPG» ([Настройки EPG](#)). На вкладке «Импорт» — общие параметры: «Включить импорт», «Хранить историю, дней» (глубина хранения прошедших событий) и «Автоочистка осиротевших каналов» (удаление каналов, у которых не осталось событий).

Внешние источники XMLTV добавляются на вкладке «Источники»: имя, «URL / путь источника», «Включено», «Интервал обновления, с» (по умолчанию раз в час), «Кодировка контента» (автоопределение или принудительно GZip), учётные данные и Cookies для закрытых веб-источников, «Сохранять загруженную копию» для диагностики.

База данных EPG

Накопленная программа передач просматривается на экране «База данных EPG» ([База данных EPG](#)). На вкладке «База данных» выбирается источник, доступны поиск каналов и фильтр по набору каналов; расписание канала листается по дням, текущая передача подсвечена. Для каждого канала можно изменить:

- «Название канала» — имя для экспорта XMLTV;
- «Часовой пояс» — если время событий при импорте не было привязано к UTC;
- «URL значка» — иконка канала;

- «Наборы каналов» — членство в наборах для выдачи внешним потребителям (*EPG для OTT middleware*).

Ручные правки могут быть перезаписаны при следующем обновлении источника — веб-интерфейс предупреждает об этом.

Вкладка «Состояние источников» показывает ход импорта: время последнего и следующего обновления, число каналов и событий, ошибки; кнопка «Обновить сейчас» запускает обновление источника немедленно, не дожидаясь расписания.

Генератор EIT

Генератор EIT вставляет в SPTS-поток таблицы программы передач из базы EPG узла — и текущую/следующую передачу, и расписание. Это позволяет формировать полноценный DVB-сигнал для потоков, источник которых EIT не несёт: программа берётся из любого источника базы EPG (*Импорт EPG/XMLTV*).

Генерация включается в секции «Генерировать EIT из EPG» вкладки «Поток» редактора потока: выбираются «Источник EPG» и «Канал EPG» (*Модификация потока*). Если источник не несёт SDT, вместе с EIT автоматически генерируется и таблица описания сервиса. Состояние генератора отображается в окне статистики потока — секция «Генерация EIT»: число событий, время последнего обновления и ошибки источника EPG.

Особенности:

- Вставка таблиц меняет состав пакетов потока: для соответствия TR 101 290 на выходе включите режим CBR (*Управление битрейтом*).
- Если источник уже несёт собственный EIT, его можно удалить фильтром «Удалять EIT» и заменить сгенерированным (*Фильтрация MPEG-TS*).
- При сборке мультимплекса EIT программ-участников — в том числе сгенерированный — переносится мультимплексором; идентификаторы таблиц приводятся к плану мультимплекса (*Генерация таблиц*).
- Программа передач, вставленная генератором, используется и для воспроизведения архива DVR по событию (*Воспроизведение архива (VOD)*).

EPG для OTT middleware

Программа передач из базы EPG выдаётся внешним системам — приставкам, приложениям и OTT middleware — двумя способами: полным документом XMLTV через выделенный EPG-сервер и суточной программой канала в JSON через HTTP API узла.

Выдача XMLTV

XMLTV раздаёт отдельный встроенный EPG-сервер. Он включается на экране «EPG-сервер» (*EPG-сервер*): «Включить EPG-сервер», HTTP-порт (по умолчанию 43973), при необходимости HTTPS с сертификатом (порт по умолчанию 43983) и «Имя хоста».

Доступ — по учётным записям (вкладка «Учётные записи» экрана «Настройки EPG», *Настройки EPG*): логин и пароль, «Набор каналов», «Дата окончания» действия, «Разрешённый IP» и признак «Приостановлен». Состав выдачи определяется набором каналов:

- наборы создаются на вкладке «Наборы каналов»;

- каналы включаются в наборы в базе данных EPG ([База данных EPG](#)); канал может входить в несколько наборов;
- учётной записи назначается один набор — без него выдача пуста.

Документ доступен по пути /xmtv (и /xmtv.gz — сжатый файл); авторизация — HTTP Digest либо через параметры запроса. Время передач выдаётся с часовым поясом канала; в документ попадают только текущие и будущие события. Если один и тот же идентификатор канала пришёл из разных источников, канал выдаётся один раз, о пропуске сообщает журнал.

Суточная программа в JSON

Для интерфейсов middleware узел отдаёт программу канала за сутки одним запросом к HTTP API в формате JSON (авторизация — как у остальных запросов API):

```
GET /data/epg/channel?src=<источник>&ch=<канал>&lang=<язык>&t=<время>
```

Возвращается список событий суток (по UTC), в которые попадает время t: начало, конец, название и описание каждой передачи на запрошенном языке; если перевода нет, используется язык по умолчанию, затем любой доступный. Пустые сутки — валидный пустой список. Ответы кешируются сервером и обновляются автоматически при поступлении новых данных EPG.

Встроенный кеш рассчитан примерно на тысячу одновременных клиентов; для более крупных инсталляций и XMLTV-, и JSON-выдачу рекомендуется закрывать кеширующим обратным прокси или CDN.

Воспроизведение архива DVR по событию программы передач (параметр epg в URL воспроизведения) описано в [Воспроизведение архива \(VOD\)](#).

1.8 Веб-интерфейс

Веб-интерфейс — основной способ управления узлом Perfect Streamer и наблюдения за ним. Это раздел контекстной справки: его структура повторяет структуру веб-интерфейса, а каждый экран и важное диалоговое окно имеют свой раздел, на который ведёт кнопка вызова справки.

Раздел описывает назначение каждого экрана и его связь с остальной документацией. Значения и правила заполнения отдельных полей приводятся прямо в интерфейсе (краткие подсказки у полей), а концепции, стоящие за настройками, — в тематических разделах руководства, на которые даны ссылки.

1.8.1 Доступ и роли

Веб-интерфейс открывается в браузере по адресу узла и порту веб-сервера: по умолчанию `http://<адрес>:8808` (или `https://<адрес>:43981` при включённом TLS). Доступны два интерфейса: новый (основной) и классический — по пути /classic. Первичный вход, порты служб и смена пароля по умолчанию описаны в разделе [Начальные настройки](#).

Вход выполняется по логину и паролю (аутентификация Digest). Права определяются ролью учётной записи:

Admin

Полный доступ: настройка потоков и служб, администрирование, просмотр.

RestrictedAdmin

Просмотр и постановка потоков на паузу.

Viewer

Только просмотр состояния — без изменения конфигурации.

Управление учётными записями и ролями — на экране [Веб-сервер и учётные записи](#).

Вход

Форма входа открывается до загрузки приложения и запрашивает логин и пароль. После успешного входа открывается последний просмотренный экран. Смена пароля по умолчанию обязательна при первом входе (см. [Начальные настройки](#)).

1.8.2 Устройство интерфейса

Слева — навигационное меню, сгруппированное по разделам; сверху — панель с общими элементами. Меню работает в двух областях, которые переключаются в верхней панели:

- **Узел** — управление текущим узлом и наблюдение за ним. Три группы меню:
 - [Мониторинг](#) — состояние потоков, клиентов, оборудования и служб;
 - [Настройка](#) — пользователи и EPG;
 - [Администрирование](#) — серверные службы, хранилища, лицензия, обслуживание.
- **Meshwork** — сводные экраны сети узлов: карта, соединения и общий каталог ресурсов (см. [Meshwork](#) и раздел [Meshwork](#)).

Состав меню зависит от роли и от того, что на узле есть. Пункт, на который нет прав, не показывается. «DVB-адаптеры» и «Оборудование» появляются по обнаруженным устройствам, а не по настроенным. Остальные условные пункты зависят не от оборудования, а от самих объектов: «DVR» требует хотя бы одного хранилища, «Транскодеры» — хотя бы одного экземпляра транскодера, «Тестовые потоки» — работающего тестового входа.

Общая панель

В верхней панели, помимо переключателя области, доступны: индикатор времени работы, колокольчик оповещений (только у роли администратора; ведёт на экран [Оповещения](#)), список горячих клавиш, кнопка вызова справки, выбор языка, переключатель открытия соседних узлов, переключатель темы оформления и выход из системы. Переключатель открытия узлов показывает текущее значение — «Новая вкладка» или «Текущая вкладка» (подсказка «Открывать узлы в») — и действует всюду, где интерфейс ведёт на админку другого узла.

1.8.3 Как работает контекстная справка

Справка вызывается из интерфейса и открывает раздел, соответствующий текущему контексту:

- для **экрана** справка открывается общей кнопкой вызова справки и ведёт на раздел этого экрана;
- для **диалогового окна** справка вызывается кнопкой «?» в самом окне и ведёт на раздел этого окна.

Каждый экран и каждое важное окно имеют собственный раздел справки, поэтому переход из интерфейса всегда ведёт на описание именно того, что открыто на экране.

Мониторинг

Группа **Мониторинг** в области «Узел» собирает экраны наблюдения за текущим узлом: состояние потоков, клиентов, оборудования и служб. Экраны обновляются автоматически. Большинство из них только показывает данные, но часть позволяет и управлять объектом: «Потоки» — создавать и настраивать потоки, «DVB-адаптеры» — добавлять, настраивать и сканировать адаптер, «База данных EPG» — править канал. Настройка служб, учётных записей и хранилищ вынесена в группы [Настройка](#) и [Администрирование](#).

Центральный экран группы — «Потоки» — вынесен в отдельный раздел [Потоки](#).

Обзор узла

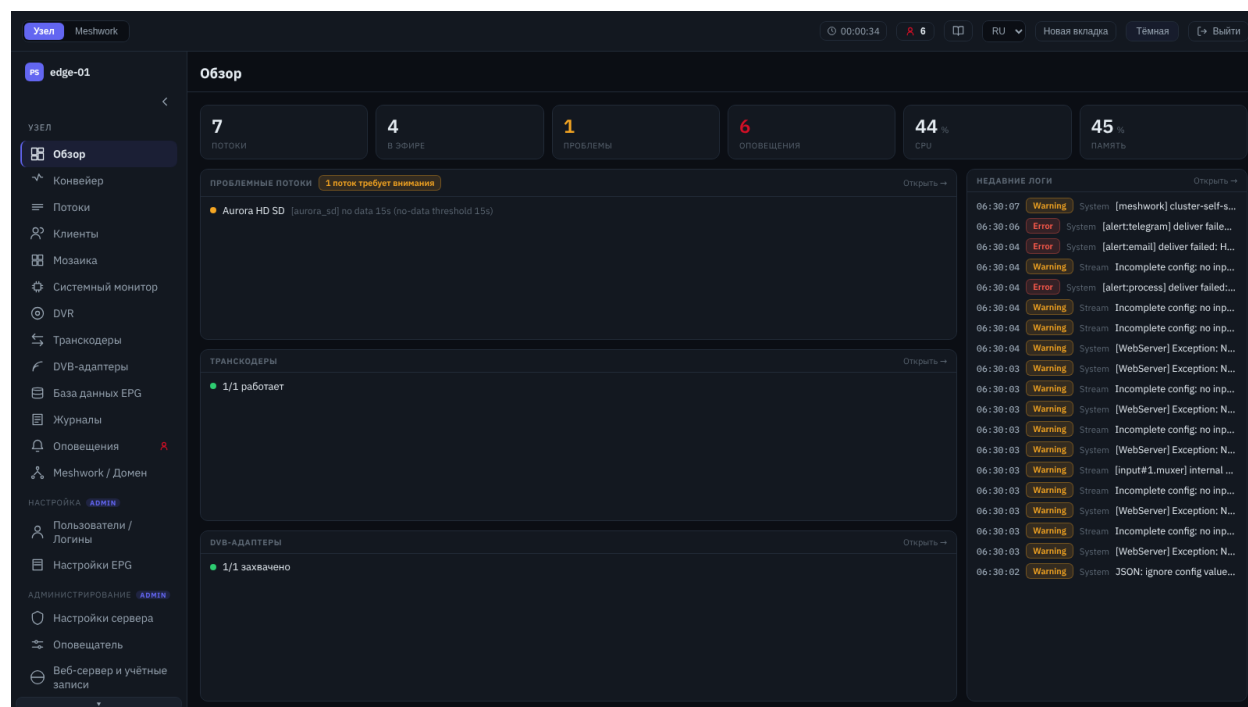


Рис. 5: Обзор узла.

Стартовая панель узла. Верхняя полоса — плитки ключевых показателей: число потоков и из них «В эфире», счётчик «Проблемы», активные «Оповещения» (в роли администратора), загрузка CPU

и памяти. Ниже — панели проблемных подсистем: «Проблемные потоки», «Транскодеры», «DVB-адаптеры» и (администратору) «Недавние логи»; каждая перечисляет неисправные объекты либо краткую сводку «в норме»; из заголовка панели ведёт переход на соответствующий экран. В самом низу — полоса «Избранное» с закреплёнными графиками сигнала DVB-адаптеров и плитками мозаики (закрепляются на своих экранах). Оценка загрузки узла подробнее — на экране [Системный монитор](#).

Конвейер

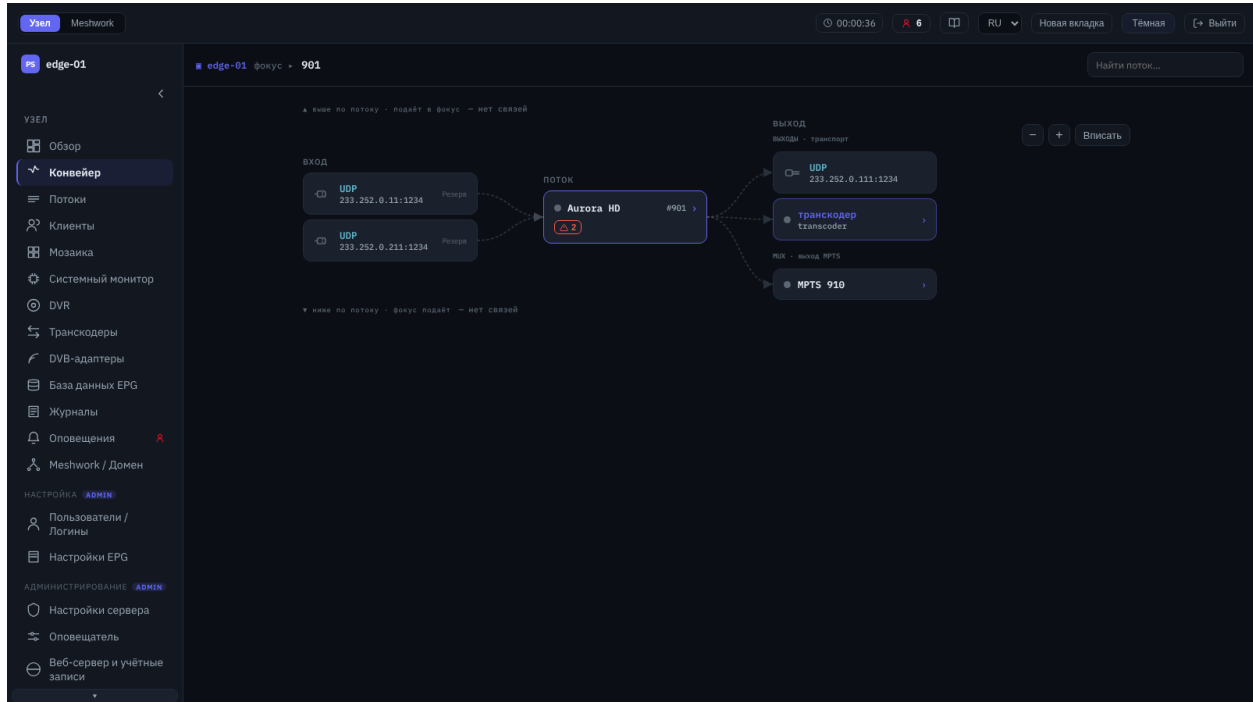


Рис. 6: Граф конвейера: одиночный поток (SPTS).

Граф обработки одной выбранной сущности: как источники, поток и выходы связаны в единый конвейер (ВХОД → ПОТОК → ВЫХОД). Вид фокуса выбирается по типу: поток SPTS или MPTS (с показом входящих и исходящих программ мультимплекса; список потоков — [Потоки](#)), [Адаптивные наборы](#), транскодер ([Транскодеры](#)) или DVB-адаптер ([DVB-адаптеры](#)). Между сущностями переходят стрелками-соседями, поиском по имени или из других экранов; текущий фокус хранится в адресе (можно дать прямую ссылку). Отдельный блок сводит локальные ретрансляции узла (peer), а «двери» на соседние узлы открывают их админку на том же фокусе. У карточек входов и выходов в транспортах общего каталога домена есть кнопка «Показать в библиотеке» ([Библиотека — этот поток](#)). Граф показывает только активные потоки и IO; приостановленные входы и выходы в нём не отображаются. Модель передачи описана в [Модель передачи: Stream, Sender, Receiver, Peer](#), обработка потока на узле — в [Streamer: особенности реализации](#), мультимплексы MPTS — в [MPTS-потоки](#).

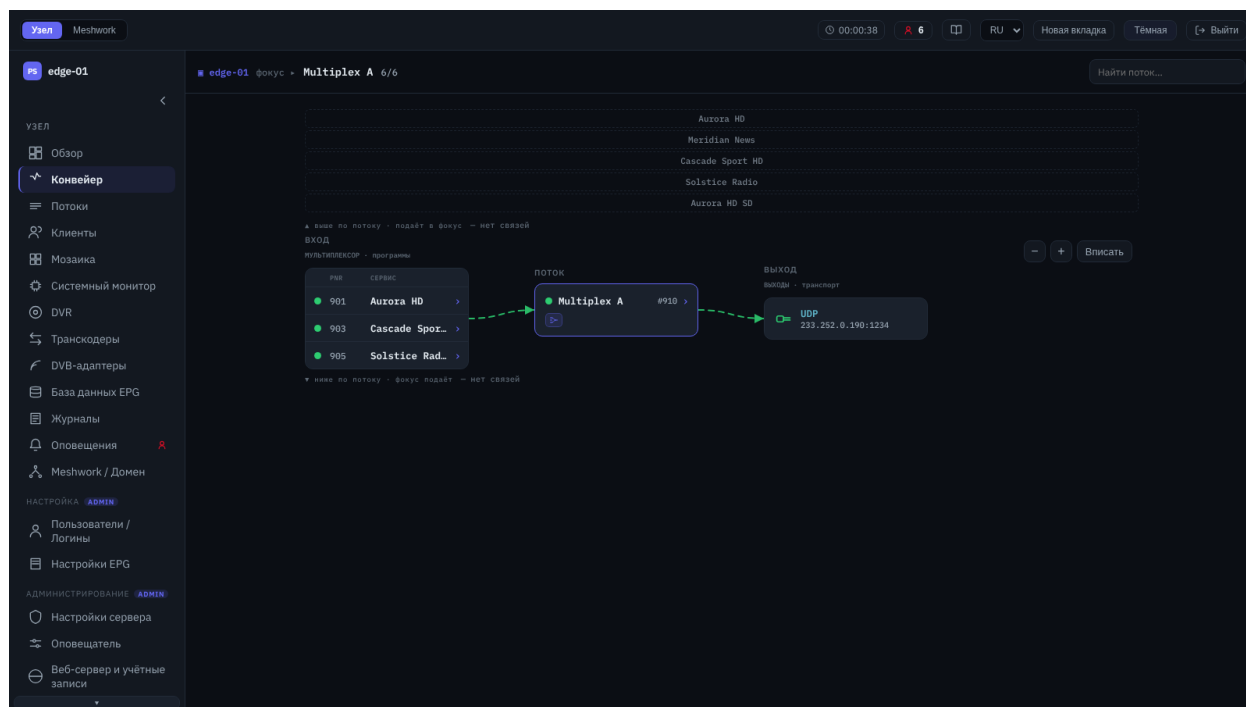


Рис. 7: Граф конвейера: мультиплексор MPTS.

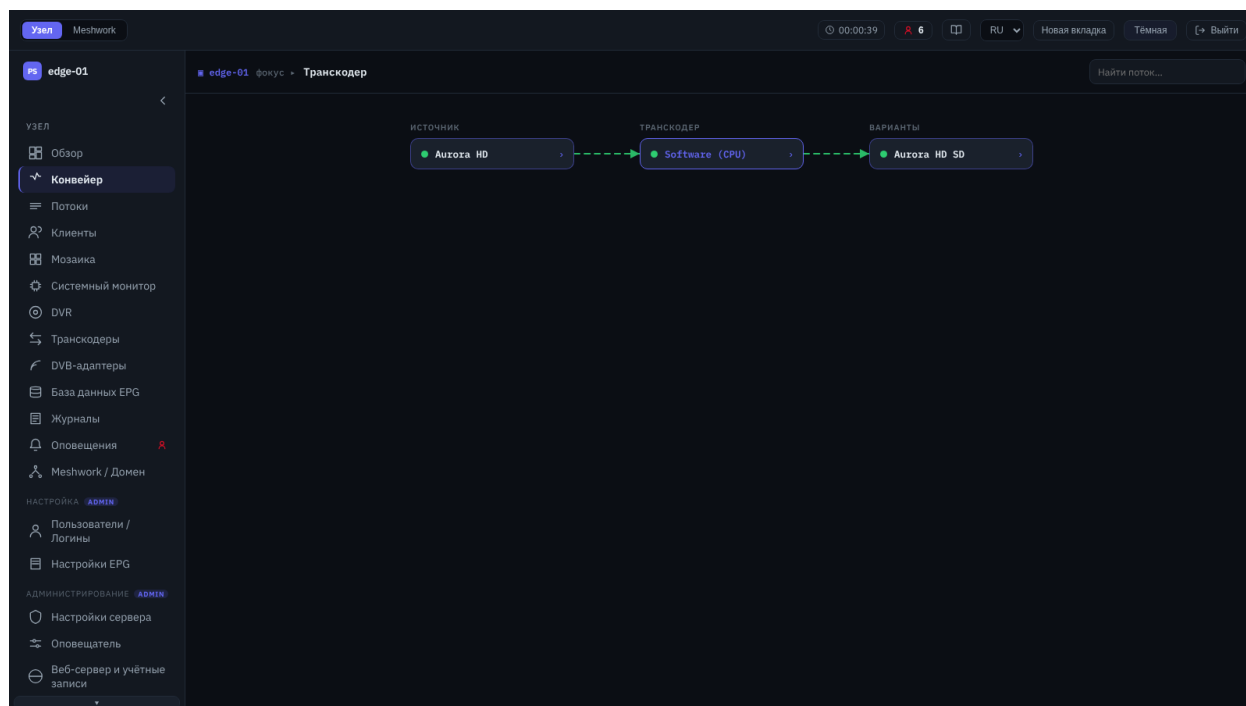


Рис. 8: Граф конвейера: транскодер.

Клиенты

Список подключённых получателей, разложенный по вкладкам протоколов раздачи: «HTTP / OTT», «SRT», «PS1» и «RIST»; у каждой вкладки — живой счётчик активных сессий. Показаны операционно важные колонки: клиент (с пометкой происхождения — «Локальный логин» или «Внешний биллинг»), адрес, поток, логин, для SRT — режим (caller/listener), потери и RTT, для OTT — тип сессии и «Качество» доставки, время подключения. По списку работает поиск (имя, логин, адрес), длинные списки разбиваются на страницы. Разворот строки открывает полный набор диагностических счётчиков сессии, обновляемый в реальном времени. Администратору доступно действие «Отключить» — оно завершает все сессии выбранного логина. Протоколы передачи описаны в [Peer-протоколы надёжной передачи](#), OTT-раздача — в [OTT и DVR](#), учётные записи получателей — на экране [Настройка \(2. Поведение клиентов\)](#).

Мозаика

Стена превью: живые миниатюры кадров — по одной плитке на каждый активный поток SPTS и на каждую программу активного MPTS (программы мультимплекса помечены значком). Сетка подстраивается под ширину экрана, размер плиток переключается элементом «Размер» (1, 1/2, 1/3). Приостановленные и остановленные потоки в мозаике не появляются. Нажатие на миниатюру открывает увеличенный кадр; звёздочкой плитку можно закрепить в «Избранном» на экране [Обзор узла](#).

Исходный кадр

Окно увеличенного просмотра выбранного кадра в исходном разрешении — для детальной визуальной проверки картинка. Внутри окна листаются соседние плитки мозаики.

Системный монитор

Состояние сервера в виде карточек: «CPU» (общая загрузка и доля процесса PSS, число ядер), «Память» (использовано и доля PSS, объём в ГБ), «Сеть» (по каждому интерфейсу — приём/передача и загрузка канала, с вложением VLAN под физический адаптер) и карточки аппаратного ускорения (NVIDIA — GPU/память/декодер/энкодер, Intel VPL). Ниже — графики показателей во времени. Отдельной секцией на узлах с DVB-тюнерами выводятся графики качества сигнала фронтендов ([Контроль сигнала](#)). Помогает оценить запас по нагрузке; рекомендации по настройке ресурсов — в [Оптимизация работы узла](#), внешний сбор метрик — в [Подключение внешних систем мониторинга](#).

Монитор DVR

Наблюдение за записью DVR по хранилищам (только просмотр). Для каждого хранилища — заполнение относительно цели очистки (полоса с меткой цели), объёмы (использовано, свободно, всего, размер архива с разбивкой TS/MP4), признак «Давление на диск», текущая фоновая задача и число привязанных потоков; активные оповещения хранилища выводятся рядом. Разворот показывает построчно потоки архива (идёт запись / простой, глубина хранения, накопленный объём, задержки чтения/записи) и диагностику обслуживания (сбор осиротевших каталогов, обрезка по давлению на диск). Для потока открывается окно [Статистика потока](#); администратору — очистка архива отдельного потока и общая команда «Очистить осиротевшие архивы». Настройка самих хранилищ — на экране [Хранилища DVR](#); запись и воспроизведение архива описаны в [OTT и DVR](#) и [DVR: архив потока](#).

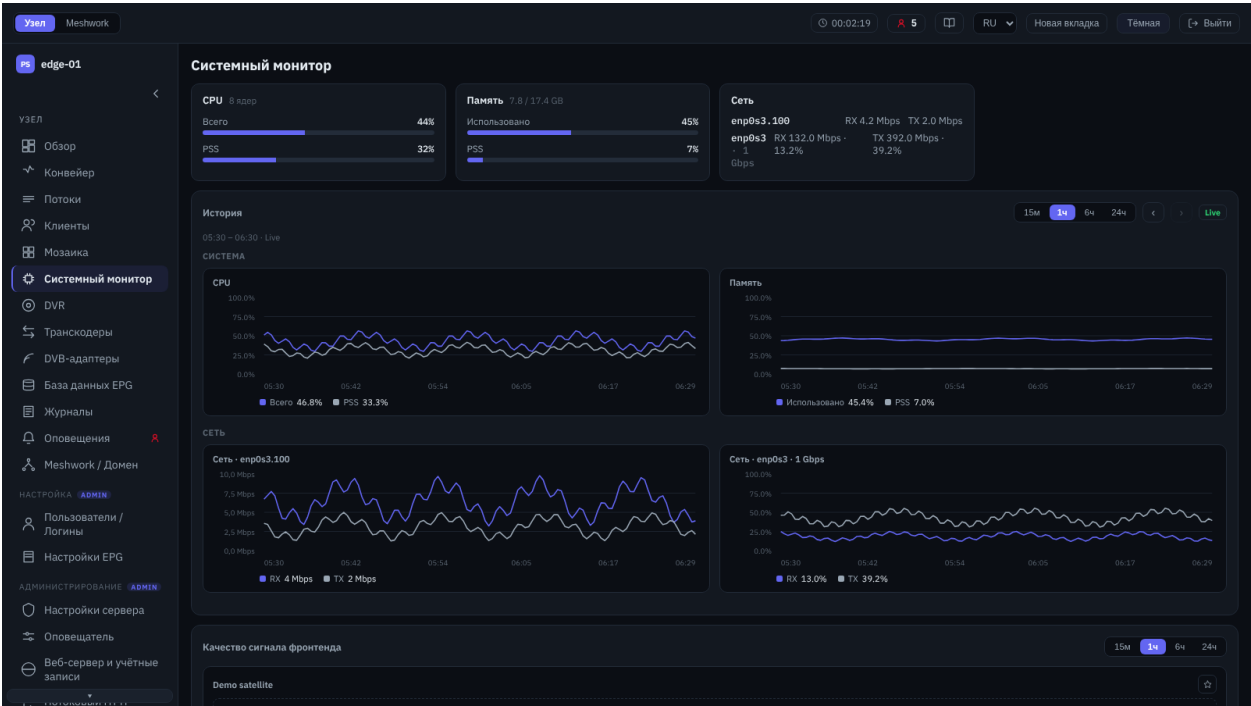


Рис. 9: Системный монитор.

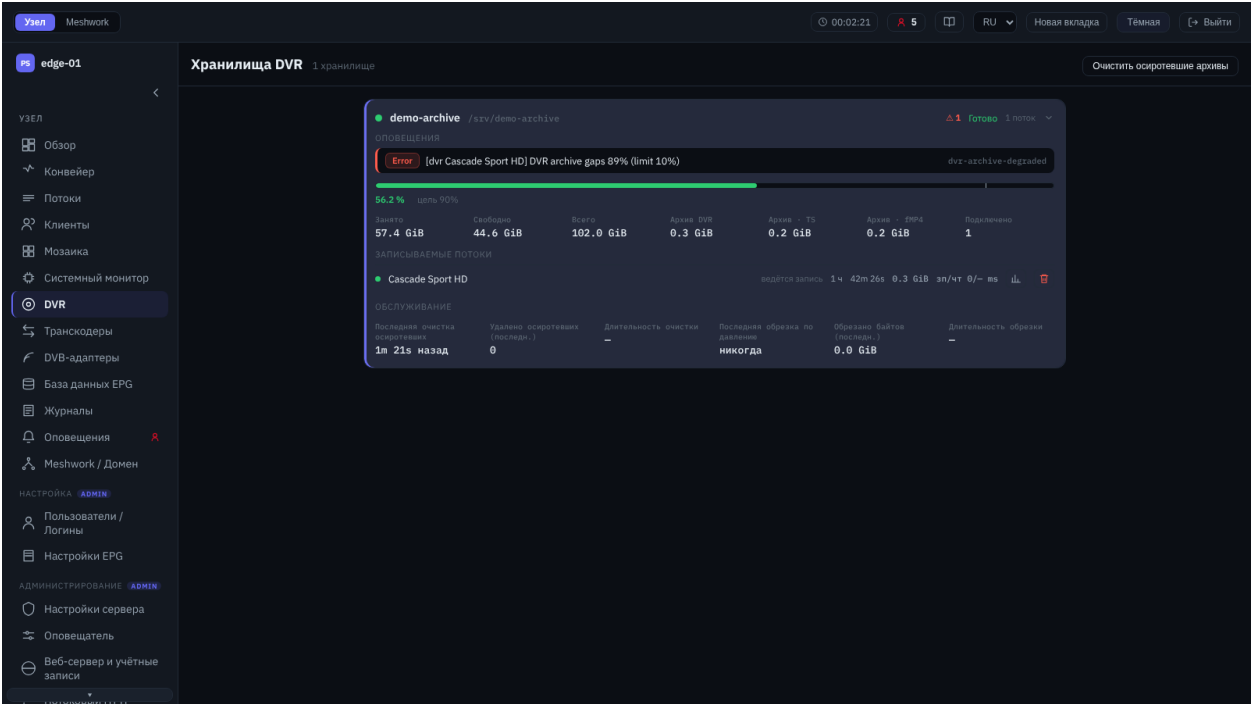


Рис. 10: Монитор хранилищ DVR.

Транскодеры

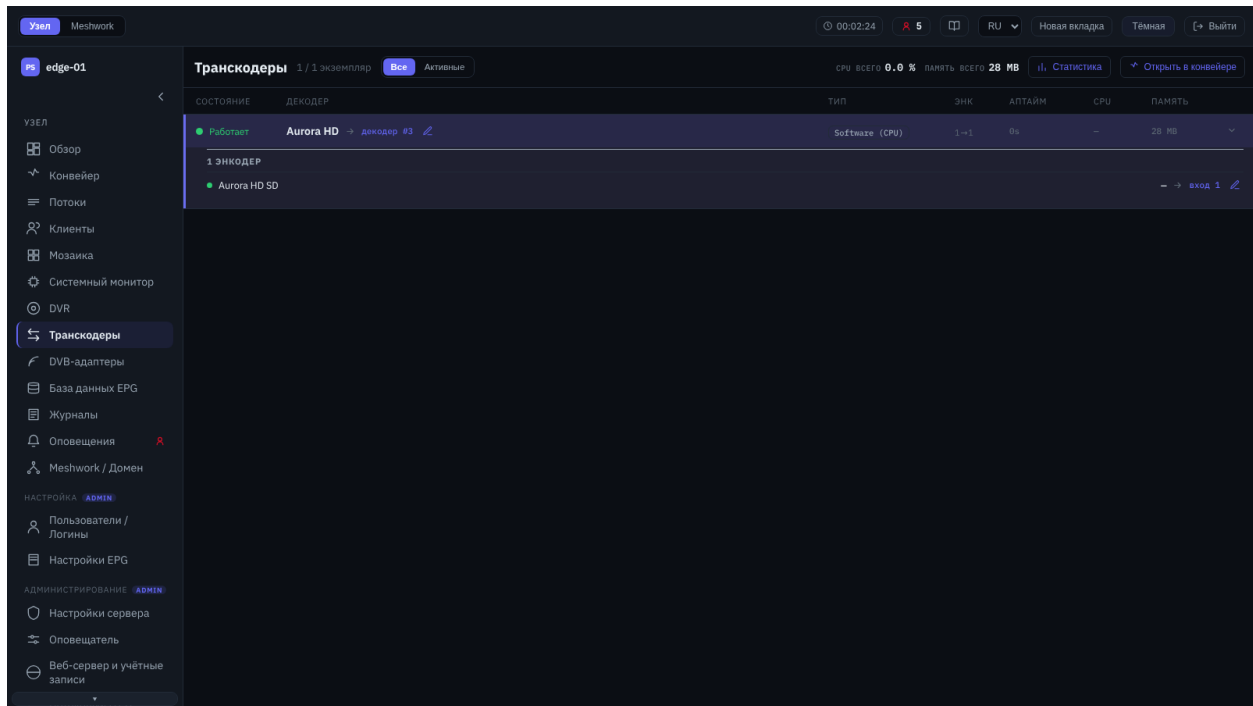


Рис. 11: Экран «Транскодеры».

Список активных экземпляров транскодеров узла с фильтром «Все / Активные». В шапке — суммарная загрузка «CPU всего» и «Память всего», а также действия над выделенным экземпляром: «Статистика» и «Открыть в конвейере» (*Конвейер*). По строке — состояние, декодер (источник), тип, число энкодеров, время работы, CPU и память; разворот показывает энкодеры с параметрами выхода (видеокодек, разрешение, кадровая частота/развёртка, аудиокодеки, реальный битрейт). Метки конвейера открывают статистику соответствующего потока. Администратору у декодера и у каждого энкодера доступна кнопка настройки: она открывает поверх списка редактор соответствующего потока (*Настроить поток*). Возможности и настройка транскодирования — в *Транскодеры*, наблюдение — в *Контроль работы*, установка пакетов и драйверов — в *Транскодеры*.

Экземпляр транскодера

Карточка отдельного транскодера — окно статистики, открываемое из шапки экрана. Показывает источник (декодер), список энкодеров (1 → N), ресурсы (CPU, время работы, PID, память) и журнал процесса: последний код выхода, текущий и предыдущий журналы запусков (с копированием) — для диагностики.

DVB-адаптеры

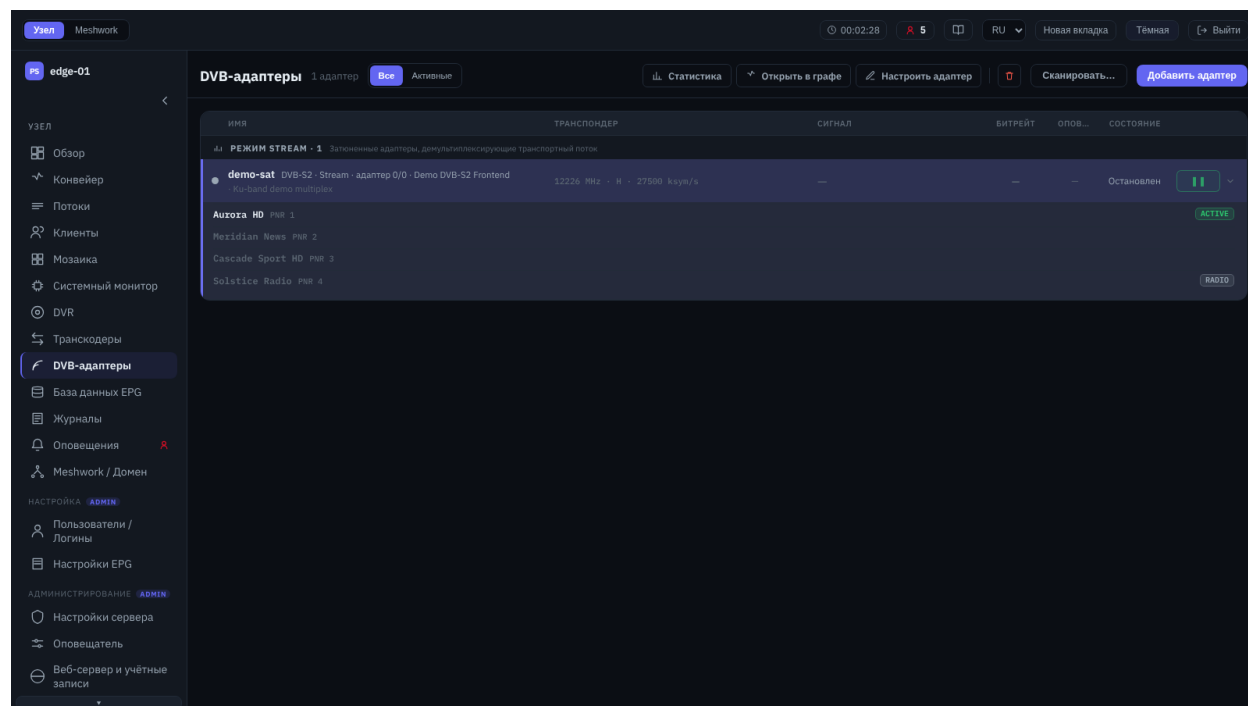


Рис. 12: Экран «DVB-адаптеры».

Единый экран DVB-приёма: конфигурация адаптеров, обогащённая живым состоянием (захват сигнала, качество, битрейт). Строки сгруппированы по режиму — «Режим Stream» (приём в поток) и «FE Monitor» (мониторинг фронтенда, настроенного другим приложением); в колонках — имя, транспондер, сигнал, битрейт, оповещения и состояние; у строки — переключатель паузы. Действия над выделенным адаптером вынесены в шапку: «Статистика», «Открыть в графе» ([Конвейер](#)), а также (администратору) «Настроить адаптер» и «Удалить адаптер». Здесь же — фильтр «Все / Активные», «Добавить адаптер» и «Сканировать...» (администратору). Приём DVB описан в [DVB-приёмник](#), наблюдение за адаптерами — в [Контроль сигнала](#).

Редактор DVB-адаптера

Окно добавления и настройки DVB-адаптера. «Режим» и «Система вещания» задаются при добавлении и далее неизменны — они определяют состав настроек. Оборудование выбирается из списка обнаруженных фронтендов (см. [Оборудование](#)). Для режима Stream доступны параметры настройки (частота, символьная скорость, FEC, модуляция и прочее — по системе вещания), секция LNB для спутникового приёма, «T2-MI», «Дешифрование BISS» и «Дешифрование CI/CAM». Приглушение оповещений адаптера и раздел «Дополнительно» (расширенные параметры) доступны в обоих режимах; режим FE Monitor читает сигнал, не перестраивая тюнер. Приём и дескремблирование описаны в [Адаптеры](#), [Дескремблирование](#) и [Декапсуляция T2-MI](#).

Сканирование DVB

Мастер поиска каналов на свободном тюнере (администратору). Проходит по шагам: выбор тюнера и системы вещания, выбор источника — «Каталог» (спутник, кабель или эфир с параметрами LNB) либо «Слепое сканирование» по диапазону частот; затем идёт перебор транспондеров с индикацией прогресса и накоплением найденных мультиплексов, у каждого — список сервисов. Найденные мультиплексы можно добавлять по одному (открывается окно настройки адаптера с предзаполненными параметрами) или выбрать несколько и добавить разом; имена формируются автоматически и проверяются на совпадения. Поиск каналов описан в [Сканирование](#).

Статистика адаптера

Окно живых параметров выбранного адаптера. Содержит: сводку состояния и параметры настройки, активные оповещения, графики истории сигнала (уровень, SNR, BER, UCB), секцию «Метрики» (фронтенд, статус захвата, частота, SNR, уровень сигнала, счётчики ошибок, для режима Stream — битрейт), список программ, «Несущие T2-MI», «Дешифраторы BISS», «Модуль CI/CAM», «Профилировщик» (загрузка потока обработки, переполнение DVR-кольца) и журнал адаптера. Администратору доступно «Сбросить статистику». Служит для настройки антенны и контроля стабильности приёма ([Контроль сигнала](#)).

Тестовые потоки

Экран тестовых потоков — генерация служебных потоков для проверки трактов ([Тестовые потоки](#)). Раздел находится в подготовке.

База данных EPG

Собранная программа передач и состояние её источников. Экран содержит вкладки:

База данных

Просмотр накопленной программы: выбор источника, список каналов с поиском и фильтром по наборам, расписание выбранного канала по дням (переключение дня и языка, подсветка текущей передачи). Администратору доступна правка канала — имя, часовой пояс, значок, принадлежность к наборам.

Состояние источников

Статус импорта по каждому источнику: состояние, время прошлого и следующего обновления, число каналов и событий, ошибка; администратору — принудительное обновление источника.

Импорт и обработка EPG описаны в [База данных EPG](#) и [Импорт EPG/XMLTV](#); источники и настройка — в [Настройка источников](#) и на экране [Настройки EPG](#).

Журналы

Узел Meshwork 00:02:34 4 RU Новая вкладка Темная Выйти

PS edge-01

УЗЕЛ

- Обзор
- Конвейер
- Потоки
- Клиенты
- Мозаика
- Системный монитор
- DVR
- Транскодеры
- DVB-адаптеры
- База данных EPG
- Журналы**
- Оповещения
- Meshwork / Домен

НАСТРОЙКА ADMIN

Пользователи / Логины

Журналы

Важность $\geq$ Все Тип источника Все ID источника Поиск mm/dd/yyyy

По mm/dd/yyyy

Применить Сначала новые Обновить

ВРЕМЯ	ВАЖНОСТЬ	ИСТОЧНИК	СООБЩЕНИЕ
27.07.2026, 10:19:11	Error	System	[alert:telegram] deliver failed: Runtime exception: telegram send failed (-1000000000000: Runtime exception: HTTP 401 {"ok":false,"error_code":401,"description":"Unauthorized: invalid token specified"}) (x15)
27.07.2026, 10:19:09	Error	System	[alert:email] deliver failed: Host not found: smtp.example.net (x15)
27.07.2026, 10:19:09	Info	Stream #901 · Aurora HD	[input#1.udp] Change state Try run -> Running
27.07.2026, 10:19:08	Info	Stream #901 · Aurora HD	[input#1.udp] Change state Stop -> Try run
27.07.2026, 10:19:08	Warning	Stream #901 · Aurora HD	Fallback: restore, change input to #1
27.07.2026, 10:19:08	Info	Stream #901 · Aurora HD	[input#2.udp] Change state Running -> Stop
27.07.2026, 10:19:07	Info	Stream #901 · Aurora HD	Fallback: check #1 and try restore
27.07.2026, 10:19:07	Error	System	[alert:process] deliver failed: File access error: process-command not executable: /usr/local/bin/demo-alert-hook (x14)
27.07.2026, 10:18:43	Warning	System	[cascade_sport.ott/segmenter] MP4/CMAF audio frames not de-framed (LATM?); building video-only #MP4

1-200 из 406 Назад Вперед

Рис. 13: Журналы.

Структурированный журнал событий узла (доступен администратору) с фильтрами и постраничным выводом. Отбор — по важности («Важность $\geq$ »), типу и идентификатору источника, тексту сообщения и диапазону дат; переключается порядок сортировки (сначала новые / старые). В колонках — время, важность, источник и сообщение; источник-поток открывает его окно [Статистика потока](#), прочие источники ведут на профильный экран. Служит для диагностики и разбора инцидентов.

Источник записи обозначается типом объекта, а для потоков — ещё номером и именем: Stream #43 · orig CBC HD; если отображаемое имя не задано, подставляется имя потока. Тем же именем с номером подписаны потоки в списке выбора фильтра «ID источника». Записи, сделанные входом или выходом потока, числятся за самим потоком и показывают его имя, а конкретные вход или выход названы в начале текста сообщения.

Оповещения

Сводный экран действующих оповещений: свои (текущего узла) либо всего домена — переключается областью в шапке. Экран показывает только активный набор — снятое оповещение из списка исчезает. В колонках — важность, источник, сообщение и возраст; источник-поток открывает его окно [Статистика потока](#). В домене из нескольких узлов у строки показывается узел-источник (с переходом в его админку). Оповещения уровня действия администратора снабжены кнопкой «Сбросить» (подтверждение вручную), действующей на своём узле. Модель оповещений, их разбор и полный список кодов приведены в [Оповещения \(алерты\)](#) и [Каталог кодов алертов](#); настройка порогов и внешней доставки — на экране [Оповещатель](#).

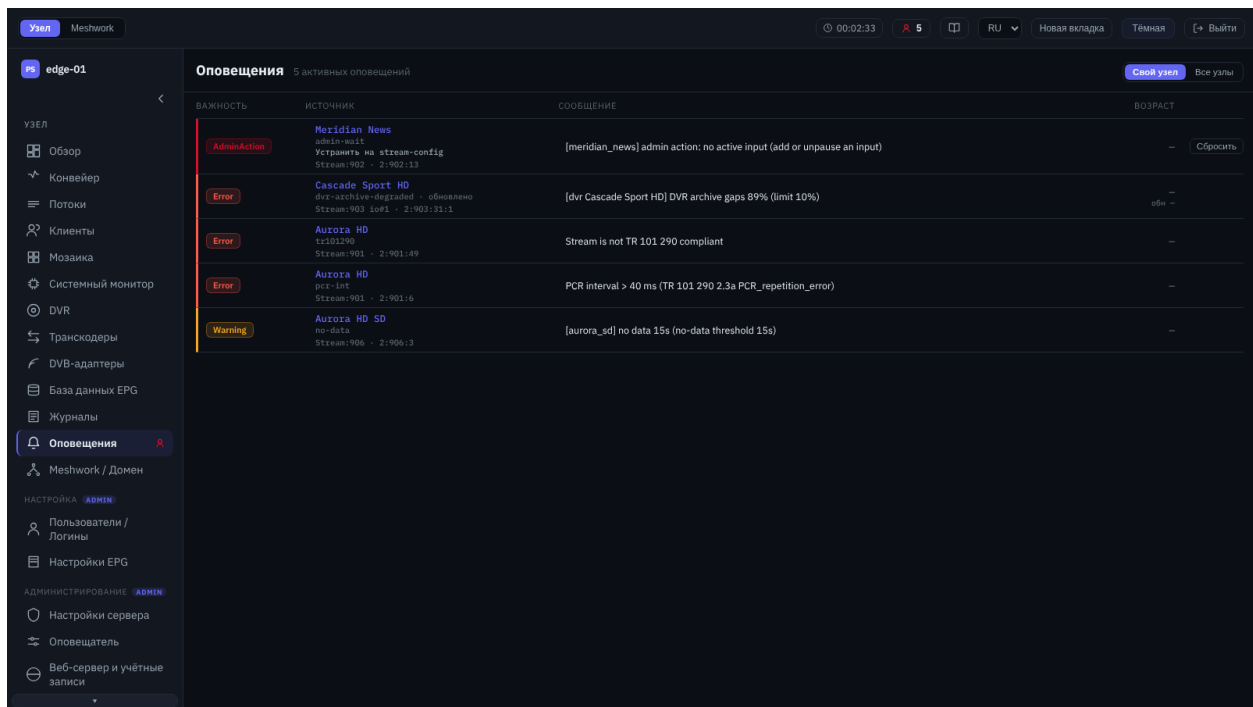


Рис. 14: Активные оповещения.

Meshwork / Домен

Взгляд на сеть Meshwork с точки зрения текущего узла. В шапке — имя узла, его примечание, наблюдаемый внешний адрес («виден как ...») и связанные домены; когда доменов нет, выводится «Нет связанных доменов».

Тело экрана — таблица пиров: узлы домена, видимые с этого узла, со столбцами «Тип» («Настроенный» или «Репликация»), «Статус» («Активен» / «Недоступен»), «Узел», «Домен», «Адрес», «NAT» и «Последний контакт». Сначала перечислены настроенные соседи, затем найденные автоматически, внутри — по имени узла. Строка текущего узла помечена «этот узел», супер-узел отмечен отдельно, сбои keep-alive — меткой «Деградация»; узел за NAT показывает «Внутренняя сеть» вместо недостижимого частного адреса.

Стрелка в конце строки разворачивает подробности пира: публичные точки входа (HTTP и HTTPS), точное время последнего контакта, телеметрию контакта (задержка, число неудач подряд и всего, последняя ошибка), заявленный и наблюдаемый адреса и поспе экземпляра; отсюда кнопкой открывается админка соседнего узла. Сетевой взгляд на пиры домена — [Пирь узла](#); общий каталог ресурсов домена показывает отдельный экран [Библиотека](#).

Понятия и сводные экраны сети описаны в разделе [Meshwork](#) (карта — [Карта сети и каталог ресурсов](#), каталог — [Общий каталог ресурсов](#), пиры — [Соседи и секреты](#)); экраны области «Meshwork» — в [Meshwork](#).

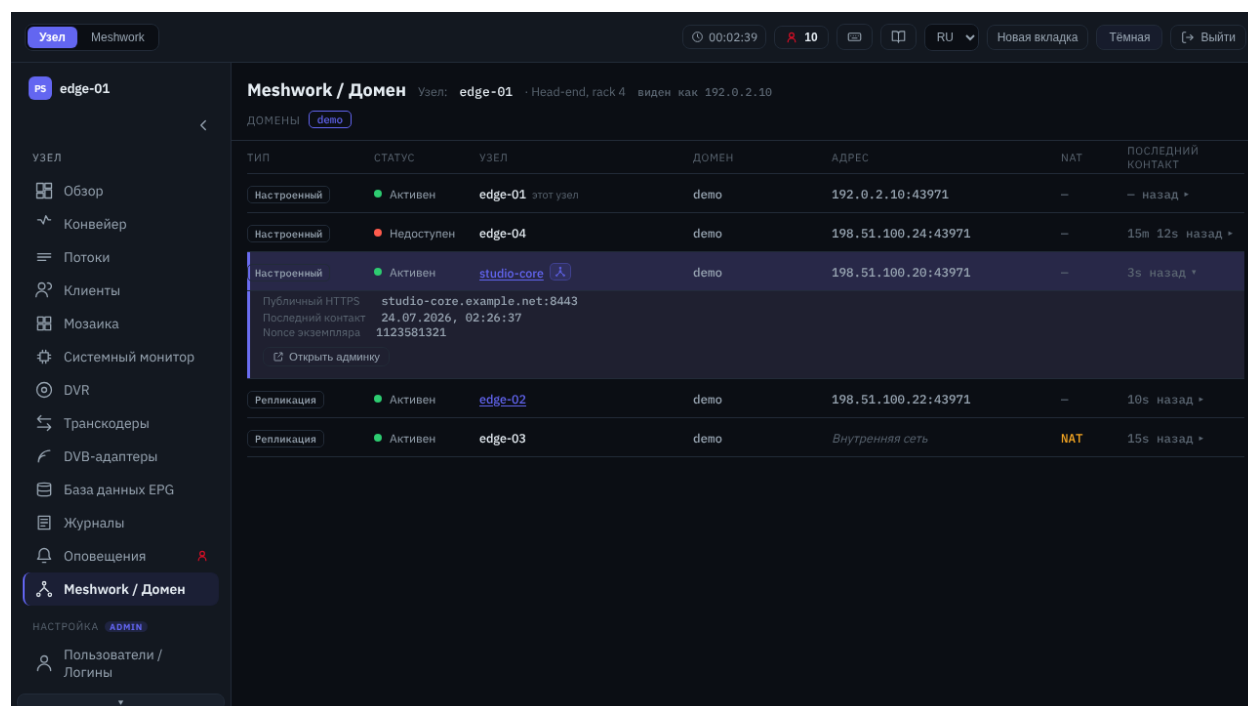


Рис. 15: Домен Meshwork: пиры узла.

Потоки

Экран **Потоки** — центральное рабочее место узла: единый список настроенных потоков с их состоянием, активным входом, битрейтом, «Аптайм» и счётчиком ошибок непрерывности «СС». Каждая строка раскрывается по клику и показывает входы и выходы потока с их состоянием, приоритетом резервирования, заметками и активными оповещениями; отдельный вход или выход отсюда можно приостановить и возобновить, а кнопка «Показать в библиотеке» рядом с адресом открывает список остальных точек домена на этом адресе (*Библиотека — этот поток*). Понятие потока и модель передачи описаны в *Модель передачи: Stream, Sender, Receiver, Peer*; обработка SPTS — в *SPTS-потоки*, мультиплекс MPTS — в *MPTS-потоки*.

Список сгруппирован по типам: одиночные потоки (SPTS) идут общим списком, ниже — группы «MPTS-потоки» и «Адаптивные потоки» (ABR-наборы). Над списком расположены средства навигации и управления:

Поиск и фильтры

Поле «Найти поток...» ищет по имени, заметке и адресу входа/выхода и подсвечивает найденную строку, не сокращая список. Фильтр «Теги» ограничивает список по меткам потоков, переключатель «SPTS / MPTS / Адаптивный» показывает или скрывает группы типов, а сегмент «Все / Активные» скрывает приостановленные потоки.

Действия над выбранным потоком

Заголовок содержит действия, применяемые к раскрытому (выбранному) потоку или ABR-набору: «Статистика потока» (*Статистика потока*), «Открыть в конвейере» (*Конвейер*), «Настроить поток» (*Настроить поток*) и «Удалить поток». Кнопка «+ Новый поток» открывает окно создания. Администратору доступно перетаскивание строк для изменения их порядка (пока список не отфильтрован).

Состояние и оповещения

Цветовой индикатор состояния, плашки признаков потока (CBR/VBR, скремблирование,

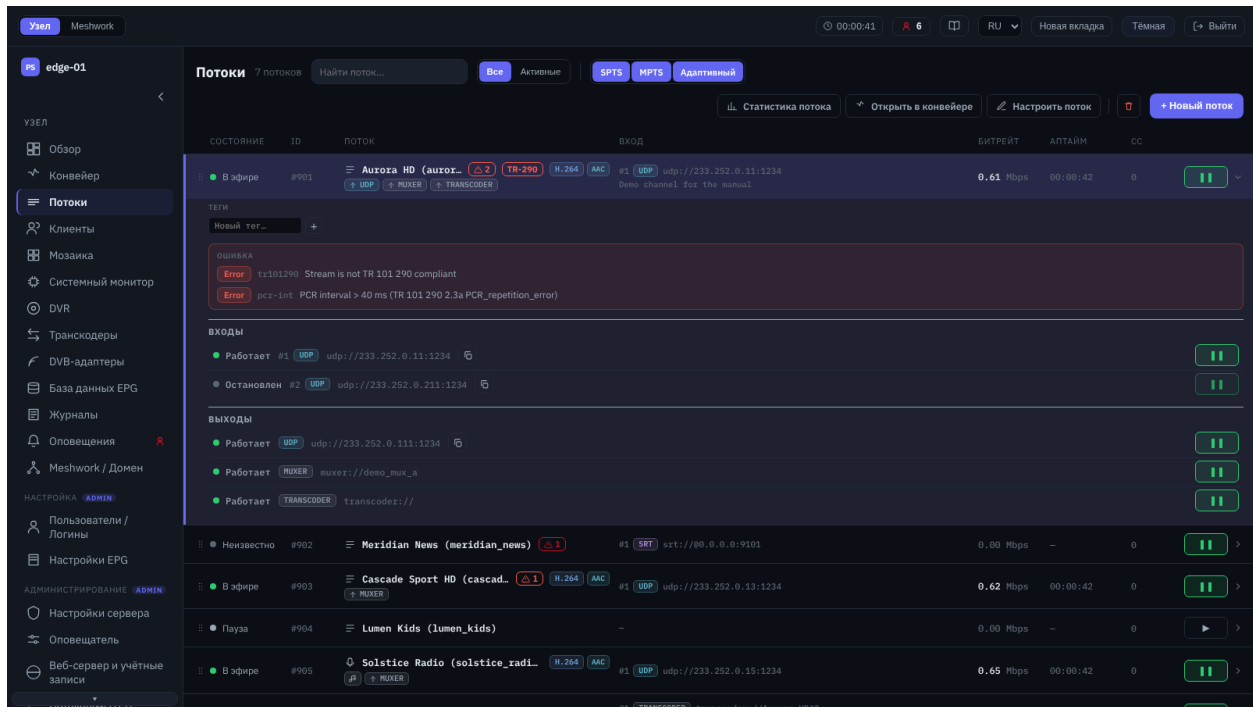


Рис. 16: Экран «Потоки»: список потоков с раскрытой строкой.

защита DRM, работа на резервном входе, только аудио или только видео, кодеки, TRACE) и отметка активных оповещений. Оповещения привязаны к потоку и его входам/выходам (**Оповещения (алертер)**); отчёт TR 101 290 и раздел глубокого анализа открываются прямо из строки.

Отдельные окна экрана описаны ниже.

Новый поток

Окно создания потока. Тип выбирается переключателем: одиночная программа (**SPTS**), мультимплекс (**MPTS**) или адаптивный набор (**Адаптивный**); для SPTS дополнительно задаётся «Тип контента». После создания SPTS- или MPTS-потока сразу открывается редактор (**Настроить поток**). Для адаптивного набора отдельный редактор не открывается — состав участников и их битрейты задаются прямо в этом окне, поскольку набор строится из уже настроенных ОТТ-потоков. Различия SPTS и MPTS описаны в *SPTS-потоки* и *MPTS-потоки*, адаптивные наборы — в *Адаптивные наборы*.

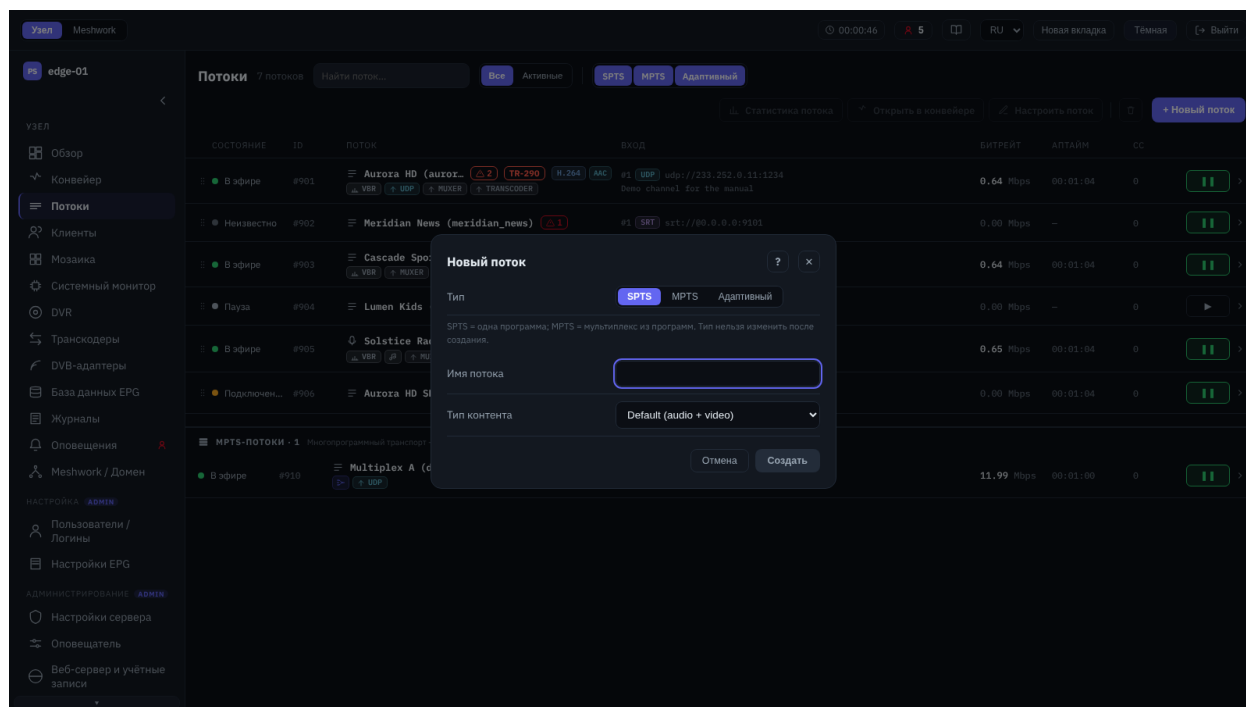


Рис. 17: Окно «Новый поток».

Настроить поток

Основной редактор потока. Настройки сгруппированы по вкладкам; для потоков MPTS вкладка «ОТТ» не показывается.

Поток

Идентификация (имя, отображаемое имя, для SPTS — тип потока, заметка) и поведение потока: таймаут, интервал проверки, повторная проверка основного входа, генерация мозаики ([Мозаика](#)), извлечение EIT в базу данных EPG ([База данных EPG](#)) и признак скремблирования для SPTS. Для SPTS здесь же настраивается генерация EIT из выбранного источника EPG и его канала ([Генератор EIT](#)).

Входы

Список источников потока с их приоритетом и резервированием ([Резервирование источников и раздача](#)). Добавление и изменение входа открывают редактор входа и выхода ([Редактор входа и выхода](#)); отдельный вход можно приостановить.

Выходы

Раздача потока по протоколам передачи ([Рее-протоколы надёжной передачи](#)) и стандартным транспортом ([Другие входы и выходы](#)). Каждый выход добавляется и правится через тот же редактор.

MPEG-TS

Режим битрейта выходного транспортного потока ([Управление битрейтом](#)) и — для SPTS — «ID сети», фиксация интервала PAT/PMT, фильтрация служебных таблиц (CAT/ECM/EMM, EIT, NIT, SDT, субтитры, TDT, телетекст; [Фильтрация MPEG-TS](#)), перезапись SDT и номера программы ([Модификация потока](#)), а также автоматическая раскладка PID программы и базовый PID её окна ([Автоматическая раскладка PID](#)).

ОТТ

ОТТ-раздача потока: режимы сервисов HLS и HTTP ([Режимы раздачи](#)), имя потока в URL,

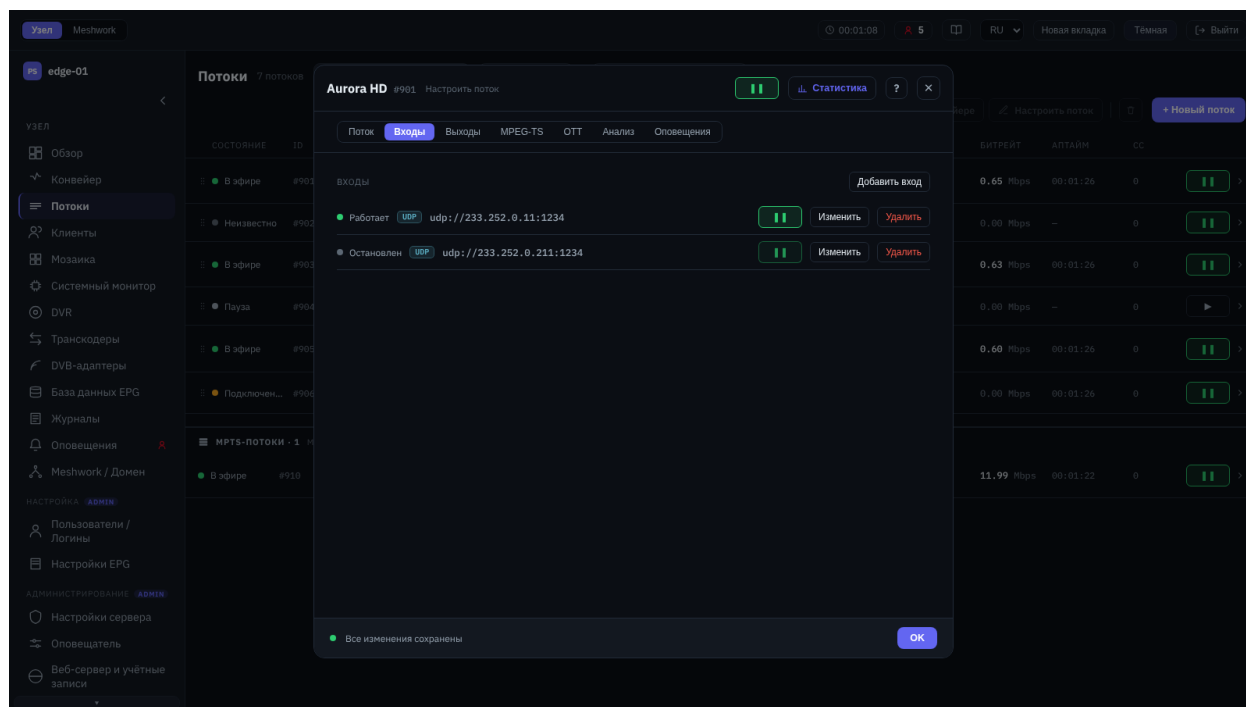


Рис. 18: Редактор потока, вкладка «Входы».

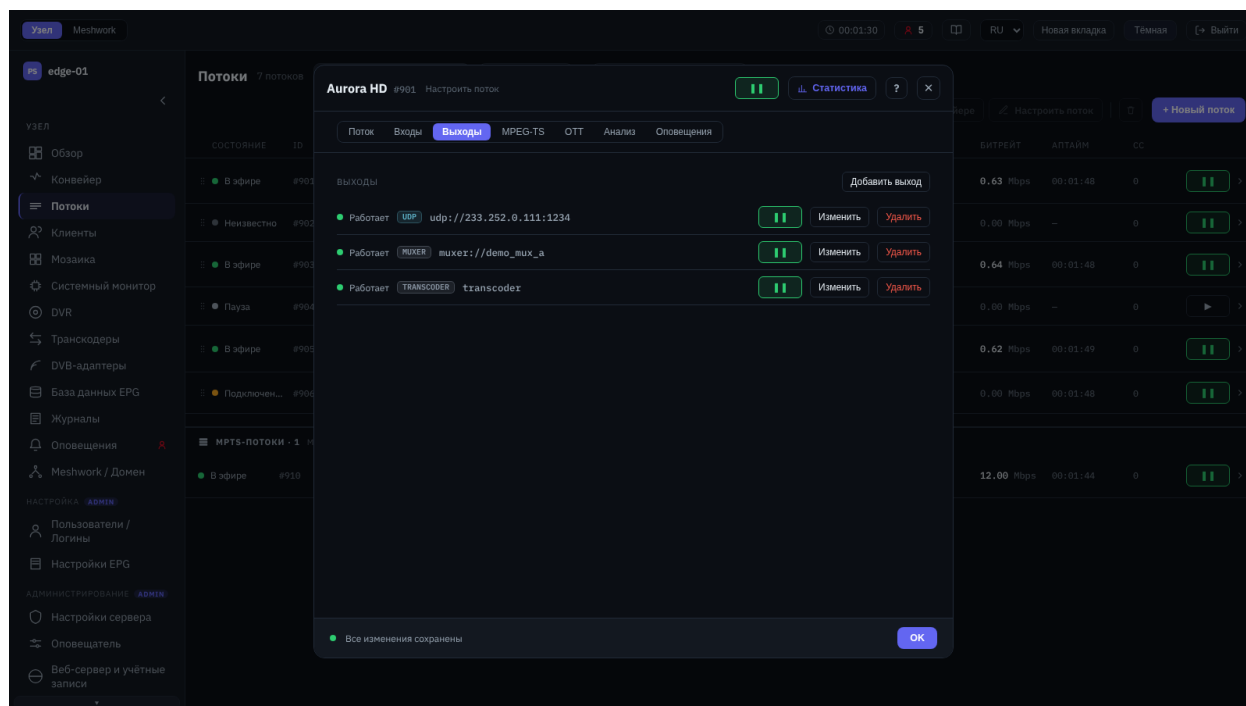


Рис. 19: Редактор потока, вкладка «Выходы».

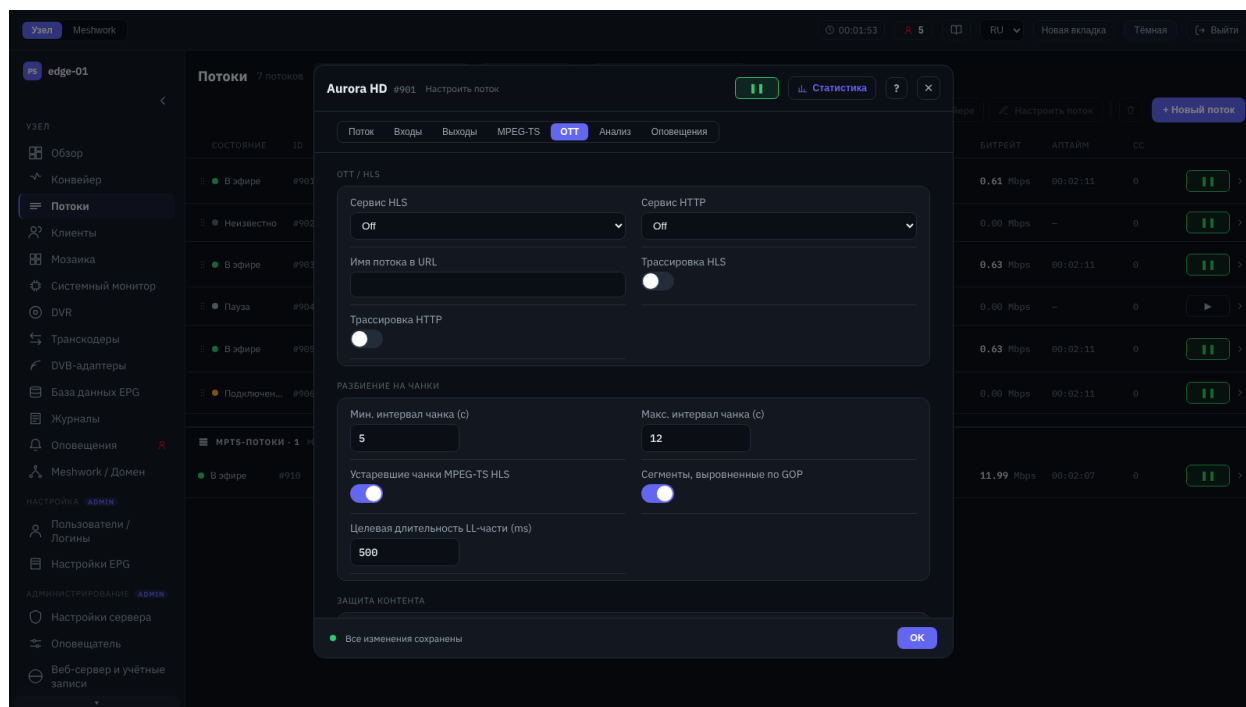


Рис. 20: Редактор потока, вкладка «OTT».

извлечение телетекстовых субтитров в WebVTT, разбиение на чанки и низкая задержка ([Сегментер](#)), запись DVR в выбранное хранилище ([OTT и DVR](#)) и защита контента DRM – HLS AES-128 или CENC ([Защита контента \(DRM\)](#)). Только для SPTS.

Анализ

Контроль качества транспортного потока: подробная трассировка, глубокий анализ элементарных потоков, монитор TR 101 290, анализ разрывов PCR/PTS и видеобuffers T-STD, компенсация дрейфа синхронизации ([Анализатор](#)).

Оповещения

Общий выключатель оповещений потока и задержка подъёма оповещений об ошибках его входов и выходов ([Оповещения \(алертер\)](#)); пороги задаются на уровне узла ([Оповещатель](#)).

Сохранение изменений, затрагивающих записанный архив DVR, дополнительно защищено подтверждением ([Подтвердите изменения DVR](#)). Отдельные окна редактора описаны ниже.

Редактор входа и выхода

Окно добавления и изменения входа или выхода потока. Тип транспорта выбирается при создании и далее неизменен (для смены транспорта вход или выход пересоздаются). Параметры разнесены по вкладкам:

Свойства

Адрес, порт, привязка к сетевому интерфейсу и параметры соединения выбранного транспорта. Реег-протоколы и стандартные транспорты описаны в [Планирование и протоколы передачи данных](#); шифрование – в [Шифрование потоков](#). Для входов PS1 и SRT доступен автоматический вход по реег-авторизации Meshwork с возможностью переключиться на ручные логин и пароль. Адрес можно вставить из буфера обмена («Вставить URL») или выбрать источник из каталога домена («Выбрать из библиотеки...», [Источники библиотеки](#)).

Для входа-транскодера здесь же задаются декодер и параметры кодирования, включая режим изменения размера кадра ([Транскодеры](#)).

MPEG-TS PID

Отбор и переназначение PID на входе или выходе: приём PID (белый список), отклонение PID (чёрный список), переназначение PID и языка аудиодорожек. Пока у потока включена автоматическая раскладка PID ([Автоматическая раскладка PID](#)), группа переназначения PID скрыта на входах и не действует; списки приёма и отклонения работают по-прежнему. На тихог-выходе переназначение PID сохраняется — оно относится к мультиплексу ([Назначение PID](#)), но исходными PID в его парах становятся PID, назначенные автоматической раскладкой.

BISS

Ключи дескремблирования BISS для закодированного источника — один ключ для SPTS либо по ключу на программу (PNR) для MPTS ([Дескремблирование BISS](#)).

Источники библиотеки

Выбор входа из общего каталога ресурсов домена Meshwork: список уже отдаваемых на узлах домена потоков того же транспорта, из которого можно подключить источник, не вводя адрес вручную. Список фильтруется по узлу-источнику и тексту и разбит на страницы; источники того же домена в реег-протоколах отмечаются признаком «пир» ([Общий каталог ресурсов](#)). Не путать с окном [Библиотека — этот поток](#): здесь по каталогу выбирают источник, там смотрят, кто ещё работает с уже заданным адресом.

Библиотека — этот поток

Окно отвечает на вопрос «кто ещё работает с этим адресом». Его открывает кнопка «Показать в библиотеке» рядом с адресом входа или выхода: в раскрытой строке списка потоков, в списках входов и выходов редактора потока, на карточках входов и выходов конвейера ([Конвейер](#)) и в окне статистики — у активного входа и в разделе «Выходы». Подзаголовок окна — адрес той точки, от которой его вызвали.

Кнопка есть только у транспортов, попадающих в общий каталог домена: UDP, RTP, Pro-MPEG, RIST, PS1 и SRT ([Общий каталог ресурсов](#)). У входа-транскодера, муксера, демуксера, файла и тестового генератора записей в каталоге нет, поэтому нет и кнопки — это не признак неисправности.

В окне перечислены остальные точки домена, работающие с тем же адресом. Собственные входы и выходы этого потока в список не попадают, включая его резервные входы: оператор и так их видит. А вот другой поток того же узла, слушающий тот же адрес, — обычный сосед и показывается наравне с остальными.

Столбец «Направление» показывает роль точки: output — она отдаёт этот поток (источник), input — принимает его (потребитель); источники перечислены первыми. Остальные столбцы те же, что в общем каталоге: «Протокол», «Узел», «Поток» и «Назначение» (адрес и порт). За адресом выводится пометка «проверено»: эту пару узел подтвердил доменной авторизацией, а не вывел из совпадения адресов — при работе через NAT адреса сторон и не совпадают. Пометку получает не больше одной строки на слушающий сокет: узел называет одного представителя из подтверждённых сессий, и он меняется по мере подключения и отключения клиентов. Отсутствие пометки поэтому не означает, что связь ненастоящая.

Клик по строке открывает админку её узла на статистике её потока. Строка своего узла открывается на месте, строка соседа — по настройке «Открывать узлы в» общей панели ([Общая панель](#)); Ctrl

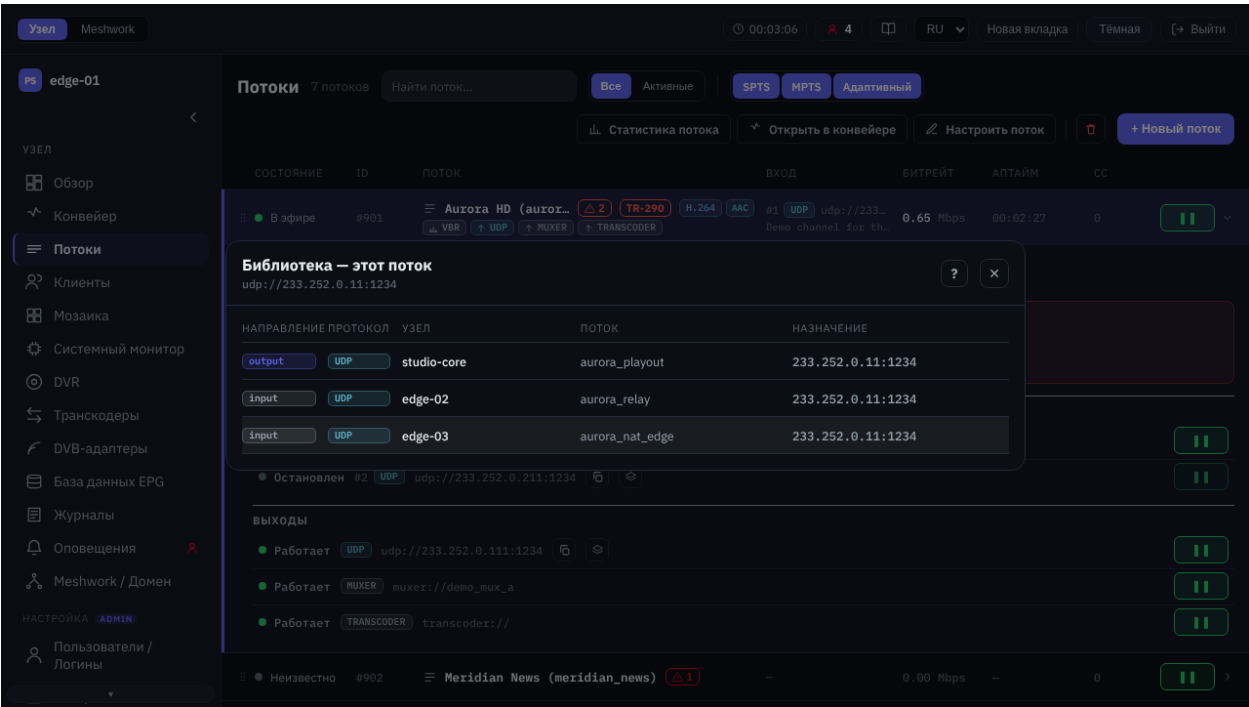


Рис. 21: Окно «Библиотека — этот поток»: у входа потока показаны источник (output) и два других потребителя (input) того же адреса.

или  принудительно открывает новую вкладку. Строка узла, недоступного или неизвестного сети, действия не имеет и несёт подсказку «Узел недоступен».

Совпадение считается по протоколу и полному адресу — группе и порту, а для UDP, RTP, Pro-MPEG и RIST ещё и по VLAN и источнику SSM. Поэтому точка на той же группе, но в другом VLAN или с другим SSM-источником, в список не попадёт, как и точка другого транспорта на том же адресе. Для PS1 и SRT совпадения адресов не требуется вовсе: вызывающая сторона называет партнёра по имени хоста, и это имя сопоставляется с составом домена — поэтому у найденной строки адрес может выглядеть совершенно иначе. Тем же способом находится и подтверждённая пара, по именам узла и потока, которые называет слушающая сторона.

Два сообщения об отсутствии соседей означают разное:

Сообщение	Что означает
«На этом адресе больше нет зарегистрированных точек.»	Точка объявлена в каталоге, соседей у неё нет.
«Эта точка не объявлена в библиотеке (объявляются только работающие сетевые входы и выходы).»	Записи о самой точке в каталоге нет: поток не работает, вход или выход приостановлен, только что запущен либо исключён из публикации полем Peer private (Общий каталог ресурсов). Соседей в этом случае не «нет», они неизвестны.

Тот же каталог целиком показывает экран [Библиотека](#).

Программы

Редактор программ мультиплекса (MPTS), открываемый из окна настройки входа-муксера: состав программ выходного потока, их порядок в PSI и назначение PID. Программу добавляют из числа настроенных SPTS-потокaов либо удаляют; порядок меняется перетаскиванием; отдельную программу можно приостановить. Когда автоматическое назначение PID выключено, для каждой программы правятся PMT PID и PID элементарных потоков. Сборка мультиплекса описана в [Мультиплексор](#), распределение программ и PID — в [Параметры программ](#) и [Назначение PID](#), транзит и анализ MPTS-потокaов — в [MPTS-потокaи](#).

Подтвердите изменения DVR

Предупреждение перед сохранением изменений потокa, которые затрагивают уже записанный архив DVR: отключение записи, переключение на другое хранилище, сокращение срока хранения или защищённого минимума, а также смена ключа DRM у потокa с записью (ранее записанные зашифрованные сегменты при этом становятся недоступны). Окно перечисляет, что именно будет затронуто, и требует подтверждения; безопасные изменения (первое подключение хранилища, увеличение срока хранения) сохраняются без запроса. Запись и хранение архива описаны в [OTT и DVR](#) и [DVR: архив потокa](#).

Статистика потокa

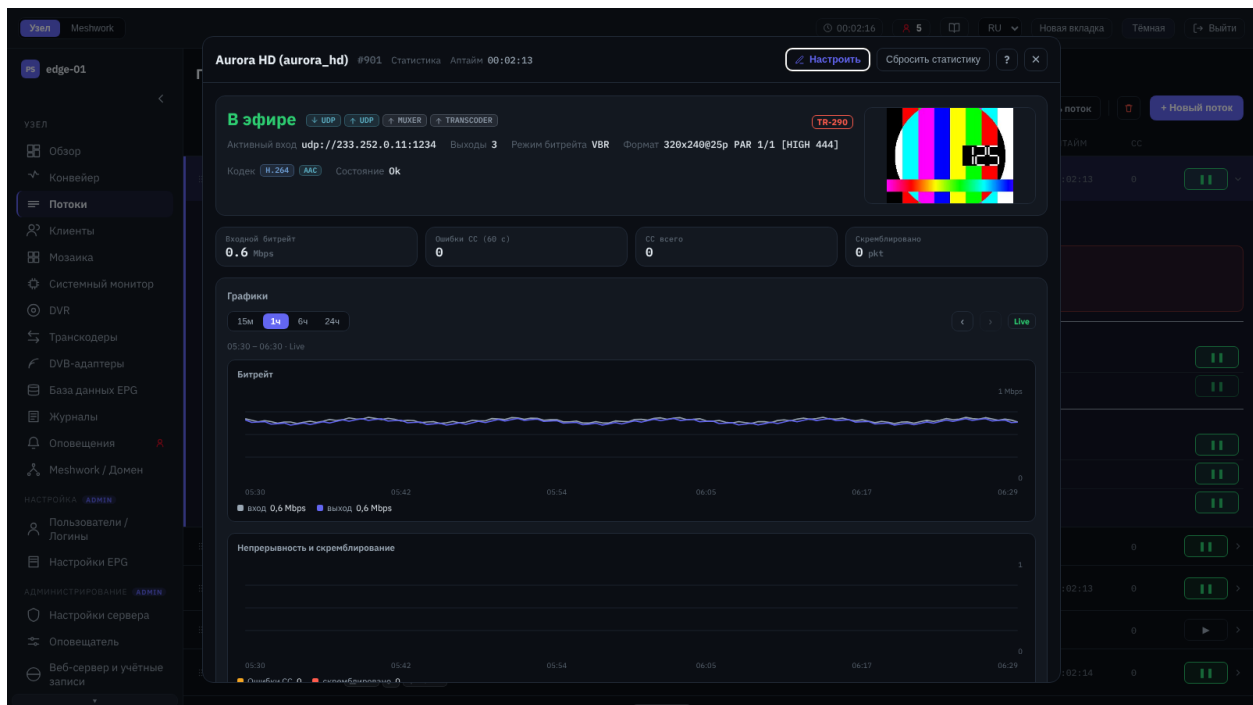


Рис. 22: Окно статистики потокa.

Подробная статистика потокa в отдельном окне, обновляемая в реальном времени. Состав разделов:

Сводка состояния

Текущее состояние и сообщение потокa, плашки признаков, активный вход и число выходов,

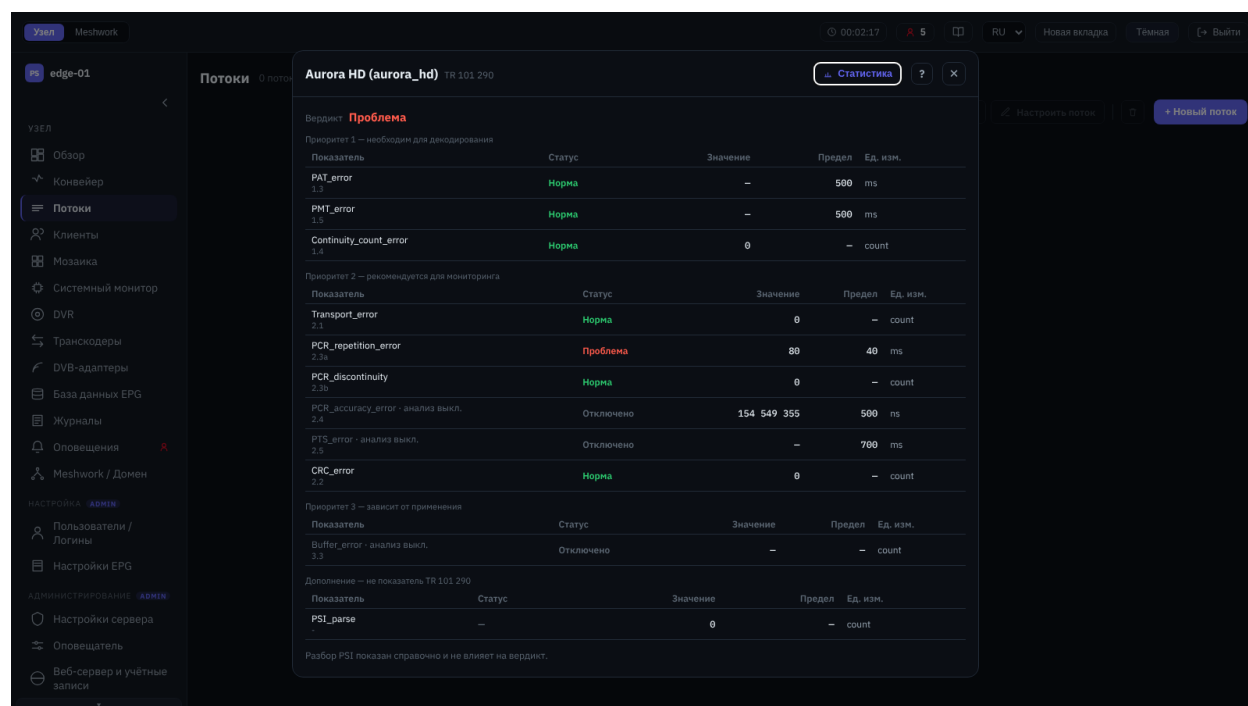


Рис. 23: Отчёт анализа TR 101 290.

режим битрейта, формат и кодеки картинки, ссылки OTT-доставки и живой снимок кадра ([Мозаика](#)). Здесь же — значок вердикта TR 101 290, открывающий полный отчёт, и кнопка «Показать в библиотеке» у активного входа ([Библиотека — этот поток](#)).

Мультиплекс / KPI

Для SPTS — ключевые счётчики (входной битрейт, ошибки CC за окно и всего, скремблированные пакеты); для MPTS — таблица мультиплекса с суммарными скоростями и состоянием программ ([Анализ по программам](#)).

Графики

Временные ряды: битрейт, непрерывность и скремблирование, джиттер PCR, потеря пакетов — с выбором окна и масштаба.

Проверка MPEG-TS

Экспресс-проверка структуры (TS, PAT, PMT, PCR, аудио/видео и их ES), общий вердикт TR 101 290 и признаки сигнализации программы ([Проверка валидности](#)).

Диагностика входа

Раскрывающиеся блоки: «Информация о медиа», «Входное соединение», дешифрование BISS, генерация EIT, «Анализатор» (метрики PCR и синхронизации, а под ними таблица PID активного входа со столбцами «PID», «Тип», «Битрейт» и «Пакеты/с»), заполнение и «Глубокий анализ» — структура GOP, буфер T-STD, паспорт видеокодека, разрывы PTS/DTS, SCTE-35 ([Постоянные измерения](#), [Углублённые анализы](#)). Состав раздела зависит от входа: у входа-муксера раздела нет вовсе — сокета у него нет, а программы показывает таблица мультиплекса; при сетевом приёме готового MPTS остаётся «Входное соединение» с показателями своего транспорта, а «Информация о медиа», «Анализатор» и «Глубокий анализ» скрыты — измерений по одной программе у такого входа нет. Дешифрование BISS показывается только у SPTS: у MPTS счётчики BISS ведутся по программам ([Дескремблирование BISS](#)). Блоки генерации EIT и заполнения появляются сами, когда поток генерирует EIT или добавляет NULL-пакеты.

Выходы

Телеметрия каждого работающего выхода: состояние, назначение, битрейт, счётчики отправленного трафика и показатели, специфичные для транспорта (PRO-MPEG FEC, PS1, RIST, SRT, RTMP). У выходов в транспортах общего каталога домена есть кнопка «Показать в библиотеке» ([Библиотека — этот поток](#)).

OTT / DVR

Шаблоны URL доставки, метрики кольца чанков и архива DVR, графики покрытия архива и реплик субтитров ([OTT и DVR](#), [Воспроизведение архива \(VOD\)](#)).

Профайлер и Журнал потока

Загрузка потоков обработки (доля ядра) и структурированный журнал событий потока.

Отчёт TR 101 290 открывается и как отдельное окно; анализатор в целом описан в [Анализатор](#), монитор TR 101 290 — в [Монитор TR 101 290](#). Администратору доступны сброс накопительной статистики и стирание архива DVR.

Адаптивный набор

Окно настройки адаптивного набора (ABR): имя набора и имя в URL, состав потоков-участников и их битрейты («0» — авто). Участниками могут быть только SPTS-потоки с включённой OTT-раздачей; их добавляют и удаляют прямо в окне, битрейт задаётся для каждого варианта. Набор публикуется одним мастер-плейлистом, по которому клиент выбирает качество. Адаптивная раздача описана в [Адаптивные наборы](#), запись DVR для OTT — в [OTT и DVR](#).

Настройка

Группа **Настройка** в области «Узел» объединяет настройки, не относящиеся к серверным службам: учётные записи доступа и подсистему электронной программы передач (EPG). Экраны группы доступны роли администратора.

Пользователи / Логины

Управление доступом получателей потоков к узлу. Экран содержит вкладки:

Локальные логины

Учётные записи получателей потоков — клиентов OTT-раздачи и реер-протоколов. Для каждой задаются логин и пароль либо «Учётная запись по IP-адресу» (поле «IP-адрес / CIDR»; поддерживаются одиночный адрес и диапазоны), «Разрешённые потоки» (пустой список — разрешены все), лимиты одновременных подключений по протоколам (PS1, SRT, HTTP/HLS), «Дата окончания», пауза и «Префиксный логин» для интеграции с middleware ([12. Интеграция с middleware](#)). Учётные записи входа в веб-интерфейс и их роли настраиваются на экране «Веб-сервер и учётные записи» ([Веб-сервер и учётные записи](#)), роли описаны в [Доступ и роли](#).

Внешние (биллинг)

Авторизация получателей потоков через внешнюю систему: вместо локальной проверки логин подтверждается HTTP-запросом к стороннему серверу. Вверху вкладки — карточка «Служба биллинга» с параметром «Интервал запросов биллинга (с)». Ниже — упорядоченный список «Серверы биллинга»; при проверке логина серверы опрашиваются в порядке списка (порядок задаётся перетаскиванием). Для сервера задаются «Имя», «URL», пароль, «Таймаут (с)», «Учёт сессий», «Префикс логина» и «Суффикс логина», а «Коннектор» в значении «Универсальный (HTTP)» полностью описывает обмен — шаблон HTTP-запроса (query, тело, «Content-Type», заголовки с шаблонными подстановками логина, пароля, IP и т. п.) и разбор ответа («Формат ответа» — HTTP-код статуса, тело JSON или тело XML — с путями к признаку

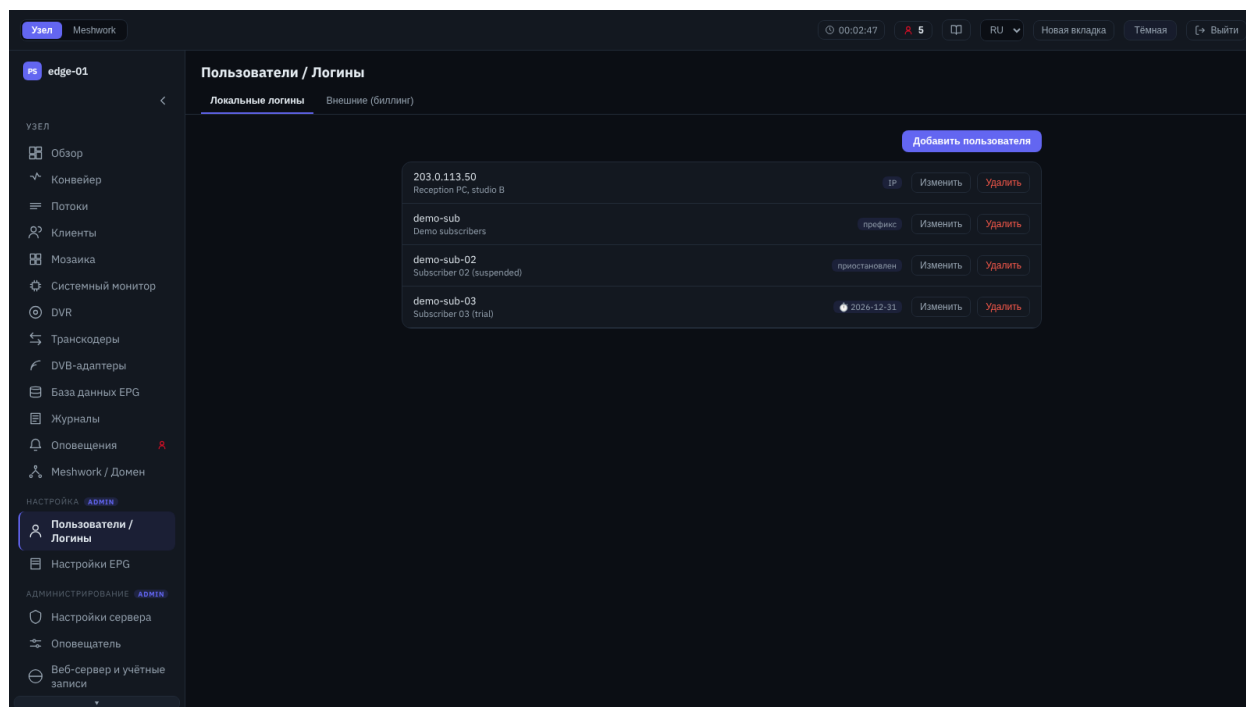


Рис. 24: Пользователи и логины.

успеха и причине отказа); «Трассировка» включает журналирование обмена. «Префиксный логин» как встроенная альтернатива внешнему биллингу описан в [12. Интеграция с middleware](#).

Пользователь: добавление и изменение

Окно создания и правки учётной записи получателя потоков; открывается кнопкой «Добавить пользователя» или «Изменить» в строке списка на вкладке «Локальные логины». Поля:

- «Логин» и «Пароль» — обычная учётная запись; при изменении пустой пароль оставляет текущий. Признак «Учётная запись по IP-адресу» заменяет пару «логин — пароль» полем «IP-адрес / CIDR» (одиночный адрес или диапазон), а «Префиксный логин» включает режим интеграции с middleware ([12. Интеграция с middleware](#)).
- «Разрешённые потоки» — потоки и адаптивные наборы, к которым допущена запись; пустой список ограничений не накладывает — разрешены все.
- «Макс. подключений» — отдельные лимиты одновременных подключений по PS1, SRT и HTTP/HLS.
- «Дата окончания» и «Приостановлен» — плановое или временное отключение доступа без удаления записи.
- «Отображаемое имя» и «Примечание» — для собственного учёта, на доступ не влияют.

Учётные записи входа в веб-интерфейс и их роли задаются не здесь, а на экране «Веб-сервер и учётные записи» ([Веб-сервер и учётные записи](#)).

Настройки EPG

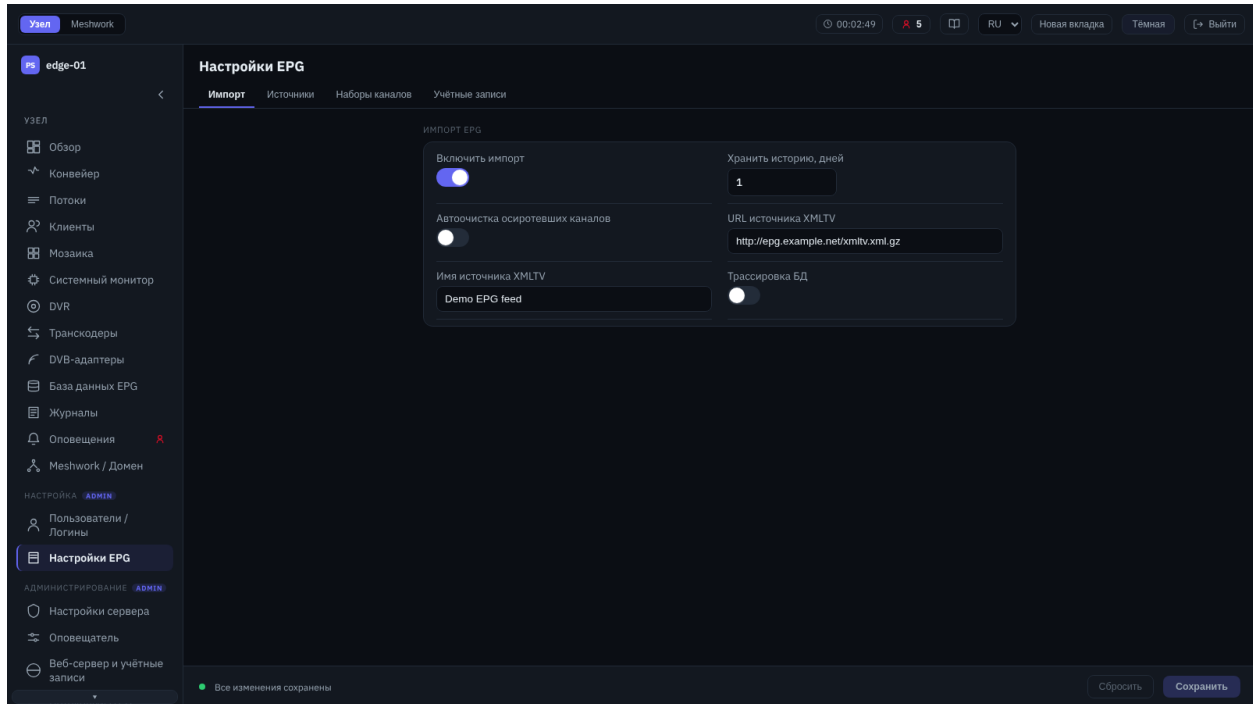


Рис. 25: Настройки EPG (импорт).

Настройка электронной программы передач: откуда берутся данные и как они раздаются. Просмотр накопленной программы вынесен на отдельный экран «База данных EPG» ([База данных EPG](#)). Экран содержит вкладки:

Импорт

Общие параметры импорта программы передач: «Включить импорт», «Хранить историю, дней» (глубина хранения прошедших событий), «Автоочистка осиротевших каналов», а также «URL источника XMLTV» и «Имя источника XMLTV» основного внешнего источника. Импорт EPG в целом описан в [Импорт EPG/XMLTV](#).

Источники

Список внешних источников XMLTV. Для каждого — «Имя», «URL / путь источника», признак «Включено», «Интервал обновления, с», «Кодировка контента» (автоопределение или принудительный GZip), учётные данные и Cookies для закрытых веб-ресурсов и «Сохранять загруженную копию» для диагностики ([Настройка источников](#)).

Наборы каналов

Именованные наборы, группирующие каналы для выдачи внешним потребителям. На вкладке набор создаётся (имя и примечание); его состав — членство каналов — задаётся в базе данных EPG ([База данных EPG](#)). Применение наборов при выдаче описано в [EPG для OTT middleware](#).

Учётные записи

Доступ внешних потребителей к встроенному EPG-серверу — загрузке XMLTV. Для учётной записи задаются логин и пароль, «Набор каналов» (определяет состав выдачи; без набора выдача пуста), «Дата окончания», «Разрешённый IP» и признак «Приостановлен». Выдача XMLTV описана в [Выдача XMLTV](#).

Подсистема EPG в целом — генерация EIT и раздача для OTT middleware — описана в [EPG/EIT](#).

Администрирование

Группа **Администрирование** в области «Узел» — серверные службы узла, хранилища, оборудование, лицензия и обслуживание. Экраны группы доступны роли администратора. Первичная настройка узла описана в [Первый старт и активация](#).

Настройки сервера

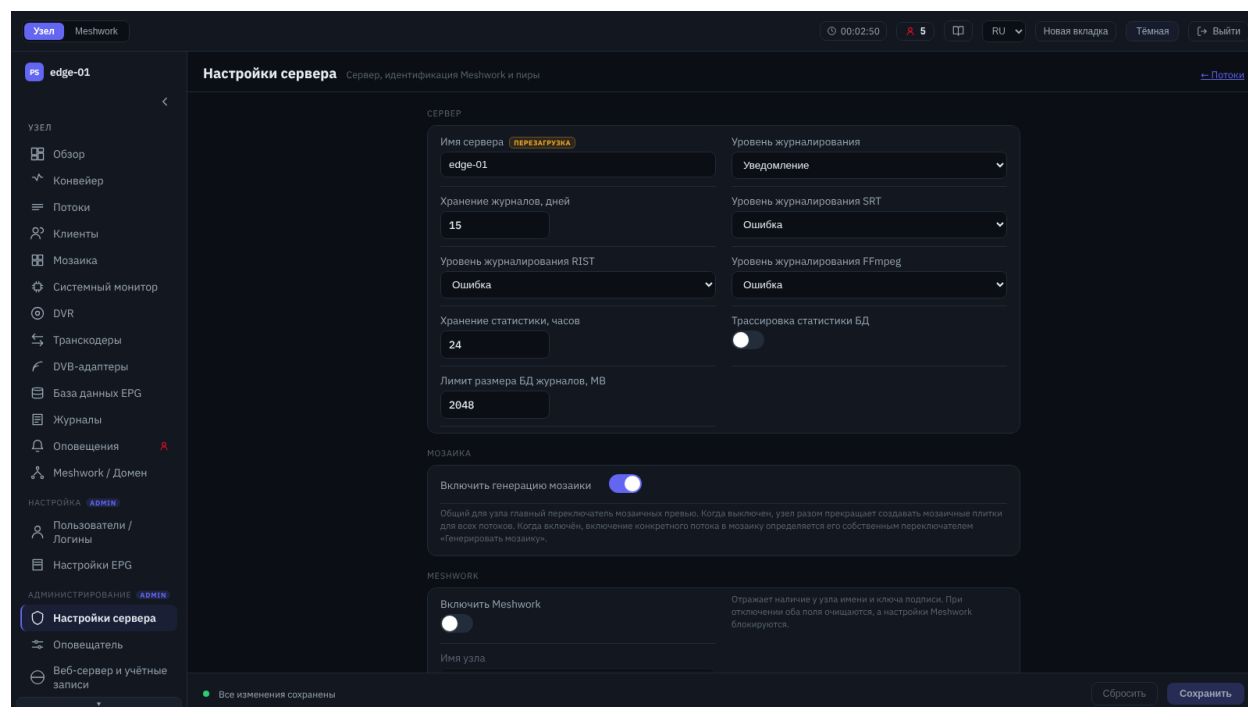


Рис. 26: Настройки сервера.

Общие параметры узла. Экран объединяет несколько групп настроек под одной панелью сохранения:

Сервер

Имя сервера, уровень журналирования (общий и отдельно для библиотек SRT, RIST и FFmpeg), срок хранения журналов, глубина и трассировка статистики БД, лимит размера базы журналов.

Мозаика

Общий для узла выключатель генерации мозаики; сама мозаика — [Мозаика](#).

Meshwork

Включение участия узла в Meshwork и его идентификация: имя узла, заметка, домен Meshwork, а также публичные HTTP- и HTTPS-порты и дополнительные домены — адреса, по которым узел доступен извне ([Узлы за NAT](#)). Модель идентификации — [Идентичность узла](#).

Пирсы Meshwork

Список узлов-пиров и общий секрет подписи для защищённого обмена ([Соседи и секреты](#)).

Контакт Meshwork (расширенное)

Параметры опроса пиров: число рабочих потоков и таймаут контакта.

Первичная настройка узла — в [Начальные настройки](#); развёртывание Meshwork — в [Построение сети Meshwork](#).

Оповещатель

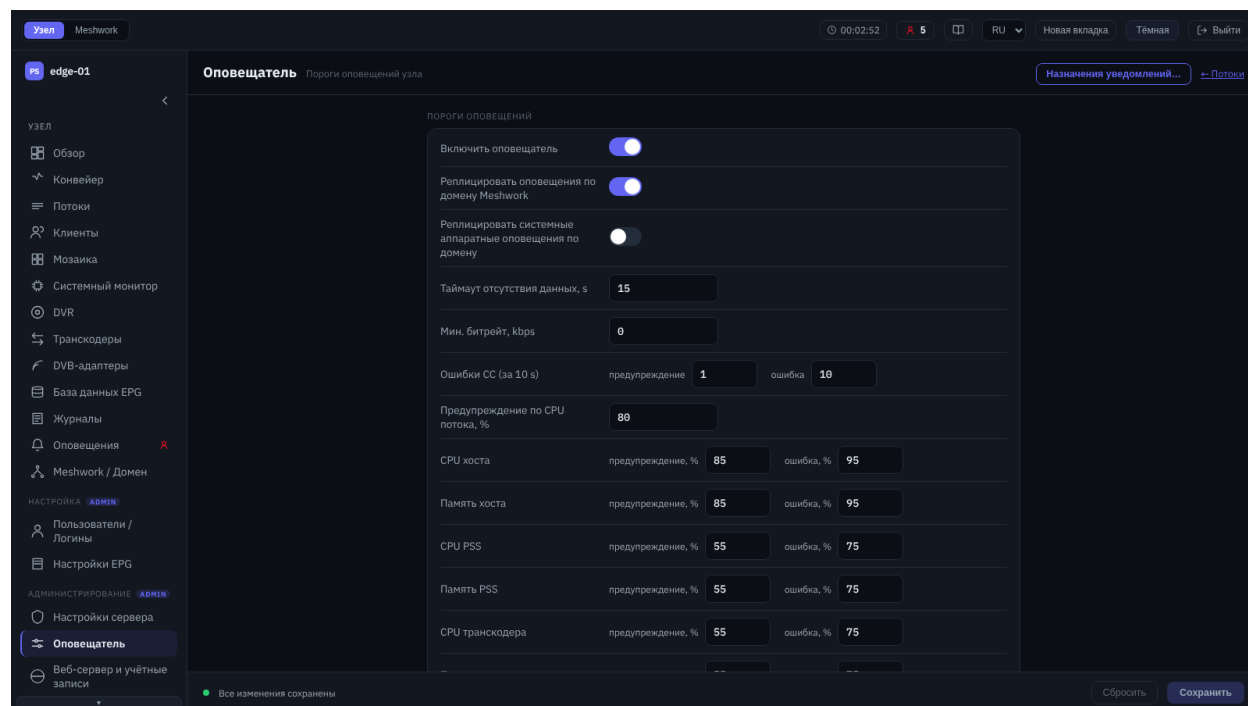


Рис. 27: Настройки оповещателя.

Пороги срабатывания оповещений узла и правила их репликации. Помимо общего включения оповещателя задаются:

- пороги потоков — таймаут отсутствия данных, минимальный битрейт, счётчики ошибок CC (пары «предупреждение / ошибка»);
- отдельное предупреждение по загрузке CPU рабочим потоком, обслуживающим поток или DVB-адаптер (в процентах);
- пороги ресурсов — CPU и память хоста, CPU и память процесса и транскодера, загрузка сети, GPU, использование диска (в процентах, пары «предупреждение / ошибка»);
- время чтения/записи хранилищ DVR;
- при наличии DVB-приёма — пороги качества сигнала фронтендов (уровень сигнала, SNR, частота битовых ошибок, некорректируемые блоки).

Репликация оповещений по домену Meshwork и отдельно — системных аппаратных оповещений — включается здесь же ([Разнос оповещений по домену](#)). Внешняя доставка настраивается в отдельном окне ([Назначения оповещений](#)).

Модель оповещений описана в [Как устроен алерт](#), пороги — в [Настройки алертера](#). Просмотр активных оповещений — на экране [Оповещения](#).

Назначения оповещений

Окно внешней доставки оповещений открывается с экрана [Оповещатель](#). Каналы разнесены по вкладкам; у каждого задаётся свой порог минимальной важности события:

Процесс

Запуск внешней команды на событие оповещения с заданным таймаутом.

Telegram

Отправка в Telegram: токен бота и список ID чатов.

E-mail

Отправка по SMTP: хост и порт сервера, режим TLS (STARTTLS, неявный TLS или без шифрования), проверка сертификата сервера, учётные данные, адреса отправителя и получателей.

Правила доставки описаны в [Внешняя доставка алертов](#).

Веб-сервер и учётные записи

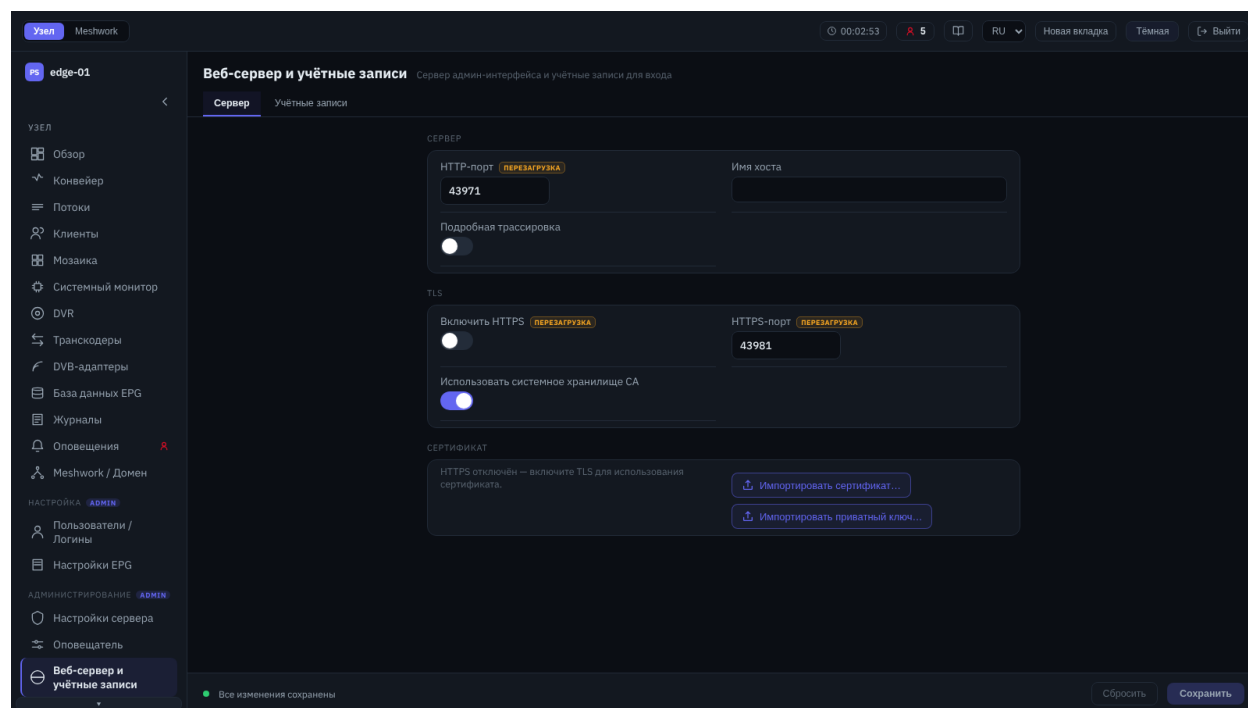


Рис. 28: Веб-сервер и учётные записи.

Настройка веб-сервера панели администрирования и учётных записей входа. Экран содержит вкладки:

Сервер

HTTP-порт и имя хоста, включение HTTPS с отдельным портом и выбором хранилища CA, подробная трассировка. Ниже — сведения о текущем TLS-сертификате (субъект, издатель, срок действия, SAN) с действиями импорта сертификата и приватного ключа и перезагрузки TLS.

Учётные записи

Локальные учётные записи входа в панель и их роли — Admin, Restricted admin, Viewer ([Доступ](#)

и роли). Для каждой задаются логин, пароль и роль; записи добавляются, изменяются и удаляются. При создании пароль обязателен; при изменении поле пароля выводится пустым, и если его не заполнить, пароль остаётся прежним.

Автоматический выпуск сертификатов — на экране [Сертификаты HTTPS](#); перезагрузка TLS без перезапуска — также в [Обслуживание](#). Порты служб и первичная настройка описаны в [Начальные настройки](#).

Потоковый HTTP-сервер

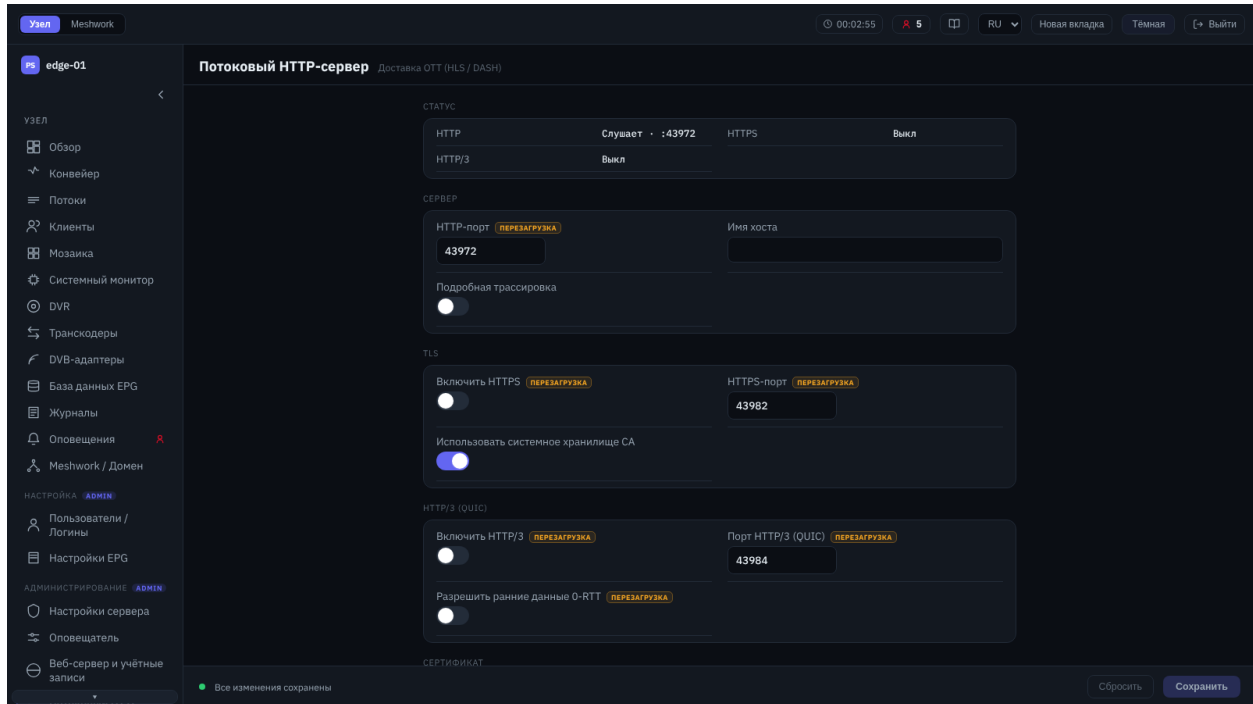


Рис. 29: Потоковый HTTP-сервер.

Настройка встроенного HTTP-сервера отдачи потоков (OTT-раздача, плейлисты и сегменты HLS/DASH). Вверху — статус слушателей (HTTP, HTTPS, HTTP/3). Настраиваются:

Сервер

HTTP-порт, имя хоста, подробная трассировка.

TLS

Включение HTTPS с отдельным портом и выбором хранилища CA.

HTTP/3 (QUIC)

Включение HTTP/3 на отдельном UDP-порту и разрешение ранних данных 0-RTT.

Ниже — панель текущего TLS-сертификата с импортом и перезагрузкой TLS. Раздача описана в [OTT и DVR](#), транспорт HTTP/3 — в [HTTP/3 \(QUIC\)](#).

EPG-сервер

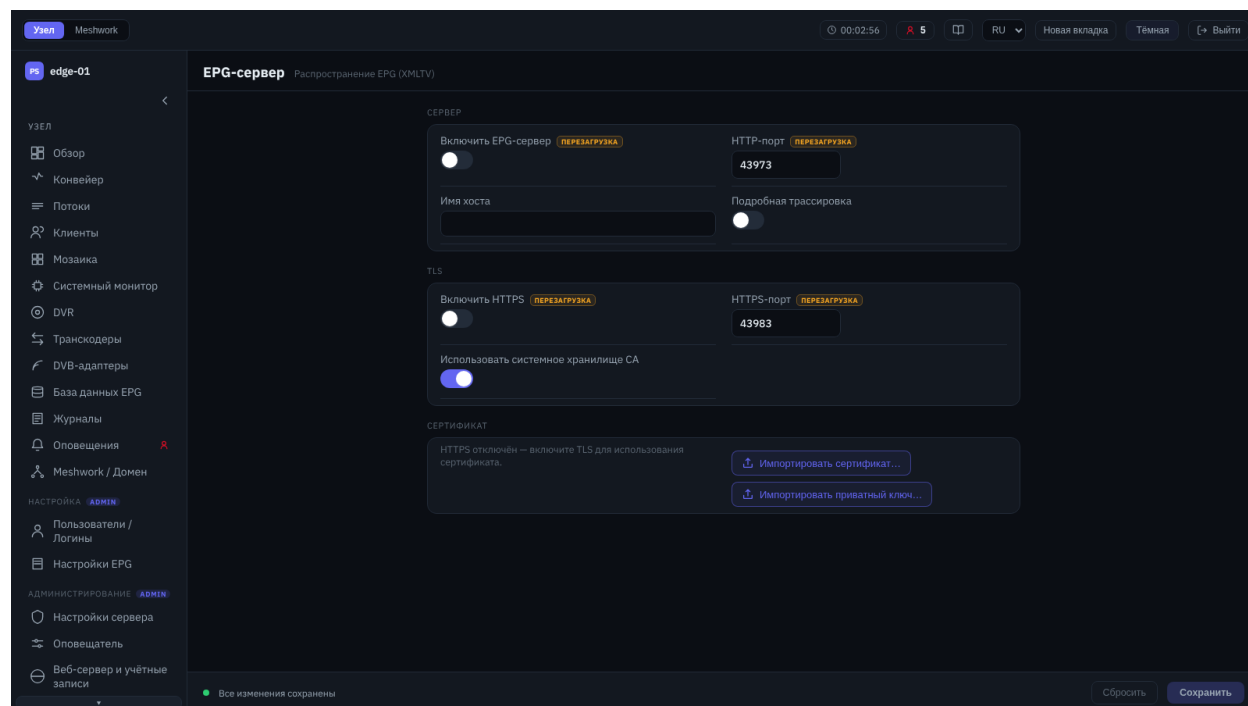


Рис. 30: EPG-сервер.

Настройка службы отдачи EPG (XMLTV) внешним потребителям — middleware и приложениям. Задаются включение службы, порт, имя хоста и подробная трассировка, HTTPS с отдельным портом и хранилищем CA, а также доступна панель TLS-сертификата. Учётные записи доступа к EPG и наборы каналов настраиваются отдельно, на экране [Настройки EPG](#). Интеграция с middleware описана в [EPG для OTT middleware](#).

Сертификаты HTTPS

Автоматический выпуск и обновление TLS-сертификатов через ACME (Let's Encrypt). Вверху — статус: состояние выпуска, время последнего запуска, учётная запись ACME и перечень выпущенных сертификатов со сроками. Настраиваются:

Автоматический выпуск

Включение и контактный e-mail. Включение означает согласие с условиями удостоверяющего центра.

Дополнительно

URL каталога ACME, порт проверки HTTP-01, срок упреждающего обновления и параметры внешней привязки учётной записи (EAB).

Действие «Выпустить / обновить сейчас» запускает выпуск немедленно. Загрузка собственных сертификатов выполняется на экранах соответствующих служб ([Веб-сервер и учётные записи](#), [Потоковый HTTP-сервер](#), [EPG-сервер](#)). Включение TLS описано в [Начальные настройки](#).

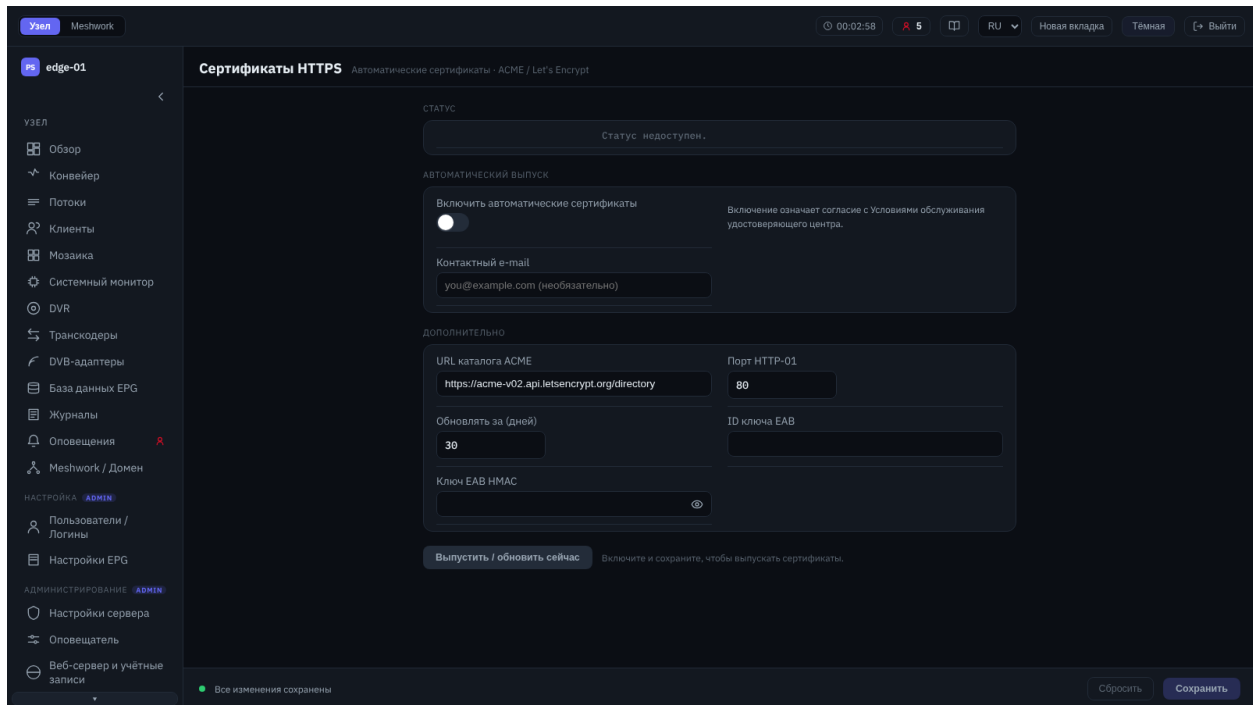


Рис. 31: HTTPS-сертификаты (ACME).

Хранилища DVR

Каталоги архива DVR. Хранилища добавляются, изменяются и удаляются; для каждого задаются имя, путь к каталогу (неизменяемый после создания), примечание и максимальное использование диска. Отдельная группа управляет поведением при нехватке места: интервал очистки, отсрочка и обрезка при переполнении, аварийный запас и гистерезис очистки. Снижение лимита использования обрезаает самый старый архив; удаление хранилища оставляет файлы на диске, но подключённые потоки теряют доступ к VOD. Текущая занятость дисков наблюдается на экране [Монитор DVR](#). Запись и воспроизведение архива описаны в [Хранилища DVR](#) и [OTT и DVR](#).

Обслуживание

Служебные операции узла, сгруппированные по разделам:

Узел

Штатная перезагрузка узла с подтверждением — кратковременно прерывает все потоки.

Сертификаты TLS

Перечитывание TLS-сертификата с диска без перезапуска — по отдельности для веб-сервера, потокового HTTP-сервера и EPG-сервера.

Резервная копия конфигурации

Скачивание gzip-снимка конфигурации узла.

Восстановление

Загрузка ранее сохранённой копии: конфигурация заменяется, узел перезагружается.

Файлы и службы описаны в [Файлы и службы](#).

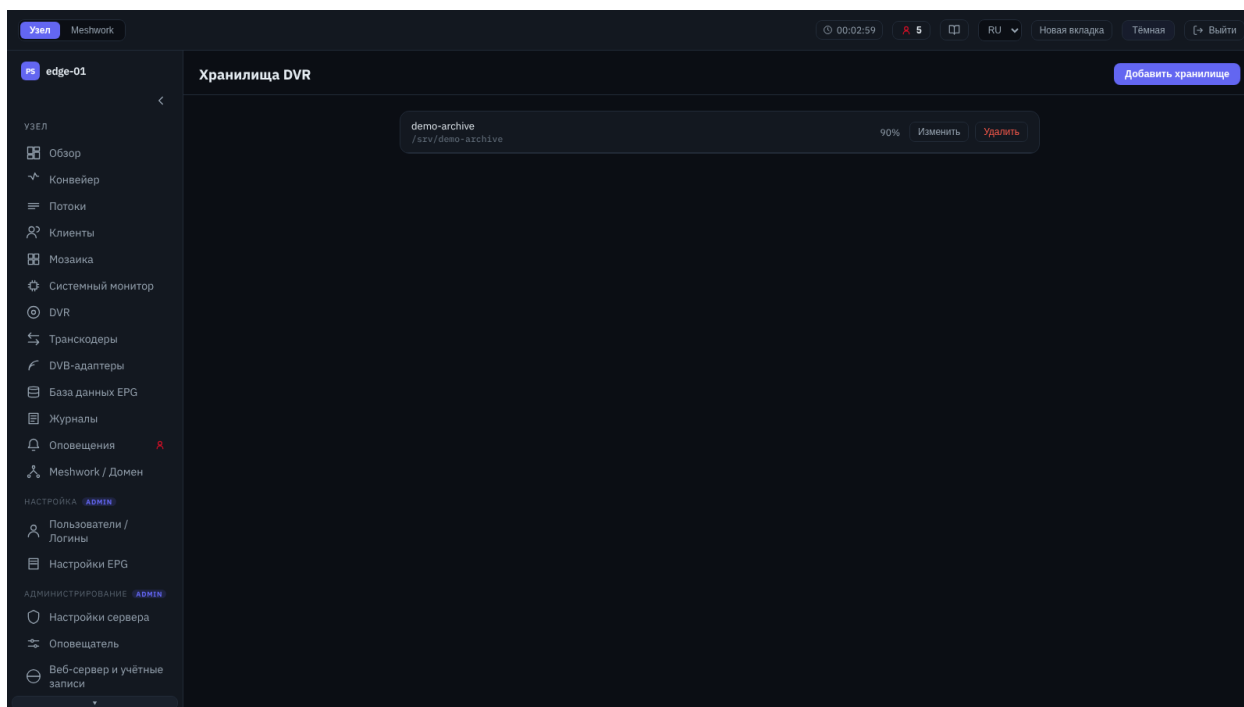


Рис. 32: Хранилища DVR.

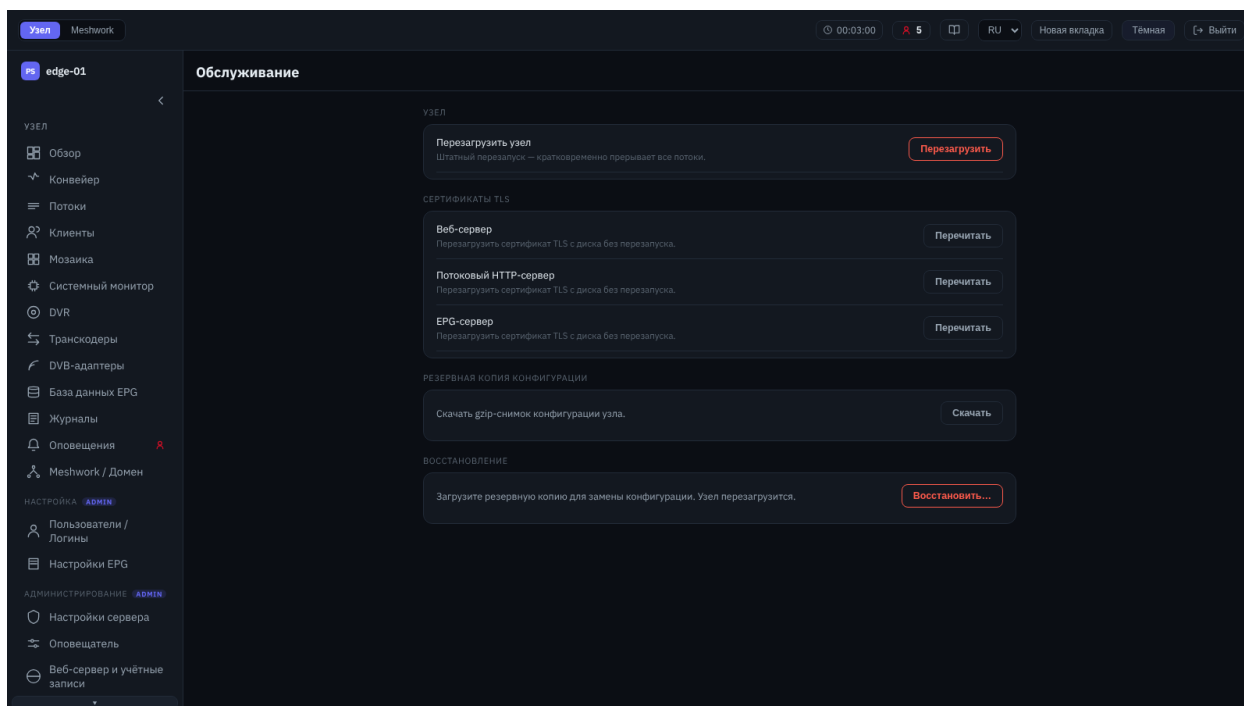


Рис. 33: Обслуживание.

Оборудование

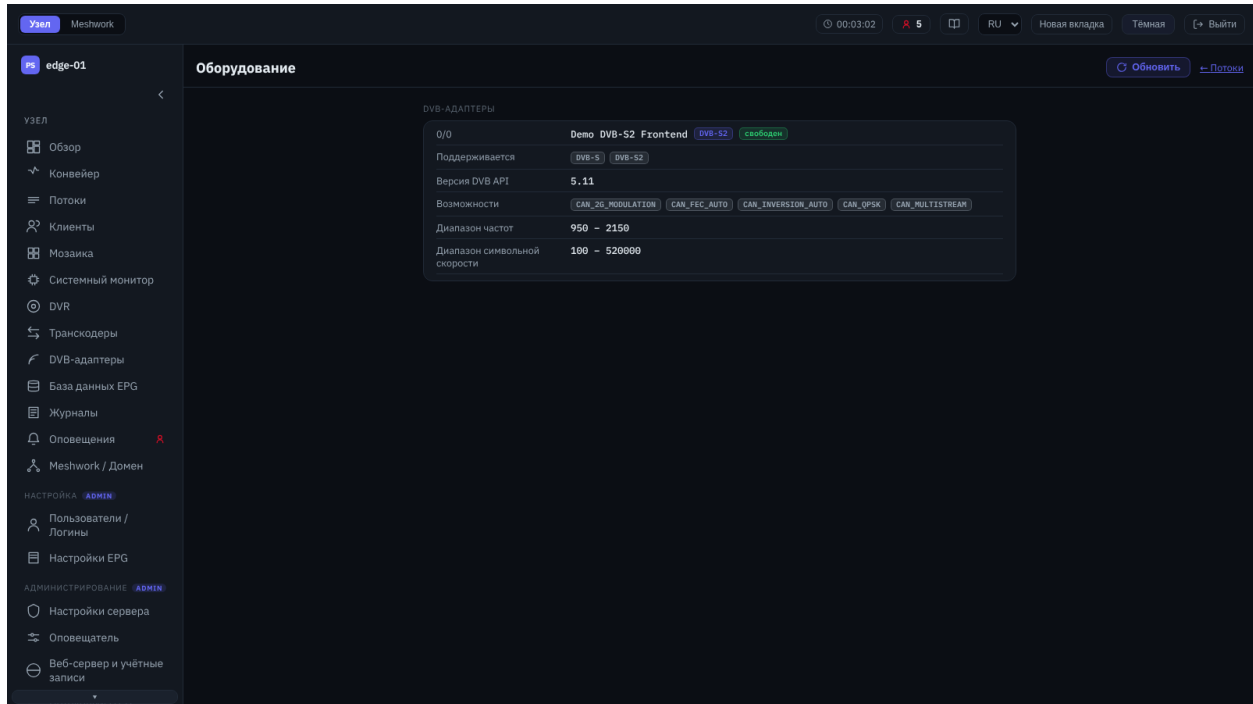


Рис. 34: Оборудование.

Обнаруженное оборудование приёма, доступное узлу (только для чтения; кнопка обновления перечитывает список). Два блока:

DVB-адаптеры

Фронтенды DVB-приёма: название, поддерживаемые системы доставки, версия DVB API, возможности, диапазоны частот и символьной скорости, признак занятости.

CAM-модули

Слоты CI/CAM: вставленный модуль и его модель, тип и состояние слота, идентификаторы CA-систем и статус подписки (дешифрования) по программам.

Приём DVB описан в [DVB-приёмник](#) и [Адаптеры](#), работа с CAM и дешифрование — в [Дескремблирование](#). Обнаруженные устройства транскодирования показаны на экране [О системе](#).

Лицензия

Сведения о лицензии узла: заказчик, тип лицензии, лимиты числа потоков и пиров (0 — без ограничений), демо-режим, начало действия и лицензированное время. При использовании аппаратного ключа показывается блок HASP (идентификатор и тип ключа). Постоянная активация по ключам SL/HL описана в [Постоянная активация](#); поведение при истечении лицензии — в [При истечении срока годовой или пробной лицензии](#).

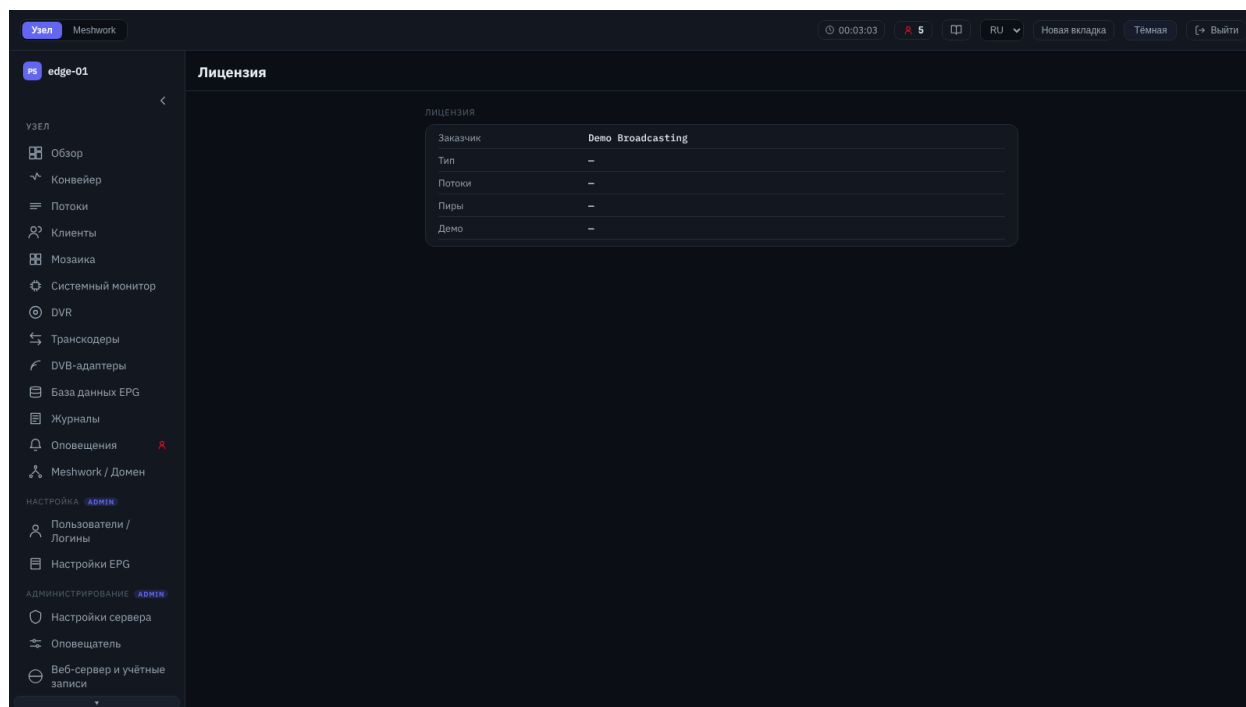


Рис. 35: Лицензия.

О системе

Сводка о системе, сгруппированная по разделам:

Лицензия

Краткие сведения о лицензии — подробности на экране [Лицензия](#).

Транскодеры

Обнаруженные бэкенды транскодирования: тип, версия исполняемого модуля, готовность и перечень устройств. Установка пакетов транскодера — в [Транскодеры](#), выбор бэкенда — в [Бэкенды и кодеки](#).

Сборка и библиотеки

Версия сборки PSS и версии встроенных библиотек, тип сборки.

Идентификация

Имя сервера, узел и его примечание, текущая учётная запись, время работы и момент запуска, ограничение по IP источника.

Вызов документации доступен также из общей панели интерфейса ([Общая панель](#)).

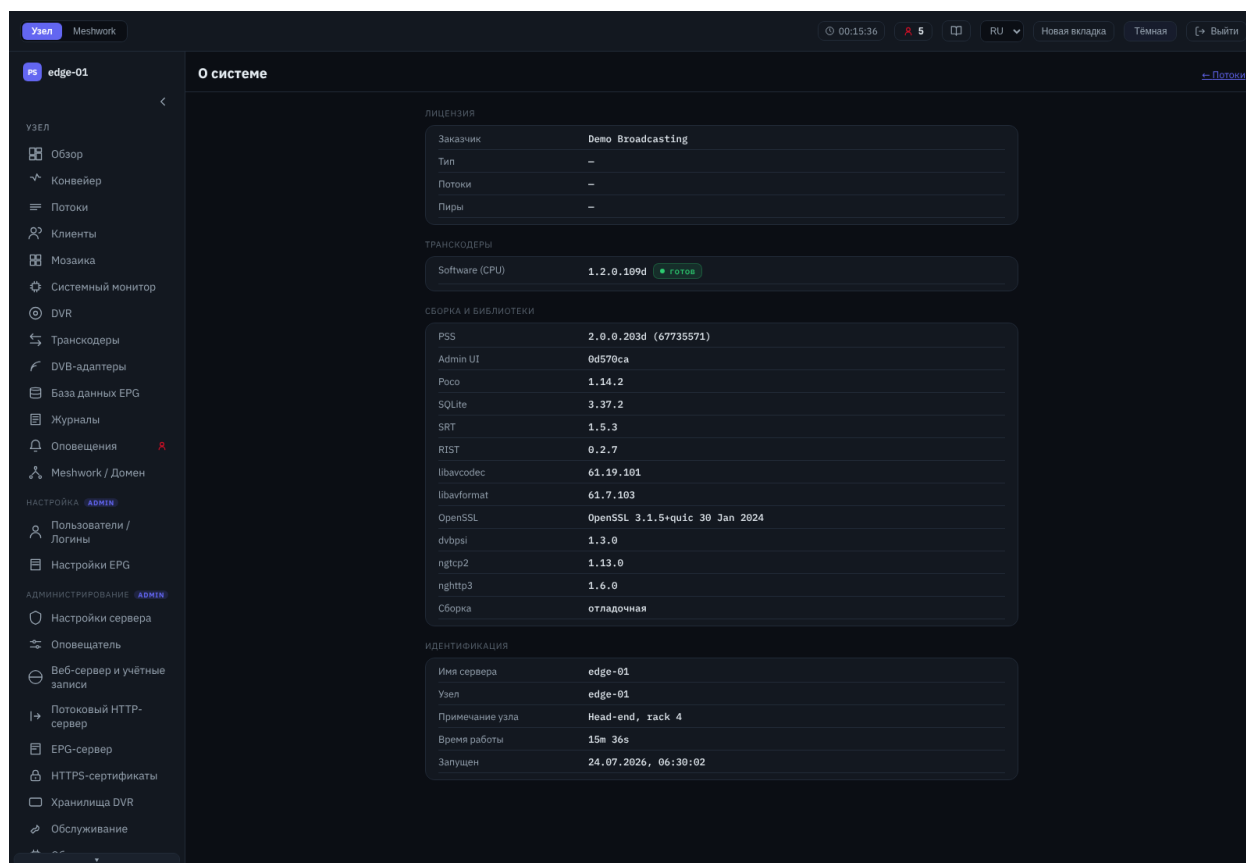


Рис. 36: О системе.

Meshwork

Область **Meshwork** (переключатель области в верхней панели) собирает сводные экраны сети узлов домена: обзор домена, топологию транспорта, состав пиров, отдельные транспортные соединения и общий каталог сетевых ресурсов. Это взгляд «сверху» на всю сеть, в отличие от экранов области «Узел», показывающих один узел.

Понятия сети, домены, автообнаружение соседей, работа за NAT и оповещения описаны в разделе [Meshwork](#).

Обзор Meshwork

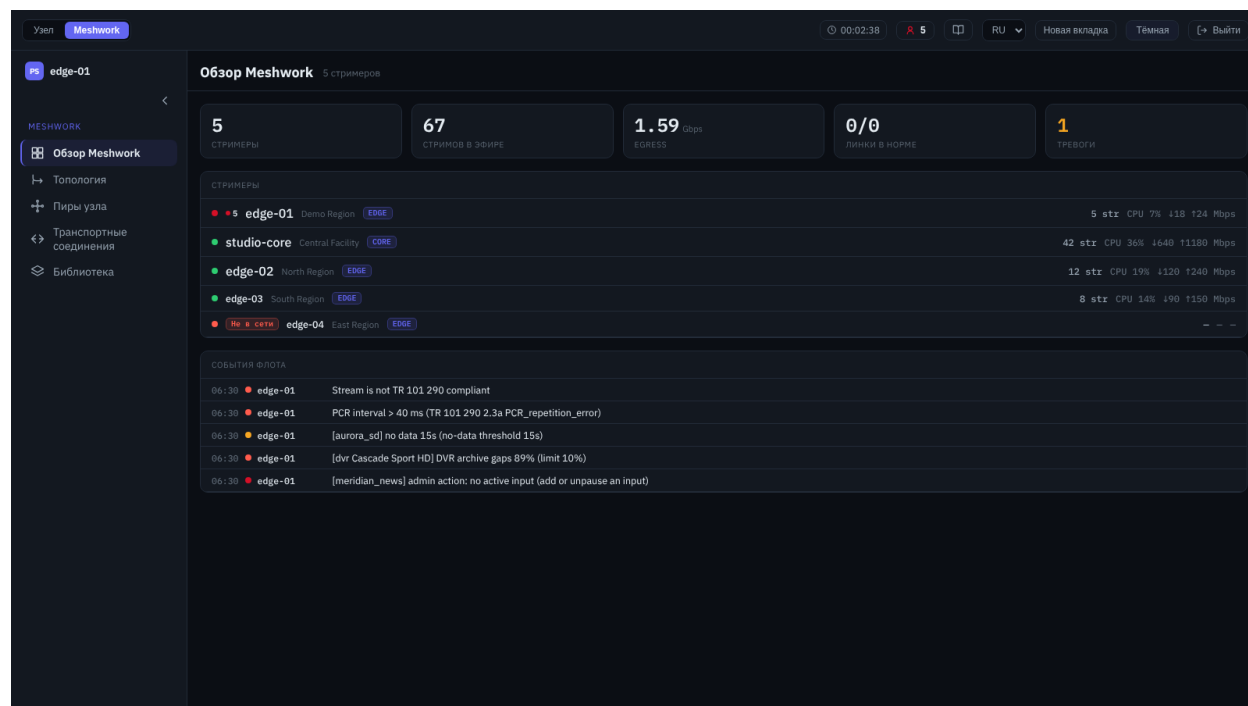


Рис. 37: Обзор Meshwork.

Сводная панель домена: все узлы сети и ключевые показатели домена в одном окне. В шапке — число узлов, ниже — строка показателей и списки.

Строка показателей объединяет плитки «Стримеры» (число узлов), «Стримов в эфире», «Egress» (суммарная исходящая полоса, Gbps), «Линки в норме» (исправные транспортные соединения к их общему числу) и «Тревоги». Плитка «Тревоги» считает неисправные узлы и неисправные активные соединения и подсвечивается при ненулевом значении.

Панель «Стримеры» — по строке на каждый узел: индикатор состояния, а у неисправного узла рядом с ним и словесная метка («Деградация» или «Не в сети»); имя узла, регион и роль, число потоков в эфире, загрузка CPU и трафик (вход/выход). Строка узла с доступной админкой ведёт к его экранам в области «Узел». Для роли администратора рядом с состоянием показывается счётчик активных оповещений узла.

Панель «События флота» видна администратору и сводит активные оповещения по всем узлам: время, узел, к которому относится событие (ссылка на его админку), и текст. Модель сети описана в [Meshwork](#), оповещения — в [Оповещения \(алерты\)](#).

Топология

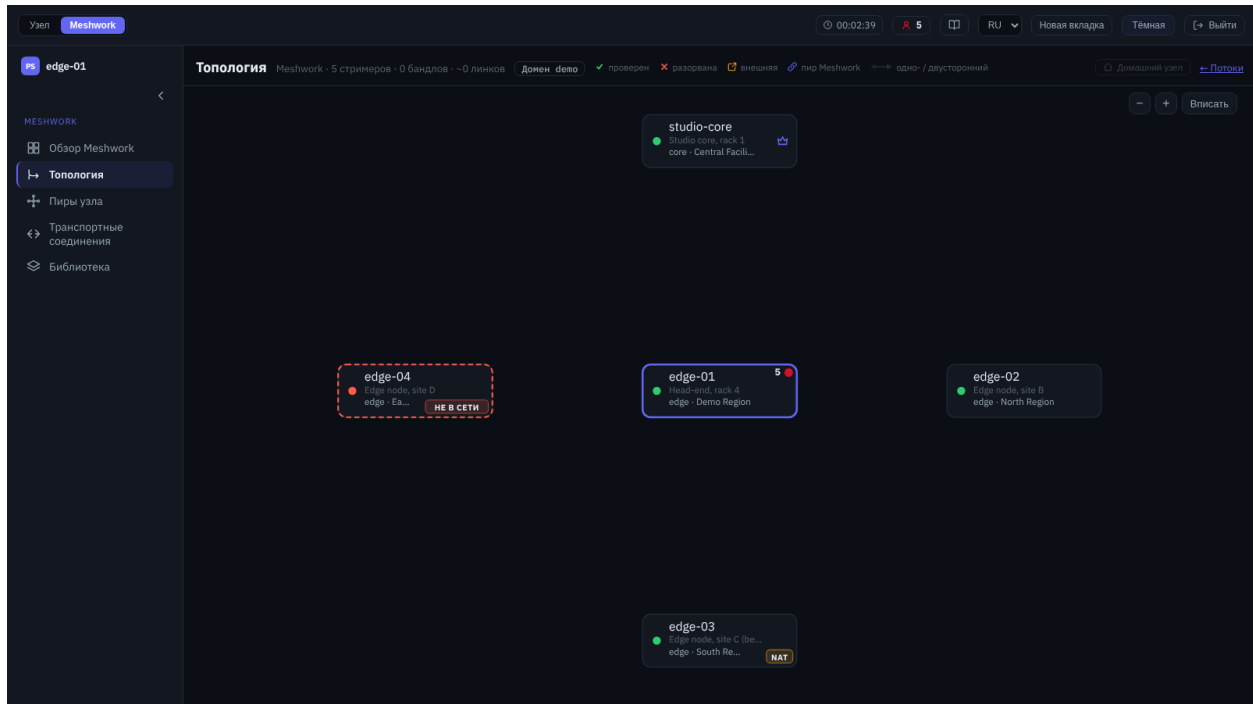


Рис. 38: Граф топологии флота.

Граф транспорта домена: узлы сети и транспортные соединения между ними, сведённые в бандлы (несколько однонаправленных линков одного направления показываются одной связью). Экран графовый; в шапке — число узлов, бандлов и линков, домен подключённого узла, обозначения связей и кнопка «Домашний узел», возвращающая центр графа на текущий узел.

На графе видно, какие узлы соединены, где связь проверена, а где выведена, какие соединения внешние (за пределами домена) и какие узлы находятся за NAT. Обозначения различают проверенную связь, разорванную (пир не в сети), внешнюю связь, связь между пирами Meshwork и направление (одно- или двустороннее). Узел может нести счётчик активных оповещений. Клик по узлу перемещает на него центр графа (повторный клик открывает его админку), клик по связи открывает [Транспортные соединения](#) с фильтром по выбранному бандлу. Карта сети и её обозначения описаны в [Карта сети и каталог ресурсов](#), поля узлов — в [Карта сети](#), работа за NAT — в [Узлы за NAT](#).

Пиры узла

Состав домена и связи между его участниками. Доступны переключатель области графа «Меш / Постоянные» и переключатель отображения «Таблица / Граф» (по умолчанию граф).

В режиме графа «Меш» показывает всех участников домена с центром на подключённом узле: достоверную звезду keер-alive от него и выведенные — не проверенные с этого узла — связи между остальными парами. «Постоянные» строит граф настроенного пиринга: узлы и направленные связи из списков соседей каждого узла (стрелка — направление настройки, двусторонняя при взаимной). Кнопка «Домашний узел» возвращает центр на текущий узел.

В режиме таблицы перечислены участники со столбцами «Узел», «Адрес», «Тип» («Постоянный» или «Авто»), «Версия», «Статус» («Активен» / «Недоступен», с меткой «Деградация» при сбоях

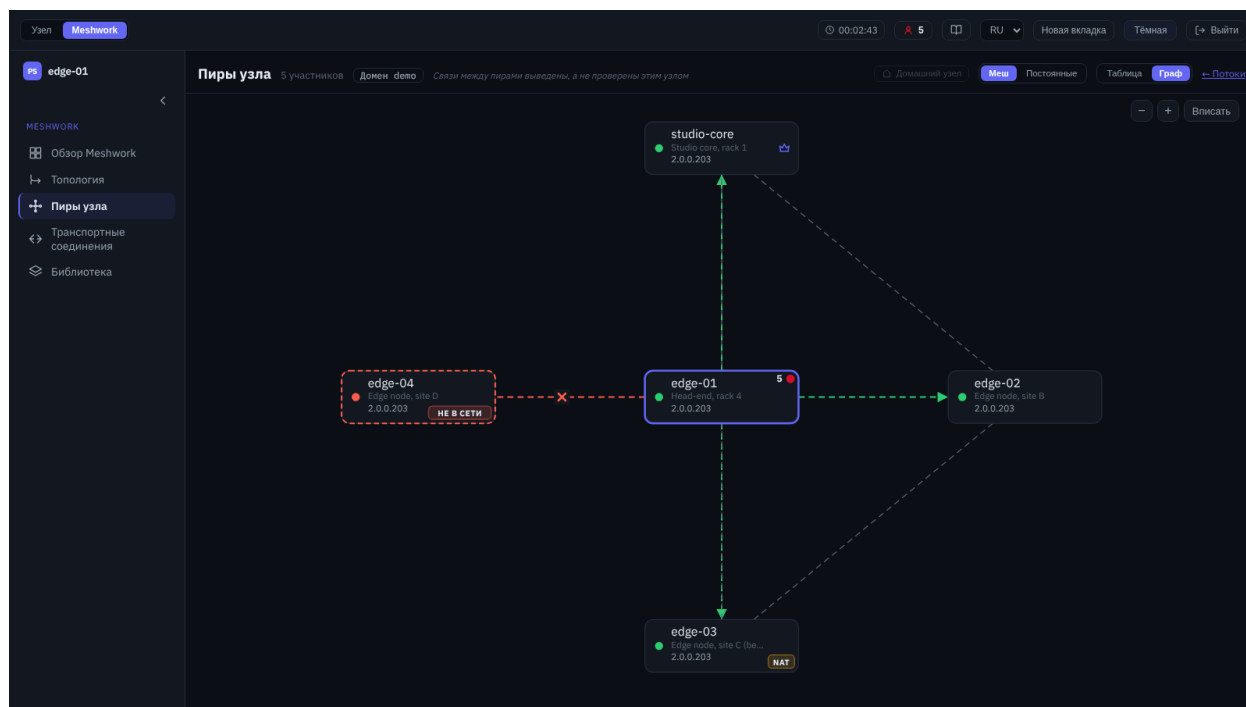


Рис. 39: Граф пиров узла (mesh).

keep-alive), «NAT» и «Последняя активность». Узел за NAT показывает «Внутренняя сеть» вместо недостижимого частного адреса. Имя узла — ссылка на его админку (недостижимый пир выводится обычным текстом); в строке есть кнопка открытия админки. Строка текущего узла помечена и раскрывает панель «Трафик Meshwork» — разбивку исходящего трафика по долям («База», «Пирры», «Библиотека», «Алерты»), скорость TX/RX и счётчики обмена. Поля строк описаны в [Карта сети](#), настройка соседей — в [Соседи и секреты](#), работа за NAT — в [Узлы за NAT](#). Взгляд на пирры с точки зрения текущего узла — экран [Meshwork / Домен](#).

Транспортные соединения

Отдельные транспортные соединения между узлами домена — те самые линки, которые на «Топологии» сведены в бандлы. В шапке — показатели «Соединения» (всего), «Исправные» (к общему числу) и «Проверенные». При переходе с графа «Топологии» таблица открывается с фильтром по выбранному бандлу; фильтр снимается ссылкой « все соединения».

Каждая строка описывает одно соединение: индикатор состояния, направление (столбец «Соединение», узел → узел), «Протокол» (PS1, SRT и другие транспортные каналы), «Поток», «Идентификацию» («проверено» или «предполагается», с пометкой «резерв» для резервного линка), «Область» («внутренняя», «внешняя», «локальная» или «неизвестно») и телеметрию — «TX→RX Mb/s», «RTT» и «Потери». Область берётся со стороны, которая подключается сама, поэтому у соединений UDP, RTP, Pro-MPEG и RIST она «неизвестно». Значения телеметрии появляются при наличии соответствующих данных, иначе показывается «—»; длинные списки разбиваются на страницы. Тип связи PS1 / SRT описан в [Общий каталог ресурсов](#), сами протоколы — в [Рег-протоколы надёжной передачи](#).

Библиотека

Узел	Домен	Вид	Протокол	Назначение	Поток	Примечание
edge-01	demo	input-listen	SRT	192.0.2.10:9101	meridian_news	Contribution from...
edge-01	demo	input-multicast	UDP	233.252.0.11:1234	aurora_hd	—
edge-01	demo	output-multicast	UDP	233.252.0.111:1234	aurora_hd	—
edge-01	demo	output-multicast	UDP	233.252.0.112:1234	meridian_news	—
edge-01	demo	output-multicast	UDP	233.252.0.116:1234	aurora_sd	—
edge-01	demo	input-multicast	UDP	233.252.0.13:1234	cascade_sport	—
edge-01	demo	input-multicast	UDP	233.252.0.15:1234	solstice_radio	—
edge-01	demo	output-multicast	UDP	233.252.0.190:1234	demo_mux_a	Multiplex A to the...
edge-01	demo	input-multicast	UDP	233.252.0.211:1234	aurora_hd	Standby feed
edge-02	demo	input-multicast	UDP	233.252.0.11:1234	aurora_relay	—
edge-02	demo	output-multicast	PRO-MPEG	233.252.0.61:5004	edge02_backhaul	FEC to the head-e...
edge-02	demo	output-multicast	RIST sec 5	233.252.0.70:5000	edge02_rist	—
edge-03	demo	input-multicast	UDP	233.252.0.11:1234	aurora_nat_edge	—

Рис. 40: Библиотека ресурсов домена.

Общий каталог занятых сетевых ресурсов домена: входы и выходы узлов по протоколам UDP, RTP, Pro-MPEG, RIST, а также PS1 и SRT. Каталог формируется автоматически из объявлений узлов и доступен только для просмотра. По нему выбирают свободные адреса и порты и находят нужные потоки в сети. В шапке — общее число записей и ссылка «[Потоки](#)».

Над списком — фильтр «Узел-источник»: по умолчанию «Все узлы», в списке значений — узлы, которые сейчас что-либо объявляют. Фильтр отбирает записи одного объявившего узла и не меняет число записей в шапке. Строки упорядочены по узлу, домену и адресу.

Столбцы: «Узел» — объявивший узел, «Домен», «Вид» (направление и способ подключения точки: output-multicast, input-caller, output-listen и т. п.), «Протокол», «Назначение» (адрес и порт), «Поток» (имя либо номер) и «Примечание». К адресу приписываются область связи («локальная», «внутренняя» или «внешняя»; тот же признак есть у транспортных соединений — [Транспортные соединения](#)), источник SSM (SSM <адрес>) и номер VLAN (VLAN <n>). Область связи выводится только у точек PS1 и SRT, которые сами подключаются к заданному адресу (*-caller): «локальная» — адрес того же узла, «внутренняя» — подтверждённый доменной авторизацией пир, «внешняя» — все остальные. У RIST рядом с протоколом выводится вес линка (вес <n>).

Клик по строке открывает админку её узла на статистике её потока: строка своего узла открывается на месте, строка соседа — по настройке «Открывать узлы в» общей панели ([Общая панель](#)); Ctrl или [⌘](#) принудительно открывает новую вкладку. Строка узла, недоступного или неизвестного сети, действия не имеет и несёт подсказку «Узел недоступен». Подсказка строки показывает время объявления записи.

Каталог и правила выбора адресов описаны в [Общий каталог ресурсов](#); взгляд со стороны отдельного входа или выхода — [Библиотека — этот поток](#).

1.9 Дополнительные материалы

Приложения для эксплуатации: масштабирование OTT-раздачи через кеширующие прокси и CDN, подключение внешних систем мониторинга, оптимизация узла под большие инсталляции и требования к оборудованию Intel для аппаратного транскодирования.

1.9.1 Кеширование и CDN для OTT-раздачи

Приложение для эксплуатации крупных инсталляций: как устроено кеширование OTT-ответов Perfect Streamer и как поставить перед узлом кеширующий reverse проху или CDN. Режимы раздачи, форматы URL и авторизация описаны в основном разделе [OTT и DVR](#).

Сервер формирует ответы трёх категорий, различающихся сроком жизни содержимого и пригодностью для кеширования промежуточными узлами (reverse проху, CDN, клиентский кеш).

1. Модель кеширования

1.1. Ресурсы и HTTP-заголовки

Ресурс	URL	Content-Type	Cache-Control
TS-сегмент (HLS)	/h<sess>/<keyHex>.ts	video/mp2t	public, max-age=60, immutable
VTT-сегмент (субтитры)	/h<sess>/sub/<pid>/<keyHex>.vtt	text/vtt; charset=utf-8	public, max-age=60, immutable
fMP4-сегмент (DASH / LL-HLS)	/h<sess>/init.mp4, /h<sess>/<key N>.m4s	video/mp4	public, max-age=60, immutable
DASH MPD	/h<sess>/index.mpd	application/dash+xml; charset=utf-8	public, max-age=1
HLS master	/hls/<stream>/<login>/<pass>/index.m3u8	application/vnd.apple.mpegurl	public, max-age=1
HLS media	/h<sess>/index.m3u8, /h<sess>/sub/<pid>/index.m3u8	application/vnd.apple.mpegurl	public, max-age=1
302 Redirect	/dash/<stream>/<login>/<pass>/index.mpd	—	no-cache, no-store
Raw TS	/http/<stream>/<login>/<pass>	video/mp2t	не задан; не кешируется

1.2. Характеристики сегментов

Шестнадцатеричный идентификатор сегмента в URL (<keyHex> в путях /h<sess>/<keyHex>.ts) формируется как CRC64 от стартового времени сегмента и ID потока и глобально уникален. URL сегмента адресует неизменное содержимое — при повторных запросах того же URL возвращается идентичный байтовый поток (пока сегмент остаётся в пределах скользящего окна).

Директива `immutable` подавляет условную ревалидацию клиентом (`If-None-Match`, `If-Modified-Since`). Значение `max-age=60` обеспечивает совместимость с типичным `timeShiftBufferDepth=40s`.

fMP4-сегменты CMAF (.m4s) и общий `init.mp4` для DASH / Low-Latency HLS адресуются аналогично и кешируются по той же модели (`immutable`, `max-age=60`). Частичные сегменты (parts) LL-HLS — это `byte-range`-запросы внутрь того же .m4s, поэтому отдельной записи в кеше не образуют.

1.3. Характеристики манифестов

`max-age=1` ограничивает верхнюю границу устаревания содержимого в кеше одной секундой. При совместном использовании с `proxy_cache_lock on` (nginx) всплески запросов к манифесту коалесцируются в один запрос на `origin` в секунду.

1.4. Вариативность содержимого

При `absPath=0` (значение по умолчанию, URL-параметра `a` нет) манифесты HLS media и DASH MPD не содержат идентификатора сессии в теле. Содержимое манифеста идентично между сессиями, принадлежащими одной (`stream`, `param`)-комбинации. Это позволяет `reverse-proxy` кешу переиспользовать запись между сессиями при нормализации ключа кеша.

При `absPath=1` (URL-параметр `a=1`) в теле манифеста присутствуют абсолютные URL, включающие схему, хост и идентификатор сессии. Содержимое становится специфичным для сессии, возможность кросс-сессионного переиспользования кеша отсутствует.

2. Поведение клиентов

Клиент	URL обновления манифеста	Воздействие на количество сессий
HLS-плееры по media playlist	/h<sess>/index.m3u8	Одна сессия на сеанс воспроизведения
DASH-плееры по корневому URL	/dash/<stream>/.../index.mpd (цикл повтора)	Обрабатывается session reuse (см. 3.2)
Браузерные плееры (MSE)	/h<sess>/... через <Location> / session URL	Одна сессия на сеанс воспроизведения

3. Специальные механизмы

3.1. HTTP 302 Redirect для DASH

Запрос вида `/dash/<stream>/<login>/<pass>/index.mpd` возвращает ответ 302 Found с заголовком `Location: /h<sess>/index.mpd`. Тело ответа пустое. Авторизация и выделение сессии выполняются на этапе обработки редиректа.

Клиенты, поддерживающие кеширование редиректа, обращаются к session URL напрямую в последующих запросах. Клиенты, не поддерживающие его, повторяют запрос редиректа. Стоимость повторной обработки редиректа ограничена проверкой аутентификации и операциями session reuse.

3.2. Session reuse для DASH

При обработке запроса `/dash/.../index.mpd` от того же логина к тому же потоку (с тем же признаком adaptive) сервер находит уже существующую DASH-сессию и возвращает её идентификатор повторно. Новая сессия не создаётся, слот лимита одновременных подключений не расходуется. Переиспользуются только сессии в одном режиме воспроизведения: живая сессия и VOD-сессия архива (параметры `t/d/epg`) не объединяются.

Применяется только к DASH. Для HLS отдельный механизм reuse не требуется: HLS-клиенты обновляют media playlist по session-URL и не создают новую сессию на каждом обновлении.

3.3. Переиспользование сегментов между сессиями

Путь `/h<sess>/<keyHex>.ts` не зависит от `<sess>` при разрешении `<keyHex>` в содержимое: `<keyHex>` глобально однозначно идентифицирует TS-сегмент в рамках потока. Nginx с нормализованным cache key (обрезающим префикс `/h<sess>/`) обслуживает все запросы одного и того же `<keyHex>` из единственной записи кеша, независимо от того, какие клиенты их выдали.

Дедупликация применима только к content-addressed именам — сегментам с 16-символьным шестнадцатеричным идентификатором (`.ts`, `.m4s` с `<keyHex>`, `.vtt`). Нумерованные сегменты live-DASH (`<номер>.m4s`), `init.mp4` и манифесты уникальны лишь в рамках своей сессии — их ключ кеша должен сохранять префикс `/h<sess>/`, иначе кеш будет отдавать содержимое одного потока зрителям другого. Для DASH дедупликацию манифестов между клиентами одного логина обеспечивает session reuse (см. 3.2).

4. Параметры запроса

Параметр	Значение по умолчанию	Влияние
a	0	1 — абсолютные URL в манифестах; 0 — относительные
s	40	timeShiftBufferDepth в секундах
m	40	Минимальная длина окна для выдачи манифеста
v	6 для режимов OTT, 3 для Peering / HLS	#EXT-X-VERSION в HLS (игнорируется DASH); явное значение в URL переопределяет значение по умолчанию
h3	отсутствует (off)	opt-in на HTTP/3 (QUIC): заставляет сервер выдать Alt-Svc в ответе; sticky на session-URL. См. HTTP/3 (QUIC)

Изменение параметра через query string обновляет сохранённые в сессии значения при её следующем переоткрытии. Параметры воспроизведения архива t, d и ерг описаны в [Воспроизведение архива \(VOD\)](#).

5. Нагрузочные характеристики

Нагрузка на origin масштабируется с числом одновременно наблюдаемых различных потоков. Увеличение числа клиентов, наблюдающих один и тот же поток, не увеличивает количество запросов к origin при наличии reverse-proxy кеша и нормализованного cache key.

Сценарий	Частота origin-запросов (реф.)
1 клиент на поток X	MPD: 0.4 req/s, segment: 0.2 req/s
N клиентов на один поток X (кеш включён)	MPD: 1 req/s, segment: 0.2 req/s
N DASH-клиентов в режиме повтора на одном потоке	MPD: 1 req/s (с proxy_cache_lock)
N клиентов на N различных потоков	MPD: 0.4·N req/s, segment: 0.2·N req/s

Сегменты дедуплицируются глобально по content-addressed именам; дедупликацию манифестов между клиентами обеспечивают session reuse (DASH) и кеш в пределах сессии.

6. Nginx как кеширующий reverse proxy

6.1. Базовая конфигурация

```

proxy_cache_path /var/cache/nginx/pss_segments
    levels=1:2 keys_zone=pss_segments:100m
    max_size=20g inactive=30m use_temp_path=off;

proxy_cache_path /var/cache/nginx/pss_manifests
    levels=1:2 keys_zone=pss_manifests:10m
    max_size=256m inactive=5m use_temp_path=off;

upstream pss_backend {
    server 127.0.0.1:43972;
    keepalive 64;

```

(continues on next page)

```

}

map $uri $pss_cache_key {
    "~^/h[0-9a-f]{16}(?<tail>(/[0-9]+)?/[0-9a-f]{16}\.(ts|m4s)|/sub/[0-9]+/[0-9a-f]{16}\.vtt)$" "stream:$tail";
    default $uri;
}

server {
    listen 80;
    server_name stream.example.com;

    location ~* "^/h[0-9a-f]{16}(/[0-9]+|/sub/[0-9]+)?/([0-9a-f]+\.(ts|m4s|vtt)|init\.\mp4)$" {
        proxy_cache                pss_segments;
        proxy_cache_key             $pss_cache_key;
        proxy_cache_valid           200 60s;
        proxy_cache_valid           404 403 0s;
        proxy_cache_lock            on;
        proxy_cache_use_stale       updating error timeout;
        proxy_cache_revalidate      on;
        proxy_ignore_headers       Vary;
        add_header                  X-Cache-Status $upstream_cache_status;

        proxy_pass                  http://pss_backend;
        proxy_http_version          1.1;
        proxy_set_header            Connection "";
        proxy_buffering              on;
    }

    location ~* "(^/h[0-9a-f]{16}(/[0-9]+|/sub/[0-9]+)?/index\.(m3u8|mpd)$|^/(hls|dash|llhls)/.*\.(m3u8|mpd)$)" {
        proxy_cache                pss_manifests;
        proxy_cache_key             $pss_cache_key;
        proxy_cache_valid           200 1s;
        proxy_cache_valid           404 403 0s;
        proxy_cache_lock            on;
        proxy_cache_lock_timeout    2s;
        proxy_cache_use_stale       updating;
        add_header                  X-Cache-Status $upstream_cache_status;

        proxy_pass                  http://pss_backend;
        proxy_http_version          1.1;
        proxy_set_header            Connection "";
    }

    location / {
        proxy_pass                  http://pss_backend;
        proxy_http_version          1.1;
        proxy_set_header            Connection "";
        proxy_set_header            X-Forwarded-Proto $scheme;
        proxy_set_header            X-Forwarded-Host $host;
        proxy_buffering              off;
        proxy_read_timeout          3600s;
    }
}

```

Порт origin в примере — порт потокового HTTP-сервера по умолчанию (43972; HTTPS — 43982). Нормализуются только content-addressed имена сегментов (см. 3.3); манифесты, init.mp4 и нумерованные сегменты кешируются по полному URI своей сессии. Директива proxy_ignore_headers Vary в location сегментов оправдана: сервер сопровождает ОТТ-ответы заголовком Vary: Origin, и без неё кеш делился бы на варианты по значению Origin клиента (см. 7).

6.2. Назначение директив

Директива	Назначение
proxy_cache_lock on	Сериализует выполнение upstream-запросов при одновременных cache miss по одному ключу
proxy_cache_use_stale updating	Возвращает устаревшую копию параллельным запросам в период обновления кеша
proxy_cache_revalidate on	Использует If-Modified-Since при cache miss с сохранённой копией
proxy_cache_valid 404 403 0s	Запрещает кеширование ошибок авторизации и 404
keepalive 64 в upstream	Поддерживает пул persistent-соединений к origin
proxy_buffering on	Для сегментов; включает буферизацию ответа в nginx
proxy_buffering off	Для раздела /; отключает буферизацию (raw streaming)

6.3. Расчёт max_size кеша сегментов

Ориентировочное значение: $\text{bitrate} \times \text{timeShiftBufferDepth} \times \text{distinct_streams} \times 2$

Пример: 10 потоков $\times$ 8 Мбит/с $\times$ 40 с $\times$ 2 $\approx$ 800 МБ. Рекомендуется задать запас 10x для учёта вариативности битрейта.

6.4. TLS-терминация

Сервер Perfect Streamer принимает соединения на портах HTTP и HTTPS. При TLS-терминации на nginx upstream использует порт HTTP. Пересылка заголовков X-Forwarded-Proto и X-Forwarded-Host обязательна для корректного формирования абсолютных URL при absPath=1.

```
server {
    listen 443 ssl http2;
    server_name stream.example.com;

    ssl_certificate      /etc/letsencrypt/live/stream.example.com/fullchain.pem;
    ssl_certificate_key  /etc/letsencrypt/live/stream.example.com/privkey.pem;
    ssl_protocols        TLSv1.2 TLSv1.3;
    ssl_session_cache    shared:SSL:10m;
    ssl_session_timeout  1d;

    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;

    location ... {
        proxy_pass          http://pss_backend;
        proxy_set_header    X-Forwarded-Proto https;
    }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    proxy_set_header    X-Forwarded-Host  $host;
    proxy_set_header    Host              $host;
    # + caching directives from 6.1
}
}

server {
    listen 80;
    server_name stream.example.com;
    return 301 https://$host$request_uri;
}

```

При HTTPS между nginx и origin применяются директивы `proxy_ssl_verify`, `proxy_ssl_trusted_certificate`. Для loopback-соединений шифрование избыточно.

6.5. Мультихост

При обслуживании нескольких `server_name` из одного nginx-процесса `$host` добавляется в `cache key` для изоляции контента:

```

map $uri $pss_cache_key {
    "~^/h[0-9a-f]{16}(?<tail>(/[0-9]+)?/[0-9a-f]{16}\.(ts|m4s)|/sub/[0-9]+/[0-9a-f]{16}\.vtt)$" "$host:stream:$tail";
    default "$host:$uri";
}

```

Размер `keys_zone` рассчитывается как 8000 ключей/MB. Для мультихост-инсталляций с тысячами потоков рекомендуется `keys_zone=...:300m` или выше.

7. Клиентское кеширование

Cache-Control: `immutable` обрабатывается современными браузерами. Клиентский кеш возвращает сегмент без условного запроса при повторном обращении (в том числе при обратном seek в пределах буфера плеера).

Service Workers могут применять стратегию `cache - first` на основании содержимого `Cache-Control`. DASH-плееры используют MSE через `SourceBuffer`; сегмент, помещённый в буфер, остаётся доступен без повторного HTTP-запроса до выхода за границу скольжения.

Для кросс-доменных запросов сервер отдаёт `Access-Control-Allow-Origin: *` вместе с заголовком `Vary: Origin, Access-Control-Request-Headers`. Кеширующий прокси, уважающий `Vary`, делит записи на варианты по значению `Origin` клиента (браузерные плееры шлют `Origin`, консольные — нет), что снижает HIT rate. Поскольку ACAO — универсальный «*», в `location` сегментов безопасно добавить `proxy_ignore_headers Vary` (см. 6.1).

8. Размещение через CDN

Perfect Streamer совместим с CDN в режиме pull-from-origin.

Origin shield. Рекомендуется размещение одного или нескольких shield-узлов между CDN edge и origin для снижения частоты запросов на origin при глобальном распределении клиентов.

Purge. Content-addressed сегменты не требуют purge. При изменении метаданных потока (кодек, разрешение) манифесты обновляются в течение max-age=1 без явного purge.

Cache warming. При ожидаемом росте нагрузки на конкретный поток допустим прогрев CDN с нескольких географических точек до начала трансляции.

Геораспределение. Сегменты (max-age=60) хорошо подходят для географически распределённого кеширования. Манифесты (max-age=1) допускают задержку доставки до одной секунды — приемлемо для live-вещания без низкой задержки.

9. Мониторинг

9.1. X-Cache-Status

Добавление `add_header X-Cache-Status $upstream_cache_status;` в каждый location с кешированием. Значения:

Значение	Описание
HIT	Ответ из кеша
MISS	В кеше отсутствовал, получен от origin и сохранён
EXPIRED	Просрочен, обновлён
UPDATING	Отдана stale-копия параллельному запросу во время обновления
STALE	use_stale вернул просроченную копию (origin недоступен)
REVALIDATED	Origin вернул 304 Not Modified
BYPASS	Сработал proxy_cache_bypass

9.2. Формат access-log

```
log_format pss_cache '$remote_addr $status $request_method "$request" '
                    '$body_bytes_sent rt=$request_time ut=$upstream_response_time '
                    'cache=$upstream_cache_status key=$pss_cache_key';

server {
    access_log /var/log/nginx/pss.log pss_cache;
}
```

9.3. Метрики

Модуль `nginx-vts` экспортирует метрики per-zone в формате Prometheus:

```
GET /status/format/prometheus
```

Рекомендуемые пороги для алертов:

Метрика	Порог	Возможная причина
Segment HIT rate	< 90% за 5 минут	Нарушена нормализация cache key; малый max_size
Manifest MISS rate	> 50% за 1 минуту	proxy_cache_lock не сериализует запросы
Upstream response time p95	> 500 мс за 1 минуту	Перегрузка origin
Cache zone fill	> 90% за 10 минут	Приближение к max_size, планируется LRU-eviction

Качество доставки на стороне узла видно по метрикам сессий на экране «Клиенты» ([Клиенты](#)).

10. Диагностика

Симптом	Вероятная причина	Решение
Segment HIT rate низкий	Vary: Origin делит кеш на варианты; нарушение нормализации в map	Добавить proxy_ignore_headers Vary в location сегментов; проверить regex в директиве map
404 на сегментах после выхода из окна	Закешированный 404 при сегменте, выпавшем из sliding window	Добавить proxy_cache_valid 404 0s в location segments
Задержка playback start 2–5 с	proxy_cache_lock_timeout превышает target latency	Снизить до 1–2 с; включить proxy_cache_use_stale updating
Manifest не обновляется	задан proxy_ignore_headers Cache-Control, и действует proxy_cache_valid с большим TTL	Явно указать proxy_cache_valid 200 1s либо убрать игнорирование заголовков
Рост TIME_WAIT на upstream	Отсутствует keepalive в upstream-блоке	Добавить keepalive 64, proxy_http_version 1.1, proxy_set_header Connection ""
403 на /dash/.../<segment>.m4s от консольных клиентов	Клиент резолвит относительные URL от pre-redirect URL	Сервер выдаёт <BaseURL> с абсолютным путём; совместимо в текущем билде
Лаги, частые ребуферы у удалённых клиентов	Низкий эффективный throughput TCP из-за slow start и idle restart при больших RTT (300 мс и выше)	Тюнинг сетевого стека Linux на origin: см. 10.1

10.1. TCP-тюнинг origin для high-RTT клиентов

Проблема проявляется у клиентов с большим RTT до origin (например, 300 мс и выше) при битрейте потока, близком к пропускной способности канала. Симптомы в плеере — частые ребуферы и разрывы; на сервере клиент при этом выглядит нормально, ошибок нет.

Причина:

- **TCP slow start.** Каждое новое TCP-соединение начинает с congestion window около 14 КБ и наращивает его за несколько RTT. При RTT 300 мс выход на полное окно занимает 2–3 секунды. За это время HLS/DASH-сегмент длительностью 5 с (4–6 МБ) скачивается заметно медленнее реального времени.
- **TCP idle restart.** Между запросами сегментов клиент по HLS pull-модели делает паузу 4–5 с. По умолчанию ядро Linux после такой паузы сбрасывает congestion window соединения обратно к initial cwnd (поведение `net.ipv4.tcp_slow_start_after_idle=1`). В результате при следующем GET keep-alive-соединение начинает передачу с медленного старта заново — даже на уже разогретой сессии.

Дополнительно ситуацию усугубляет то, что congestion control CUBIC по умолчанию плохо переносит длинные RTT и пакетные потери на промежуточных участках сети.

Решение — два sysctl-параметра на origin:

```
# Keep congestion window across idle pauses inside keep-alive sessions.
sysctl -w net.ipv4.tcp_slow_start_after_idle=0

# Use BBR instead of CUBIC: better behaviour on long-RTT paths
# with mild packet loss; paces sending instead of bursting.
modprobe tcp_bbr
sysctl -w net.ipv4.tcp_congestion_control=bbr
```

Для постоянного применения:

```
cat > /etc/sysctl.d/99-pss-net.conf <<EOF
net.ipv4.tcp_slow_start_after_idle = 0
net.ipv4.tcp_congestion_control    = bbr
EOF
sysctl --system
```

Главный эффект даёт первый параметр (`tcp_slow_start_after_idle=0`) — он напрямую устраняет повторный slow start между сегментными запросами в одном keep-alive-соединении. Второй (BBR) даёт дополнительную устойчивость и применяется ко всем новым соединениям. Тюнинг не требует перезапуска Perfect Streamer. К клиентам HTTP/3 (QUIC) эти параметры не относятся: QUIC работает поверх UDP с собственным управлением перегрузкой.

11. Безопасность

11.1. Session URL

URL формата `/h<sess>/...` выполняет функцию сессионного токена — не требует повторной аутентификации. Срок жизни ограничен таймаутом неактивности 60 секунд; при отсутствии активности сессия удаляется автоматически.

Требования:

- HTTPS для всех OTT-путей (`/hls/`, `/dash/`, `/h<sess>/`) в production;

- Session ID в заголовке Location ответа 302 не кешируется (no-cache, no-store).

11.2. Rate limiting

```
limit_req_zone $binary_remote_addr zone=dash_top:10m rate=5r/s;
limit_req_zone $binary_remote_addr zone=hls_top:10m rate=5r/s;
limit_req_zone $binary_remote_addr zone=llhls_top:10m rate=5r/s;

server {
    location /dash/ {
        limit_req zone=dash_top burst=20 nodelay;
        proxy_pass http://pss_backend;
    }
    location /hls/ {
        limit_req zone=hls_top burst=20 nodelay;
        proxy_pass http://pss_backend;
    }
    location /llhls/ {
        limit_req zone=llhls_top burst=20 nodelay;
        proxy_pass http://pss_backend;
    }
}
```

Session URL (/h<sess>/) не требует rate limiting — обработка дешёвая, ответы кешируются.

11.3. Кеширование ответов с ошибками

```
proxy_cache_valid 200 60s;
proxy_cache_valid 301 302 0s;
proxy_cache_valid 404 403 0s;
proxy_cache_valid any 1s;
```

Запрещает кеширование редиректов (уникальный sess в Location) и ответов с ошибками авторизации или отсутствия ресурса.

11.4. Ограничение сетевого доступа к origin

Порт потокового HTTP-сервера (по умолчанию 43972; 43982 для HTTPS) должен быть закрыт для внешнего трафика. Допустимые конфигурации:

1. Bind Perfect Streamer на 127.0.0.1 (при локальном nginx)
2. Firewall-правило:

```
iptables -A INPUT -p tcp -m multiport --dports 43972,43982 ! -s 10.0.0.0/8 -j DROP
```

12. Интеграция с middleware

12.1. Модель prefix-login

Perfect Streamer поддерживает делегирование идентификации пользователя middleware-системе через механизм prefix-login — встроенную альтернативу полной интеграции с внешним биллингом (учётные записи, проверяемые внешней системой, настраиваются на экране «Пользователи / Логины», вкладка «Внешние (биллинг)», [Пользователи / Логины](#)).

У локального логина включается признак префикса, после чего сервер принимает URL с логином вида <prefix><идентификатор_подписчика>:

```
/dash/test1/sub42/xxx/index.mpd
/hls/test1/sub43/xxx/index.m3u8
```

Здесь sub — настроенный логин-префикс, а 42/43 — идентификаторы подписчиков middleware; пароль общий.

12.2. Статистика по подписчикам

В списке клиентов каждая сессия несёт и конфигурационный логин, и фактический URL-логин подписчика (match-login), по которому middleware сопоставляет сессии со своими абонентами. Данные видны на экране «Клиенты» ([Клиенты](#)).

12.3. Ограничения prefix-login

- **Общий пароль.** Все подписчики prefix-пула используют одно значение пароля. Компрометация пароля предоставляет доступ к любому <prefix><строка>.
- **Гранулярность доступа.** Список разрешённых потоков применяется ко всему prefix-пулу. Доступ на уровне отдельного подписчика обеспечивает интеграция с внешним биллингом.
- **Ротация пароля.** Изменение пароля отключает всех активных подписчиков. Для постепенной замены требуется временное использование двух prefix-логинов.

1.9.2 Подключение внешних систем мониторинга

pss-metrics — универсальный экспортёр метрик для Perfect Streamer

Один CLI-скрипт на Python 3, который забирает статистику из HTTP API веб-сервера PSS и формирует вывод для самых распространённых систем мониторинга:

- Zabbix (UserParameter, Low-Level Discovery, zabbix_sender trapper)
- Prometheus (текстовый формат экспозиции для textfile collector)
- InfluxDB / Telegraf (line protocol либо JSON для exec input)
- Универсальный JSON для произвольных скриптов и Nagios-style проверок состояния

Экспортёр — единый самодостаточный файл без сторонних зависимостей, использует только стандартную библиотеку Python 3.6+ (*urllib*, *xml.etree*, *json*, *argparse*).

Файлы

pss-metrics.py	main CLI (executable)
userparameter_pss.conf.example	UserParameter template for Zabbix

Установка

Экспортёр поставляется в `/opt/pss/monitoring/pss-metrics.py`. Убедитесь, что установлен Python 3.6 или новее:

```
# RHEL / Rocky / AlmaLinux
yum install -y python3

# Debian / Ubuntu
apt-get install -y python3
```

Дополнительные пакеты не требуются.

Конфигурация

По умолчанию `pss-metrics` подключается к узлу по петлевому адресу, определяя порт веб-сервера из `/opt/pss/config/pss.json` (либо `/opt/pss/config/pss_default.json`); если порт не удалось прочитать ни из одного файла, используется `http://127.0.0.1:43971`. Параметры можно переопределить через переменные окружения, файл `/etc/pss-metrics.conf` (формат `ключ=значение`) или флаги командной строки. Приоритет: CLI > env > файл > значения по умолчанию. Исключение — `PSS_URL`: из файла он может быть переопределён автоопределением порта; задавайте URL через переменную окружения или флаг командной строки.

Поддерживаемые переменные:

PSS_URL	full URL, e.g. <code>http://10.0.0.1:8808</code>	(auto by default)
PSS_USER	web server login (if authorization is enabled)	
PSS_PASS	web server password	
PSS_TIMEOUT	HTTP timeout, in seconds	(default 5)
PSS_CACHE_DIR	cache directory	(default <code>/run/pss- ↪metrics</code>)
PSS_CACHE_TTL	cache TTL, in seconds	(default 10)
PSS_CA_BUNDLE	path to CA bundle for HTTPS	
PSS_INSECURE	1 – disable TLS certificate verification	
PSS_VERBOSE	1 – log requests to stderr	

Кеш: каждый запуск делает не более одного HTTP GET на конечную точку в окне TTL. При TTL=10 с даже сотни проверок `UserParameter` в минуту дают ~6 HTTP-запросов в минуту на каждую конечную точку API.

Быстрый старт

Проверка состояния (Nagios-style exit codes):

```
pss-metrics.py health
# OK: total=42 running=15 stopped=27 unhealthy=0 version=<версия>
```

Zabbix LLD-обнаружение:

```
pss-metrics.py discover streams
pss-metrics.py discover inputs --running-only
pss-metrics.py discover outputs
```

Получение одной метрики (использовать в Zabbix UserParameter):

```
pss-metrics.py get summary.running
pss-metrics.py get stream.10031.bitrate
pss-metrics.py get input.10031.1.speed1
pss-metrics.py get output.10031.1.speed
pss-metrics.py get sysmon.cpu.self-usage
pss-metrics.py get server.server-version
```

Полный экспорт:

```
pss-metrics.py dump --format=json
pss-metrics.py dump --format=prometheus
pss-metrics.py dump --format=influx
pss-metrics.py dump --format=zabbix-trapper --zabbix-host=streamer-01
```

Пути метрик

pss-metrics get принимает путь, разделённый точками. Пустой вывод означает «значение отсутствует» (например, метрика существует только для запущенных потоков).

server.<attr>	e.g. server.server-version, server.uptime
summary.<key>	total running stopped unhealthy
	input_bitrate_kbps output_bitrate_kbps
sysmon.cpu.<attr>	self-usage total-usage cores
sysmon.memory.<attr>	self-usage-kb available-kb total-kb
sysmon.netbw.<iface>.<attr>	rx-bw tx-bw (interface name as in XML)
stream.<id>.<attr>	any <stream> attribute
input.<sid>.<iid>.<attr>	any <input> attribute
output.<sid>.<oid>.<attr>	any <output> attribute

Полезные атрибуты по потокам (из /data/stream/detail):

```
stream: state, state-str, bitrate, thread-usage, mpts
input:  speed1, recv-bytes, recv-packets, recv-err,
        stat-disc, stat-disc1, stat-scrambled, stat-scrambled1,
        health-state-good, health-status, check-status
output: speed, sent-bytes, sent-packets, sent-err, uri, type
```

Интеграция с Zabbix

Поддерживаются два сценария – выберите подходящий вашему окружению.

1) Статический UserParameter + LLD (Zabbix agent v1 / v2)

Скопируйте `userparameter_pss.conf.example` в `/etc/zabbix/zabbix_agentd.d/pss.conf`, перезапустите `zabbix-agent`, импортируйте на сервере шаблон с LLD-прототипами, использующими ключи `pss.discover`. Пример привязок:

```
UserParameter=pss.discover[*],/opt/pss/monitoring/pss-metrics.py discover $1
UserParameter=pss.get[*],/opt/pss/monitoring/pss-metrics.py get $1
UserParameter=pss.health,/opt/pss/monitoring/pss-metrics.py health
```

На Zabbix-сервере:

```
Discovery rule key:      pss.discover[streams]
Item prototype keys:    pss.get[input.{#STREAM_ID}.1.speed1]
                       pss.get[stream.{#STREAM_ID}.bitrate]
                       pss.get[summary.unhealthy]
```

2) Trapper (push) через zabbix_sender

Запускайте по таймеру (cron / systemd) и направляйте вывод в pipe:

```
/opt/pss/monitoring/pss-metrics.py dump --format=zabbix-trapper \
  --zabbix-host="$(hostname)" \
  | zabbix_sender -z zabbix.example.com -i -
```

Интеграция с Prometheus

Два варианта.

а) Textfile collector (рекомендуется для one-shot окружений).

Запускайте периодический экспорт через systemd-timer или cron:

```
*/1 * * * * /opt/pss/monitoring/pss-metrics.py dump --format=prometheus \
  > /var/lib/node_exporter/textfile_collector/pss.prom.$$ \
  && mv /var/lib/node_exporter/textfile_collector/pss.prom.$$ \
    /var/lib/node_exporter/textfile_collector/pss.prom
```

`node_exporter` отдаст файл через `-collector.textfile.directory`.

б) Прямой scrape через маленькую обёртку (например, `socat + pss-metrics dump`) либо любой сторонний HTTP-прокси на ваш выбор.

Интеграция с Telegraf / InfluxDB

Telegraf *inputs.exec*:

```
[[inputs.exec]]
  commands = ["/opt/pss/monitoring/pss-metrics.py dump --format=influx"]
  interval = "10s"
  timeout = "5s"
  data_format = "influx"
```

Для JSON-парсера используйте *-format=json* и настройте *data_format = «json»* с указанием путей к полям.

HTTPS и аутентификация

Экспортёр рассчитан на запуск на самом узле: запросы с локального адреса веб-сервер PSS принимает без авторизации. Переменные *PSS_USER/PSS_PASS* передаются как HTTP Basic — они пригодятся, когда доступ к API опубликован через промежуточный прокси с Basic-авторизацией; собственную Digest-авторизацию веб-сервера PSS экспортёр не проходит.

Доступ по HTTPS:

```
PSS_URL=https://streamer.example.com:43981 pss-metrics.py health
```

Самоподписанные сертификаты: установите *PSS_INSECURE=1* (не рекомендуется) либо укажите *PSS_CA_BUNDLE=/path/to/ca.pem*.

Коды возврата

pss-metrics следует Nagios-конвенции:

```
0 OK
1 WARNING      (e.g. running streams report unhealthy)
2 CRITICAL     (PSS is unreachable)
3 UNKNOWN      (invalid arguments / internal error)
```

get выводит пустую строку и завершается с кодом 0, если запрошенная сущность отсутствует — это соответствует ожиданию Zabbix, что пустое значение трактуется как «NOT_SUPPORTED», а не как сбой агента.

Диагностика

```
pss-metrics.py -v health      # log every HTTP request to stderr
pss-metrics.py --cache-ttl=0 ... # bypass cache while debugging
rm -rf /run/pss-metrics      # purge cache
```

1.9.3 Оптимизация работы узла

Рекомендации для инсталляций с большим числом потоков, когда возникает дефицит CPU или памяти.

Нагрузка на CPU

- Включайте фильтрацию и модификацию MPEG-TS ([Фильтрация MPEG-TS](#)) только там, где они действительно нужны: по умолчанию все фильтры выключены, каждый включённый добавляет обработку в горячем пути потока. То же относится к углублённым анализам ([Углублённые анализы](#)).
- Мозаика декодирует кадры всех потоков. Её можно отключить целиком в настройках сервера, выборочно у отдельных потоков («Генерировать мозаику») либо реже обновлять, увеличив «Интервал проверки (с)». Учтите: интервал проверки общий — он же управляет повторной проверкой источников и возвратом на основной вход ([Резервирование источников](#)), поэтому его увеличение замедляет и реакцию резервирования.
- Текущая загрузка по потокам видна в окне статистики (секция «Профайлер») и на экране «Системный монитор» ([Системный монитор](#)).

Память

Perfect Streamer периодически возвращает неиспользуемую память операционной системе, а в поставляемом systemd-юните по умолчанию ограничено число параллельных пулов выделения памяти (переменная окружения MALLOC_ARENA_MAX). Это сдерживает постепенный рост резидентной памяти при работе с десятками потоков; сам по себе рост не является следствием утечки. Менять значение вручную обычно не требуется.

Базы данных статистики и EPG

Ошибки переполнения очереди базы данных статистики или базы EPG возникают при недостатке быстродействия диска: базы размещены на медленном носителе либо система перегружена.

Расположение баз задаётся параметром data-dir файла конфигурации pss.properties (по умолчанию — каталог конфигурации). Возможные решения:

1. Перенести базы на быстрый носитель (SSD) параметром data-dir. Вариант с /tmp использует память системы: требует оценки свободной памяти и настройки срока хранения статистики; при перезагрузке данные теряются.
2. Уменьшить детализацию статистики — параметр dbstat-detail (интервал в секундах, по умолчанию 5; допустимо до 30).

1.9.4 Транскодер Intel VPL: поддерживаемое оборудование

Какое оборудование Intel подходит для аппаратного транскодирования, какие кодеки доступны на разных поколениях графики и как убедиться, что оборудование распознано. Установка пакетов транскодера описана в разделе [Транскодеры](#), настройка самого транскодера — в [Транскодеры](#).

Кратко: нужна встроенная или дискретная графика Intel уровня Gen9 и выше. Транскодер определяет, какой рантайм Intel есть на машине, и подключается к нему; один и тот же исполняемый файл `tcivpl` работает и на новом, и на старом оборудовании.

Два рантайма Intel

Рантайм	Оборудование
oneVPL (VPL v2)	Gen12 и новее: Tiger Lake, Rocket Lake, Alder Lake (в том числе Alder Lake-N: N100, N305), Raptor Lake, Arc
Media SDK v1 (MSDK v1)	Gen9–Gen11: Skylake, Kaby Lake, Coffee Lake, Comet Lake, Ice Lake

Выбор рантайма автоматический и отдельной настройки не требует: диспетчер VPL перечисляет реализации, найденные для выбранного устройства `/dev/dri/renderD<N>`, а транскодер берёт первую пригодную для аппаратной работы. Отдельного правила «предпочитать VPL v2» в транскодере нет — для Gen9–Gen11 нового рантайма попросту не существует, поэтому там всегда оказывается Media SDK v1, а на Gen12 и новее обычно установлен только oneVPL.

Учтите при подборе оборудования: процедуры установки транскодера ([Транскодеры](#)) ставят только рантайм VPL v2, то есть рассчитаны на Gen12 и новее. Для Gen9–Gen11 рантайм Media SDK v1 устанавливается отдельно, и в новых дистрибутивах его может не быть (см. ниже).

Кодеки

Рантайм	Декодирование	Кодирование
VPL v2 (Gen12+)	H.264, HEVC, MPEG-2	H.264, HEVC, MPEG-2
MSDK v1 (Gen9–Gen11)	H.264, MPEG-2	H.264, MPEG-2

Чего нет:

- **HEVC на MSDK v1 не поддерживается** — ни декодирование, ни кодирование: в Media SDK v1 HEVC подключается через плагинный интерфейс, которого диспетчер oneVPL не предоставляет. Для HEVC нужна машина Gen12 и новее. При попытке транскодер пишет явную ошибку — `HEVC decode is not supported on the Intel Media SDK v1 runtime` либо `HEVC encode is not supported on the Intel Media SDK v1 runtime`.
- AV1, VP9 и VC-1 не поддерживаются ни одним из рантаймов — в вещании они не нужны.

Три перечисленных кодека — это максимум, с которым умеет работать транскодер. На VPL v2 он не накладывает собственных ограничений, поэтому реальный набор кодеков конкретного адаптера определяет рантайм; что доступно на машине, покажет `tcivpl --hw-list` (см. ниже). Общая, независимая от бэкенда матрица кодеков транскодера приведена в [Бэкенды и кодеки](#).

Урезанное оборудование Atom: не поддерживается

Платформы на ядрах Atom до Gen12 — Apollo Lake, Gemini Lake, Elkhart Lake, Jasper Lake (Celeron и Pentium серий J и N: J4125, J6412, N5105 и подобные) — для транскодирования не подходят.

В них урезанный медиадвижок: полноценного аппаратного кодирования нет, а кодирования MPEG-2 нет вообще. Декодирование на таких чипах может работать, но транскодировать нечем — используйте программный транскодер (пакет **pstreamer-tcs**) или другое оборудование. Транскодер такое оборудование не отбраковывает заранее: он запускается и завершается ошибкой при инициализации кодера.

Важное исключение: Alder Lake-N (N100, N305 и подобные) — тоже Atom-ядра, но графика у них полноценная Gen12, они поддерживаются.

Отдельно: у старых процессоров Atom нет AVX, а часть готовых пакетов Intel собрана с AVX. В этом случае процесс транскодера падает с `Illegal instruction` — инструкции выполняет библиотека драйвера или рантайма Intel, загруженная в процесс, а не код самого транскодера.

Что должно быть установлено

Компонент	RHEL / AlmaLinux / Rocky	Debian / Ubuntu
Драйвер VA-API iHD (обязателен)	intel-media-driver	intel-media-va-driver-non-free
Рантайм VPL v2 (Gen12+)	intel-vpl-gpu-rt	libmfxgen1
Рантайм MSDK v1 (Gen9–Gen11)	intel-mediasdk	libmfx1
VA-API, DRM	libva, libdrm	libva2, libva-drm2, libdrm2

Рантайм нужен ровно один — тот, что соответствует оборудованию. Драйвер iHD обязателен в любом случае: на старом i965 инициализация VA-API пройдёт, но декодер не поднимется, причём отдельного сообщения об этом нет — драйвер виден только в журнале при старте потока. В Debian и Ubuntu берите именно вариант `intel-media-va-driver-non-free`: в `intel-media-va-driver` часть кодеков вырезана.

Имена пакетов зависят от того, откуда они ставятся. Приведённые Debian-имена — из репозитория Intel Graphics, который подключается по процедуре установки ([Ubuntu 22/24](#)); в собственном репозитории Ubuntu тот же рантайм VPL v2 называется `libmfx-gen1.2` и версии там заметно старше.

Диспетчер Intel VPL (в дистрибутивах — `libvpl2`) поставляется в составе пакета транскодера: он устанавливается в `/opt/pss/lib/libvpl.so.2`, путь прописывается в каталоге `/etc/ld.so.conf.d/`. Но диспетчер только выбирает рантайм — если подходящего рантайма на машине нет, транскодировать нечем.

Рантайм MSDK v1 из новых дистрибутивов убран: `intel-mediasdk` есть в EPEL 9, но не в EPEL 10; `libmfx1` есть в Debian 12 и Ubuntu 24.04, но его уже нет в следующих выпусках. На таких системах для оборудования Gen9–Gen11 рантайм придётся устанавливать отдельно.

Пакет транскодера ни драйвер, ни рантайм за собой не тянет, поэтому успешная установка сама по себе не означает, что аппаратное транскодирование заработает — это проверяется отдельно. Учтите также, что пользователь службы должен иметь доступ к устройствам `/dev/dri` ([Ubuntu 22/24](#)).

Как проверить

```
/usr/local/bin/tcivpl --hw-list
```

Команда печатает XML со списком реализаций, которые нашёл диспетчер VPL — одной строкой, без переносов. Для каждой указаны имя рантайма (`device-name`), версия API, списки кодеков декодера и кодера, адаптер и признак пригодности к аппаратной работе. Пример вывода на Gen9–Gen11 (показан с переносами для читаемости):

```
<vpl mfx-version="2.15">
  <device id="0" device-name="mfxhw64" api-version="1.35" legacy="1"
    dec-codecs="h264,mpg2" enc-codecs="h264,mpg2"
    vendor-id="8086" device-id="4555" drm-render="128"
    supported="1"/>
</vpl>
```

Как читать вывод:

- `legacy="1"` — реализация сообщает версию API ниже 2.0, то есть это старый рантайм Media SDK v1; без этого признака — oneVPL 2.x. На это и на `api-version` следует опираться в первую очередь: значение `device-name` приходит от диспетчера как есть (на практике — `mfxhw64` для Media SDK v1 и `mfx-gen` для oneVPL).
- `supported="1"` — реализация пригодна для аппаратной работы. Запись без этого признака равноценна её отсутствию.
- Для Media SDK v1 список кодеков всегда `h264,mpg2` — это набор, поддерживаемый транскодером на этом рантайме, а не результат опроса оборудования. Для oneVPL списки берутся из того, что сообщает рантайм.
- `mfx-version` — версия API заголовков, с которыми собран транскодер, а не версия установленного рантайма.

Пустой список (элемент `<vpl>` без единого `<device>`) означает, что аппаратного транскодирования на этой машине не будет: не установлен рантайм или драйвер, GPU не Intel либо нет доступа к `/dev/dri`. Сообщение `No vpl libraries found` — отдельный случай: не найдена сама библиотека-диспетчер.

Команда только перечисляет реализации и ничего не выбирает. Какая из них будет использована, определяется при запуске потока — по устройству, заданному в настройках транскодера. При старте потока транскодер пишет в журнал две строки: выбранный рантайм с версией API и версию VA-API вместе с именем драйвера. По второй строке видно, что работает именно драйвер iHD, а не i965.

Обнаруженные бэкэнды транскодирования, их версии и найденные устройства показаны также на экране «О программе» (*О системе*), а загрузка GPU — на экране «Системный монитор» (*Системный монитор*).

1.10 FAQ

Ответы на вопросы, которые чаще других возникают при эксплуатации: совместимость SRT со сторонним ПО и приём мультикаста на высоких битрейтах.

1.10.1 SRT: авторизация по логину и паролю в стороннем ПО

Между двумя узлами Perfect Streamer авторизация SRT настраивается штатно, полями входа и выхода ([SRT](#)). Работа с другим ПО не гарантируется: параметр stream ID не стандартизован и в разном ПО реализован по-разному.

Механизм совместимости общий: принимающая сторона берёт stream ID подключающегося клиента и ищет по нему учётную запись в списке локальных логинов ([Пользователь: добавление и изменение](#)). Совпасть должно поле «Логин» — целиком со значением stream ID. Например, при ссылке

```
srt://Stream_IP:port?streamid=!#:u=1234567890,password=1234567890
```

в поле «Логин» вводится

```
!#:u=1234567890,password=1234567890
```

Синтаксис stream ID для Perfect Streamer значения не имеет: строка не разбирается по частям, а сравнивается целиком. Исключений три — символы, которые узел трактует сам:

- | — разделитель: часть строки до первого такого символа считается логином, остальное паролем учётной записи;
- * — stream ID не задан; клиент авторизуется по адресу, то есть учётной записью с признаком «Учётная запись по IP-адресу»;
- @ в начале строки зарезервирован за служебной авторизацией между узлами домена Meshwork.

Длину stream ID Perfect Streamer не ограничивает — ограничение приходит от библиотеки SRT и составляет 512 байт вместе с завершающим нулём. Значения длиннее примерно 500 символов передавать не следует: слишком длинная строка либо обрывает соединение, либо уходит пустой без сообщения об ошибке.

Разное ПО формирует stream ID по-своему, поэтому, если приём по ссылке не работает, включите на выходе SRT дополнительную настройку «Trace». В журнале потока, который принимает подключение, появятся адрес клиента, полученный stream ID и причина отказа — по этой строке видно, что сторона прислала на самом деле. Дальше stream ID можно скорректировать: например, убрать лишние символы в начале строки, мешающие передаче значения.

1.10.2 Рекомендации по работе с UDP-мультикастом

Задача. Стабильно принять UDP-мультикаст с суммарным битрейтом в несколько сотен Мбит/с (1 Гбит/с и более) на одном сервере.

Проблема. При приёме на сетевые карты с интерфейсом RJ-45 с ростом трафика выше нескольких сотен мегабит начинаются нарастающие потери пакетов мультикаста. Тонкая настройка карты не помогает — в современных версиях операционных систем уже используются оптимальные значения. Карты на лучших чипах проблему не снимают, особенно после 500 Мбит/с. Не помогает и бондинг двух карт: одна мультикаст-группа всегда приходит на один физический канал и на одну очередь приёма, поэтому агрегируется суммарная полоса, а не запас по отдельному потоку.

Решение. По опыту эксплуатации для приёма и передачи UDP-мультикаста следует использовать 10-гигабитные сетевые карты с интерфейсом SFP+. Достаточно бюджетной карты уровня Intel X520-DA1/2 (Intel 82599ES) — в наших тестах она устраняет потери пакетов при трафике мультикаста свыше 1 Гбит/с. Дополнительно заметно снижается нагрузка на CPU: обработка прерываний и очередей приёма на таких картах обходится дешевле.

Учтите при диагностике: счётчик ошибок приёма `recv-err` в статистике входа считает только ошибки чтения сокета и рассинхронизацию по границе TS-пакета. Потери, вызванные переполнением приёмного буфера ядра, до приложения не доходят и в этот счётчик не попадают — их видно командой `netstat -su` (строка `RcvbufErrors`) и по ошибкам непрерывности в анализаторе потока ([Анализатор](#)).

1.10.3 Рекомендации по настройке сети для мультикаста

Раздел относится к сетевым входам UDP / RTP и Pro-MPEG. На приём SRT эти параметры не влияют: библиотека SRT задаёт размер буфера сама, а задержка и запас на ретрансмиссию настраиваются на самом входе ([SRT](#)).

Размер приёмного буфера сокета задаётся дополнительной настройкой входа «Socket buffer (bytes)». По умолчанию она равна нулю — узел ничего не запрашивает, и сокет получает размер буфера, заданный в системе параметром `net.core.rmem_default`. Как только значение задано явно, начинает действовать потолок `net.core.rmem_max`: ядро молча урезает запрос до этого предела, не сообщая об усечении ни в журнал, ни в статистику. Поэтому потолок поднимают до того, как задавать размер буфера у входа.

Параметры задаются отдельным файлом в `/etc/sysctl.d/`:

```
cat > /etc/sysctl.d/99-pss-net.conf <<EOF
net.core.rmem_default=8388608
net.core.rmem_max=16777216
EOF
```

Применить:

```
sudo sysctl --system
```

Значение `rmem_max` — только потолок, память по нему не резервируется. Значение `rmem_default` применяется ко всем UDP-сокетах системы; если такой глобальный размер нежелателен, оставьте его без изменений и задайте «Socket buffer (bytes)» только тем входам, которым он действительно нужен. Ориентир для расчёта — битрейт потока, умноженный на допустимую паузу в обслуживании сокета: 8 МБ покрывают около 64 мс при 1 Гбит/с.

1.10.4 Flussonic и SRT

Пример ссылки SRT для приёма на ПО «Flussonic»:

```
srt://Stream_IP:port?streamid=flussonic
```

Где **streamid** — это логин клиента в ПО «Flussonic», который задаётся в разделе «Конфигурация — Настройка пиров».

При добавлении нового клиента достаточно указать только логин, в тестовой ссылке указан «flussonic». ПО «Flussonic» при работе с SRT автоматически генерирует *streamID*, и если не указать в ссылке логин *streamID* на приёме, то поток приниматься не будет, а Perfect Streamer не сможет его отдать.

Аналогичным образом можно задать *streamID* в формате логина и пароля, создав в Perfect Streamer учётную запись, у которой поле «Логин» равно `!#::u=1234567890,password=1234567890`, и прописав во «Flussonic» ссылку:

```
srt://Stream_IP:port?streamid=!#::u=1234567890,password=1234567890
```

Возможно указать *streamID* в формате IP-адреса у Perfect Streamer — для этого у учётной записи включается признак «Учётная запись по IP-адресу», а в поле адреса вводится:

- 192.168.1.1 — единственный IP;
- 192.168.1.1-192.168.1.254 — диапазон IP.

В обоих случаях ссылка для приёма по IP во «Flussonic» будет выглядеть так: `srt://Stream_IP:port?streamid=*`

Поток привязывается к IP-адресу клиента: приём по этой ссылке возможен только с указанного IP-адреса (или из указанного диапазона).

1.11 Инструменты

Perfect Streamer Toolkit — консольные утилиты, дополняющие стример. Они входят в состав пакета **pstreamer** и после установки находятся в каталоге `/opt/pss/tools/`:

- **ts_analyze** — анализатор транспортного потока MPEG-TS с проверкой соответствия TR 101 290 (*TS Analyze Perfect Streamer Toolkit v2.3 — TR 101 290*);
- **mpts_migrate** — миграция идентичности DVB SI/PSI работающего мультимплекса на Perfect Streamer (*MPTS Migrate Perfect Streamer Toolkit v1.1 — миграция идентичности MPTS*).

1.11.1 TS Analyze Perfect Streamer Toolkit v2.3 — TR 101 290

Часть **Perfect Streamer Toolkit** — <https://pstreamer.tv>

Консольный **анализатор транспортного потока MPEG-TS** с проверкой соответствия **ETSI TR 101 290 V1.4.1** и валидацией модели T-STD буферов **ISO/IEC 13818-1**.

Читает **UDP multicast/unicast** или **TS-файлы**, автоматически обнаруживает PCR PID через PAT/PMT и выводит подробный или краткий отчёт на stdout.

Инструмент входит в состав пакета **pstreamer** и после установки находится в `/opt/pss/tools/ts_analyze` — отдельно устанавливать ничего не нужно.

Что проверяется

Анализатор обходит каждый TS-пакет и сообщает о нарушениях:

- **Priority 1** (декодируемость TS): синхронизация TS, потеря синхронизации, наличие и CRC PAT/PMT, счётчик непрерывности, наличие PID
- **Priority 2** (рекомендуемый мониторинг): индикатор ошибок передачи, ошибки CRC, повторение / точность / разрывы PCR, интервал PTS, наличие CAT
- **Priority 3** (расширенный мониторинг): интервалы NIT/SDT/EIT/TDT, неупомянутые PID, переполнение / недополнение T-STD буферов

Дополнительно выдаёт:

- **Точность PCR** до ± 500 нс — регрессия по байтовым позициям
- **Дрейф PCR** в ppm (только live-режим)
- **Модель T-STD буферов** для каждого элементарного потока (live-режим)
- Валидация **размеров SI-секций** относительно лимитов ISO/EN с предупреждениями о совместимости EIT-on-STB (>1024 Б)
- **UDP IAT** (inter-arrival time) — статистика джиттера на уровне дейтаграмм (только live-режим)

Использование

```
ts_analyze [options] <input>
```

Входы

Форма	Описание
udp://239.1.1.1:1234	UDP multicast
udp://eth0@239.1.1.1:1234	UDP multicast на указанном интерфейсе
udp://192.168.1.100:1234	UDP unicast
udp://lo@127.0.0.1:12655	UDP unicast на loopback (тестовый источник)
/path/to/file.ts	Локальный TS-файл

RTP-инкапсуляция UDP определяется и распаковывается автоматически.

Опции

Опция	Описание	По умолчанию
-t, --time <sec>	Длительность анализа в секундах	30
-s, --short	Краткий итоговый отчёт	–
-f, --full	Полный детализированный отчёт	да
-b, --bitrate <Mbps>	Подсказка битрейта TS для файлового режима	38.8
-p, --pcr-pid <pid>	Анализировать только указанный PCR PID (десятичный или 0xNNNN)	авто
-l, --pcr-limit <ms>	Лимит ошибки повторения PCR в мс	40
--no-eit	Пропустить анализ EIT — P3.7..P3.10 показываются как N/A, вклад EIT в P2.2 / итог по размерам секций исключается	EIT включён
--no-nit	Пропустить анализ NIT — P3.1, P3.2 показываются как N/A, вклад NIT в P2.2 / итог по размерам секций исключается	NIT включён
--no-color	Отключить ANSI-цвета на выводе	цвета включены
--xml	Структурированный XML на stdout (без stderr)	текст
-h, --help	Показать справку	–

Примеры

```
# 30-second TR 101 290 check on a multicast stream
ts_analyze -t 30 udp://239.10.10.1:1234

# short summary, save to log (no color)
ts_analyze -s --no-color -t 60 udp://239.10.10.1:1234 > report.txt

# analyze a TS file
ts_analyze -t 30 recording.ts

# analyze only a single PCR PID
ts_analyze -p 0x0100 -t 30 udp://239.10.10.1:1234

# machine-readable XML for CI / monitoring
ts_analyze --xml -t 30 udp://239.10.10.1:1234 > result.xml

# skip EIT/NIT analysis (e.g. for streams where they are intentionally absent)
ts_analyze --no-eit --no-nit -t 30 udp://239.10.10.1:1234
```

Отключение анализа EIT или NIT

В некоторых потоках EIT или NIT намеренно отсутствуют (закрытые сети, лабораторные источники, только OTT-доставка и т.п.). Соответствующие проверки P3 из TR 101 290 будут всегда давать FAIL по итоговому шлюзу OVERALL — это просто шум.

Используйте `--no-eit` и/или `--no-nit`, чтобы исключить эти проверки:

Флаг	Пропускаемые проверки	Побочные эффекты
<code>--no-eit</code>	P3.7 (EIT actual P/F), P3.8 (EIT other P/F), P3.9 (EIT actual schedule), P3.10 (EIT other schedule)	Секции EIT не собираются, поэтому их вклад в P2.2 (CRC) и в сводку размеров SI-секций равен нулю. Предупреждение о совместимости EIT-on-STB подавляется.
<code>--no-nit</code>	P3.1 (NIT actual), P3.2 (NIT other)	Секции NIT не собираются, поэтому их вклад в P2.2 (CRC) и в сводку размеров SI-секций равен нулю.

Пропущенные проверки появляются в отчёте как N/A с пометкой disabled (`--no-eit`) / disabled (`--no-nit`), а в XML — как `applicable="false" result="N/A"`. В кратком отчёте вместо счётчика ошибок отображается NIT=off / EIT=off.

Флаги затрагивают только обработку EIT (PID 0x0012) и NIT (PID 0x0010) — все остальные проверки TR 101 290 (P1.x, P2.x, SDT, TDT, CAT, T-STD, дрейф PCR, IAT) выполняются как обычно.

Режим вывода XML (`--xml`)

`--xml` заставляет анализатор выводить единый самодостаточный UTF-8 XML-документ на **stdout**. Вся служебная информация (баннер, «Stream locked», «PCR PIDs discovered», ежесекундный прогресс, сводка захвата, предупреждения о короткой длительности) подавляется; **stderr остаётся пустым**, если только не произошёл реальный сбой (не удалось открыть вход, нет PCR-данных, поток слишком короткий, прерывание сигналом). ANSI-цвета принудительно отключены.

Код выхода такой же, как в текстовом режиме: 0 при OVERALL=PASS, 65 при OVERALL=FAIL, плюс стандартные коды ошибок / сигналов (1, 2, 3, 130, 143).

Структура XML верхнего уровня:

```
<?xml version="1.0" encoding="UTF-8"?>
<ts_analyze version="2.3">
  <source>udp://239.1.1.1:5000</source>
  <timestamp>2026-04-30T12:00:00+0300</timestamp>
  <duration_s>30.00</duration_s>
  <packets total="..." null="...">
  <ts_bitrate_mbps>20.012</ts_bitrate_mbps>

  <programs>
    <program number="1" pmt_pid="0x0100" pcr_pid="0x0101">
      <es pid="0x0101" stream_type="0x1b" name="H.264/AVC"/>
      <es pid="0x0102" stream_type="0x03" name="MPEG-2 Audio"/>
    </program>
  </programs>

  <tr101290>
    <check id="1.1" name="TS Sync Byte Error" applicable="true" errors="0" result=
    ↪ "PASS"/>
    ...
```

(continues on next page)

(продолжение с предыдущей страницы)

```

<check id="2.3" name="PCR Repetition Error"
    applicable="true" errors="0" result="PASS" soft_violations="3"/>
...
<check id="3.4" name="Unreferenced PIDs" applicable="true" count="2" result="INFO
"/>
...
</tr101290>

<si_section_size oversize_total="0" eit_stb_warn_total="12">
    <table name="EIT_actual_pf" max_bytes="1380" std_limit="4096"
        oversize="0" eit_stb_warn="12" result="WARN_STB"/>
    ...
</si_section_size>

<iat datagrams="33252" intervals="33251" min_ms="0.002" max_ms="5.009"
    avg_ms="0.150" stddev_ms="0.295" p95_ms="0.872" p99_ms="1.076"
    max_jitter_ms="4.272" gap_threshold_ms="100.0" gaps_over_threshold="0"/>

<pcr_pids>
    <pcr_pid value="0x0101" sid="1" pcr_count="1500" discontinuities="0"
        estimated_bitrate_mbps="18.750">
        <interval samples="1499" min_ms="19.812" max_ms="20.195"
            avg_ms="20.001" p95_ms="20.102"
            iso_hard_violations="0" tr_soft_violations="0"
            rec_violations="0" result="PASS"/>
        <accuracy samples="1499" min_ns="-148.3" max_ns="201.7" abs_max_ns="201.7"
            avg_ns="2.1" stddev_ns="45.6" p95_ns="102.3"
            violations="0" result="PASS"/>
        <drift measured="true" ppm="0.618" limit_ppm="30"
            verdict_mode="informational" result="PASS"/>
        <tstd overflows="0" underflows="0" max_fill_bytes="34218" result="PASS">
            <es_buffer es_pid="0x0101" stream_type="0x1b"
                capacity_bytes="3000000" measuring="true"
                es_bitrate_mbps="15.200" max_fill_bytes="34218"
                overflows="0" underflows="0"/>
        </tstd>
    </pcr_pid>
</pcr_pids>

<unreferenced_pids count="2">
    <pid value="0x01ff"/>
    <pid value="0x0200"/>
</unreferenced_pids>

<overall result="PASS"/>
</ts_analyze>

```

Ключевые соглашения:

- Все PID форматируются как 0xHHHH (4-значный hex с префиксом 0x).
- Атрибут `result` принимает значения PASS / FAIL / WARN / N/A / INFO / SKIP / ok / WARN_STB.
- Для неприменимых проверок (файловый режим, отсутствие скремблирования и т.п.) элемент всё равно присутствует с `applicable="false"` и `result="N/A"`, чтобы потребители схемы видели стабильную форму.
- `<drift>` несёт `verdict_mode="informational"` для $30 \text{ s} \leq T < 300 \text{ s}$ и

`verdict_mode="hard"` для $T \geq 300$ s; `result="SKIP"` для запусков короче 30 с.

- `<iat>` отсутствует для запусков в файловом режиме.
- `<overall>` отражает тот же шлюз, что и строка OVERALL в текстовом отчёте, и совпадает с кодом выхода процесса.

Вывод — корректно сформированный (well-formed) XML (валидируется через `xmllint -noout`); подавайте напрямую в XSLT, Python `lxml` и т.п. без доработок парсинга.

Чтение отчёта

Цвета статусов

Статус	Цвет	Значение
PASS / ok	зелёный	Проверка соответствует стандарту
WARN / WARNING / WARN (STB)	жёлтый	Мягкое нарушение, не влияет на OVERALL
FAIL / ERROR	красный	Нарушение стандарта, влияет на OVERALL
INFO / NOTE / N/A	по умолчанию	Только информационно

Используйте `--no-color` при перенаправлении в лог-файлы или в не-ANSI-терминалы.

Минимальная длительность анализа

Длительность	Покрытие	Код выхода
< 2 с	Недостаточно — анализ отклоняется	3 (ошибка)
2–10 с	P1 + P2 надёжны; некоторым проверкам P3 может не хватить данных	0 + WARN
10–30 с	P1 + P2 + большинство P3; TDT (30 с) может не успеть	0 + NOTE
≥ 30 с	Полное покрытие всех проверок TR 101 290	0

Длительность по умолчанию — **30 секунд**, достаточно для полного покрытия TR 101 290. Используйте `-t <sec>` для увеличения (например, для приёмо-сдаточных тестов по дрейфу PCR) или сокращения (быстрые smoke-проверки).

Во время работы анализатор каждую секунду обновляет однострочный индикатор прогресса на `stderr`:

```
Progress: 47.3% (14.2s / 30.0s, 330614 packets)
```

Строка использует carriage-return (`\r`), чтобы оставаться одной строкой в терминале; перенаправьте `stderr` (`2>/dev/null`), чтобы её подавить.

Вердикт по дрейфу PCR — двухуровневое окно

Допуск тактовой частоты PCR по **ISO/IEC 13818-1 §2.4.2.1** и **ETSI TR 101 290** — **±30 ppm**. Значение дрейфа в отчёте получается линейной регрессией кумулятивных секунд PCR относительно времени прихода по часам стенда; статистическая ошибка убывает как $1 / T^{(3/2)}$, поэтому окно анализа должно быть достаточно длинным, чтобы шум измерения был ниже границы ±30 ppm.

Поэтому анализатор ставит вердикт по дрейфу в зависимости от длительности анализа:

Окно	Вердикт при выходе дрейфа за допуск	Влияние на OVERALL
$T < 30 \text{ с}$	skipped (шум доминирует относительно границы ±30 ppm)	нет
$30 \text{ с} \leq T < 300 \text{ с}$	WARN — только информационно	нет
$T \geq 300 \text{ с}$	FAIL	OVERALL = FAIL, exit 65

300 с — приёмочное окно (значение продукта; стандарт **ETSI TR 101 290** интервалы измерений не нормирует) — достаточно длинное, чтобы даже бёрстовый/loopback-канал доставки усреднялся ниже 1 ppm шума измерения, и выход за допуск отражал тактовый генератор кодера, а не сеть. Полный отчёт показывает текущий уровень в строке `Verdict mode` блока PCR DRIFT.

Чтобы получить жёсткий вердикт PASS/FAIL по дрейфу, запускайте с `-t 300` или больше.

Рекомендации по качеству источника (информационно; уровни вердикта не меняют):

Источник	Минимальное окно для ±5 ppm	Для ±2 ppm	Приёмка
Вещательный multicast (CBR-сеть, джиттер < 100 мкс)	30 с	60 с	5 мин
Стабильная IP-сеть (джиттер < 200 мкс)	30 с	2 мин	5–10 мин
Loopback / бёрстовый отправитель (UDP unicast на lo)	5 мин	15 мин	30 мин
Калибровка / лабораторное измерение	—	30 мин	1+ час

Примеры:

```
# Quick PCR drift check on a real broadcast multicast (30 s)
ts_analyze -t 30 -s udp://239.1.1.1:5000

# Reliable check on a loopback source (5 min)
ts_analyze -t 300 -s udp://lo@127.0.0.1:12655

# Lab acceptance (30 min, full report to file)
ts_analyze -t 1800 -f --no-color udp://239.1.1.1:5000 > acceptance.txt
```

Если один и тот же поток анализируется несколькими короткими окнами и значение дрейфа колеблется более чем на несколько ppm между окнами, узким местом является джиттер доставки (темп подачи отправителя или сеть), а не тактовый генератор кодера — увеличьте окно.

Коды выхода

Код	Значение
0	Анализ завершён, OVERALL = PASS
1	Ошибка аргументов или входа
2	В потоке нет данных PCR
3	Длительность потока меньше минимума 2 с, либо данные не получены вовсе (источник не вещает, неверный адрес, не выполнено multicast-подключение, брандмауэр или неверный интерфейс — анализатор печатает подсказку)
65	Анализ завершён, OVERALL = FAIL — нарушение TR 101 290 / ISO 13818-1
130	Прервано SIGINT (Ctrl+C) — анализ отменён, отчёт не выводится
143	Прервано SIGTERM — анализ отменён, отчёт не выводится

65 — это EX_DATAERR из POSIX <sysexits.h>: «входные данные некорректны». Используйте его в CI / мониторинге для шлюзования по соответствию потока:

```
ts_analyze -s -t 60 udp://239.1.1.1:5000 || {
    case $? in
        65) echo "stream does not conform — see report" >&2 ;;
        130) echo "interrupted by user" >&2 ;;
        *) echo "tool error" >&2 ;;
    esac
}
```

Коды 130/143 следуют конвенции POSIX shell 128 + signal_number, поэтому \$? после Ctrl+C совпадает с тем, что bash возвращает для любого процесса, убитого SIGINT/SIGTERM. При прерывании анализатор выводит одну строку на stderr (Analysis interrupted by signal N — no report produced.) и полностью пропускает генерацию отчёта.

Пример вывода

Полный отчёт (фрагмент)

```
=====
MPEG-TS ANALYZER v2.3 — TR 101 290 FULL REPORT
Perfect Streamer Toolkit https://pstreamer.tv
=====

Source      : udp://239.1.1.1:5000
Timestamp   : 2026-04-30 12:00:00
Duration    : 30.00 s
Packets     : 398936 total, 12045 null
TS bitrate  : 20.012 Mbit/s
-----

=====
TR 101 290 — PRIORITY 1 (TS decodability)
=====
| 1.1  TS Sync Byte           : 0      PASS
| 1.4  Continuity Count       : 0      PASS
| 1.6  PID Absence (5s)       : 0      PASS
| ...

```

(continues on next page)

(продолжение с предыдущей страницы)

=====				
TR 101 290 – PRIORITY 2 (recommended monitoring)				
=====				
	2.3	PCR Repetition	:	0 PASS
	2.5	PCR Accuracy	:	0 PASS
...				
=====				
OVERALL COMPLIANCE: PASS – stream is TR 101 290 compliant				
=====				

Краткий отчёт

MPEG-TS Analyzer v2.3 | Perfect Streamer Toolkit https://pstreamer.tv
TR 101 290 Summary | udp://239.1.1.1:5000 | 30.0s | 2026-04-30 12:00:00

P1: sync=0 CC=0 PAT=0 PMT=0 PID=0 P2: TEI=0 CRC=0 PTS=0 P3: NIT=0 SDT=0 EIT=0
TDT=0 unref=2
IAT: dgrams=33252 avg=0.150 ms max=5.009 ms stddev=0.295 ms p99=1.076 ms gaps>100.
000 ms=0

PCR PID SID Count Intv max Jitter max Drift Interval
Accuracy T-STD

0x0101 1 1500 20.195 ms 76.4 ns 0ppm PASS PASS
PASS

OVERALL: PASS

Примечания

- **Файловый режим:** дрейф PCR, модель T-STD буферов и UDP IAT не измеряются — для них требуется референс реального времени. Остальные проверки работают в обоих режимах.
- **Один транспортный поток:** за раз анализируется один MPTS или SPTS.

1.11.2 MPTS Migrate Perfect Streamer Toolkit v1.1 — миграция идентичности MPTS

Часть **Perfect Streamer Toolkit** — <https://pstreamer.tv>

Захватывает идентичность DVB SI/PSI работающего многопрограммного MPEG-TS-потока (MPTS) и воспроизводит её на экземпляре Perfect Streamer (PSS) на том же хосте. Результат: пользовательские приёмники (STB / TV) продолжают работать **без повторного сканирования каналов** после миграции или переключения на резерв.

Инструмент входит в состав пакета **pstreamer** и после установки находится в `/opt/pss/tools/mpts_migrate` — отдельно устанавливать ничего не нужно.

Предусловия

Перед запуском утилиты убедитесь, что:

- **PSS работает** на том же хосте (или на хосте, доступном через `--pss-base`). Утилита ищет `pss` в `/proc` и читает `pss.json`, чтобы получить `admin-порт` (если в конфиге его нет — 43971).
- **Источник MPTS доступен**, если планируется захват (режимы 1, 2, `save+apply`): URL, переданный позиционным аргументом `<input>`, должен отдавать MPEG-TS-поток. Для UDP multicast нужно, чтобы IGMP / firewall разрешали приём; для файлов — путь должен существовать.
- **Целевой MPTS уже настроен в PSS**: утилита не создаёт новые потоки. Объект MPTS и хотя бы столько SPTS-фидеров, сколько сервисов в инвентаре, должны существовать заранее. Сервисы без свободного фидера показываются в диалоге и могут быть пропущены.
- **HTTP admin API открыт на localhost** для чтения `/data/stream` (используется при `verify`) и записи `/config/stream` (`apply`).

Что мигрируется

Все идентификаторы, видимые приёмником, на уровне транспортного потока и сервисов:

- **Транспортный поток**: TSID, ONID, network ID, network name, **provider name** (применяется как `mux-wide sdt-provider-name`, если все исходные сервисы имеют одно и то же значение), дескриптор доставки (параметры наземного / кабельного / спутникового вещания), версии PAT/SDT/NIT
- **На сервис**: `service_id`, `pmt_pid`, `pcr_pid`, `service_type`, имя сервиса, логический номер канала (LCN), флаг `free-CA`, флаги `EIT-present` / `EIT-schedule`
- **Элементарные потоки**: PID-ы (применяется `identity remap` — см. *Ограничения*), типы потоков, языковые теги
- **Условный доступ**: CA-дескрипторы на уровне программы и ES

Имена сервисов / провайдеров в DVB-кодировках, отличных от ASCII (например, ISO-8859-5 для кириллицы), декодируются в UTF-8 автоматически.

Для каждого сервиса ES PID remap строится как `identity`-пары (`mpegts-pid-old` $\equiv$ `mpegts-pid-new`) для каждого PCR / video / audio / teletext / data PID, поэтому в результирующем мультиплексированном выводе исходные PID-ы сохраняются **byte-exact**. Старые приёмники, кеширующие PMT после первого сканирования, продолжают работать без перенастройки.

На PSS 2.0.0.193 и новее план дополнительно включает у `muxer`-входа целевого MPTS штатный режим сохранения исходных PID (выключенный Automatic PID mapping) и задаёт для каждого сервиса захваченный порядковый номер программы в PAT (`program order`) — эфирный порядок программ переживает миграцию. Поддержка определяется автоматически по конфигурации целевого стримера; на более старых версиях эти ключи не отправляются (утилита печатает предупреждение), и единственным механизмом закрепления PID остаются `identity`-пары, а порядок программ в PAT не сохраняется.

Сценарии использования

- **Failover:** переключение декодеров с основного MPTS на резервный с сохранением каналов на стороне приёмника
- **Аппаратная миграция:** переезд работающего мультиплекса с одного PSS-хоста на другой без инструкций для зрителей
- **Снимок до / после обновления:** захват мультиплекса перед апгрейдом PSS, повторное применение после, чтобы гарантировать бит-идентичный SI/PSI
- **Ручная правка и повторное применение:** захват, правка migrate.json (переименование сервисов, смена LCN, корректировка service_type), повторное применение
- **Dry-run-проверка:** вывод каждого HTTP POST, который *был бы* отправлен, без воздействия на PSS

Быстрый старт

```
# Capture from a live stream and apply to local PSS in one run
mpts_migrate udp://239.1.1.1:1234

# Capture, save to migrate.json and apply
mpts_migrate -s udp://239.1.1.1:1234

# Capture only – write to file, do not apply
mpts_migrate -o backup.json udp://239.1.1.1:1234

# Apply a previously saved JSON
mpts_migrate -i backup.json

# No arguments – load ./migrate.json and apply
mpts_migrate

# Preview what apply would do, without changes
mpts_migrate -i backup.json --dry-run
```

Рабочий процесс

1. **Захват** — утилита открывает поток, парсит PAT / PMT / SDT / NIT / EIT в течение -t секунд (по умолчанию 30) и строит инвентарь; измеряется суммарный битрейт потока.
2. **(Опционально) Сохранение** — с -s или -o инвентарь записывается в JSON для повторного использования.
3. **Поиск PSS** — обнаруживает работающий PSS через скан /proc, читает pss.json для получения admin-порта (если в конфиге его нет — 43971); --pss-base http://host:port обходит автообнаружение.
4. **Подтверждение сопоставления** — интерактивный диалог спрашивает, как сопоставить каждый захваченный сервис с существующим SPTS-фидером; --non-interactive принимает предложения и завершает работу при конфликте; --target-mpts <id> пропускает запрос выбора MPTS.

5. **Авто-снятие паузы с фидеров** — для каждого сопоставленного SPTS / muxer-output утилита посылает {"pause" : false}, если фидер был на паузе, чтобы мультиплексор реально получал данные после apply.
6. **Авто-настройка битрейта** — если $\text{captured_bitrate} \times (1 + \text{headroom}\%)$ превышает `mpegts-output-bitrate` целевого MPTS, утилита поднимает этот лимит на PSS одним POST (с округлением вверх до ближайших 1000 kbps). Отключается через `--no-bitrate-adjust`.
7. **Планирование** — сравнивает инвентарь с текущим деревом `/config/stream` на PSS, готовит HTTP POST-запросы только для отличающихся полей. ES PID remap генерируется как identity-пары (`mpegts-pid-old` $\equiv$ `mpegts-pid-new`), чтобы мультиплексированный вывод сохранял каждый исходный PID без изменений — включая PCR, video, audio, teletext, SCTE-35, DSM-CC. На PSS 2.0.0.193 и новее план дополнительно включает режим сохранения исходных PID у muxer-входа цели и задаёт порядок программ в PAT (см. *Что мигрируется*).
8. **Применение** — отправляет запланированные POST-запросы, затем переключает MPTS `pause/unpause`, чтобы PSS перечитал конфигурацию; при `--dry-run` план только печатается.
9. **Verify** (включён по умолчанию, даже если план пустой) — захватывает результирующий MPTS через один из UDP-выходов PSS и сравнивает с целью; критические расхождения (TSID, ONID, service_id, name, type, LCN) → код выхода 5.

Повторный запуск всего конвейера **идемпотентен**: второй прогон сообщает `no changes needed`, если PSS уже соответствует инвентарю, а `verify` подтверждает это новым захватом.

Опции CLI

Выбор режима

Опция	Описание	По умолчанию
<code>-f, --file <path></code>	Путь к migration JSON (используется как импорт по умолчанию без <code>-i</code> и как цель сохранения с <code>-s</code>)	<code>./migrate.json</code>
<code>-s, --save</code>	Сохранить захваченный инвентарь в migration-файл (комбинируется с потоковым входом; <code>apply</code> всё равно выполняется)	—
<code>-o, --output <file></code>	Только захват: запись в файл, без <code>apply</code>	—
<code>-i, --input <file></code>	Только <code>apply</code> : загрузка файла, без захвата	—

Захват

Опция	Описание	По умолчанию
<code>-t, --time <sec></code>	Максимальная длительность захвата	30
<code>-b, --bitrate <Mbps></code>	Подсказка битрейта TS для пейсинга в файловом режиме	38.8

Apply / Verify

Опция	Описание	По умолчанию
--target-mpts <id>	Пропустить запрос выбора MPTS; применить к этому stream id на PSS	—
--non-interactive	Авто-принять предложения диалога; завершиться при конфликте	—
--dry-run	Распечатать план POST'ов, ничего не отправлять	—
--pss-base <url>	Переопределить автообнаружение PSS (например, http://host:8808)	авто
--no-verify	Пропустить пост-apply-захват и diff	verify включён
--verify-time <s>	Окно захвата для верификации	10
--no-bitrate-adjust	Не поднимать mpegts-output-bitrate на целевом MPTS	подъём включён
--bitrate-headroom <pct>	Запас по битрейту над измеренным значением при подстройке	15

Прочее

Опция	Описание
-v, --verbose	Подробный лог (каждый HTTP POST и ветка диалога)
-h, --help	Показать справку и выйти

Migration-файл (migrate.json)

Человекочитаемый JSON с format_version: 1. Расположение по умолчанию ./migrate.json; переопределяется -f. Пример формы:

```
{
  "format_version": 1,
  "tool": "mpts_migrate",
  "capture": {
    "source": "udp://239.1.1.1:1234",
    "captured_at_utc": "2026-04-30T08:15:00Z",
    "duration_s": 8.2,
    "packets": 109344
  },
  "transport_stream": {
    "transport_stream_id": 1234,
    "original_network_id": 8442,
    "network_name": "Operator",
    "delivery": { "type": "terrestrial", "frequency_khz": 522000 }
  },
  "services": [
    {
      "service_id": 1, "pmt_pid": 256, "pcr_pid": 256,
      "service_type": 1, "service_name": "Channel 1",
      "provider_name": "MyProvider", "logical_channel_number": 101,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "program_order": 1,
    "free_ca_mode": false,
    "elementary_streams": [
      { "pid": 256, "stream_type": 27, "language": "rus" }
    ]
  }
}
}

```

Файл можно править перед повторным apply: переименовать сервисы, изменить LCN, сменить service_type, скорректировать network_name – mpts_migrate -i migrate.json отправит только изменённые поля.

Подключение к PSS

- **Автообнаружение:** сканирование /proc/<pid>/comm на pss, чтение его --config-файла, выбор web-server.bind-port (если ключ отсутствует – 43971).
- **Вручную:** --pss-base http://host:port полностью пропускает автообнаружение. Полезно для удалённого PSS или когда --dry-run должен сформировать план без живого PSS.
- Admin REST API не требует аутентификации на localhost.

Верификация

Если --no-verify **не** указан (по умолчанию), после применения плана утилита:

1. Ищет UDP-выход на localhost у целевого MPTS либо временно добавляет на 127.0.0.1:<auto> (с пометкой added by mpts_migrate for verification).
2. Захватывает живой MPTS через этот выход в течение --verify-time секунд (по умолчанию 10).
3. Сравнивает захваченный инвентарь с целью:
 - **Критическое** расхождение (TSID, ONID, network name, service_id, name, type, LCN) → код выхода 5.
 - **Мягкое** расхождение (PMT PID, количество ES, имя провайдера, порядок программ в PAT) → выводится как warning, код выхода 0.
4. Если целевой MPTS перегружен (битрейт выхода $\geq$ настроенного mpegts-output-bitrate), печатается WARNING: target MPTS is overloaded – см. *Bitrate adjust* ниже.

Dry-run

--dry-run печатает каждый HTTP-запрос, который утилита *послала бы* (path, JSON-тело), но не отправляет ни одного. Полезно для:

- Просмотра плана с оператором перед коммитом.
- Генерации воспроизводимых наборов изменений в CI / change-management.
- Работы при недоступном PSS (комбинировать с --pss-base http://host:port).

Dry-run не запускает верификацию.

Bitrate adjust

По умолчанию, если `captured_bitrate × (1 + headroom%)` превышает `mpegts-output-bitrate` целевого MPTS, утилита поднимает этот лимит на PSS перед применением изменений SI/PSI. Без достаточного запаса перегруженный MPTS теряет низкоприоритетные данные — типичные симптомы: пропадание звука на радиосервисах, периодические ошибки CRC EIT. Отключается через --no-bitrate-adjust, запас регулируется через --bitrate-headroom <pct> (по умолчанию 15).

Коды выхода

Код	Значение
0	Успех — apply и (если включена) верификация прошли чисто. Также возвращается успешным --dry-run.
1	Ошибка аргументов / файла / обнаружения
2	В источнике не найдено PAT — вход не является валидным MPEG-TS, либо окно захвата слишком короткое; также возвращается, когда оператор отклонил подтверждение сопоставления или вышел из диалога (apply прерван)
3	Один или несколько HTTP POST'ов apply не прошли
4	Apply прошёл, но целевой MPTS не вернулся в состояние <i>Running</i>
5	Верификация не прошла — захваченный поток отличается от цели в критических полях

Ограничения и подводные камни

- **PSS должен заранее держать целевой MPTS и фидеры:** утилита не создаёт новые потоки. Целевой MPTS и хотя бы столько SPTS-фидеров, сколько сервисов в инвентаре, должны существовать заранее; сервисы без свободного фидера пропускаются в диалоге.
- **Источник MPTS должен быть доступен** при захвате (режимы 1, 2, save+apply): URL, переданный позиционным аргументом <input>, должен отдавать MPEG-TS-поток; для UDP multicast IGMP / firewall должен это разрешать.
- **ES PID remap применяется автоматически:** для каждого сервиса identity remap (`mpegts-pid-old ≡ mpegts-pid-new`) генерируется для каждого PCR/video/audio/teletext/data PID, чтобы мультиплексированный вывод сохранял исходные PID-ы byte-exact. Раскладка PMT остаётся стабильной от миграции к миграции — даже старые приёмники (STB до 2015 г., Samsung до серии H, LG до WebOS 3.0), кеширующие PID-ы, не требуют пересканирования.

- **Дескриптор доставки должен соответствовать реальному носителю** для приёмников, перенастраивающихся через NIT (DVB-T/T2/C/S). Несоответствующий блок `delivery` может увести приёмник на неверную частоту.
- **Согласованность LCN** между основным и резервным MPTS критична для failover — если они различаются, позиции каналов в списке приёмника сместятся после переключения.
- **Provider name на PSS — общий для мультиплекса** (один `sdt-provider-name` на `muxer-input` MPTS). Утилита применяет его автоматически, если все захваченные сервисы имеют одно и то же значение; если у разных сервисов разные `provider_name`, печатается `warning`, поле остаётся нетронутым — выбор за оператором.
- **Не инструмент конфигурации PSS:** `mpts_migrate` трогает только поля идентичности SI/PSI, флаг `pause` на каждом фидере, (опционально) `mpegs-output-bitrate`, а на PSS 2.0.0.193 и новее — ключи режима сохранения PID и порядка программ. Кодеры, входы, шифрование, расписания и т.п. он не настраивает.
- **На PSS старше 2.0.0.193** режим сохранения исходных PID у муксера и порядок программ в PAT не поддерживаются: утилита печатает предупреждение, PID закрепляются только `identity`-парами, а порядок программ после миграции может отличаться от исходного.

Диагностика

Симптом	Вероятная причина / решение
Error: no PAT seen in stream	источник не MPEG-TS, IGMP/firewall блокирует multicast или -t слишком короткий
Error: cannot reach PSS	использовать <code>--pss-base http://host:port</code> , чтобы обойти автообнаружение
Apply прошёл, но MPTS остаётся на паузе	проверить лог PSS; перезапустить с <code>-v</code> , чтобы увидеть полный план POST'ов
Верификация показывает критическое расхождение	сравнить цель и захваченный поток в JSON; обычно фидер ошибочно сопоставлен с не тем сервисом в диалоге
WARNING: target MPTS is overloaded	поднять <code>mpegs-output-bitrate</code> на PSS либо <code>--bitrate-headroom <higher %></code> ; без запаса повреждаются аудио радиосервисов и PSI-таблицы

1.12 Справочные материалы

1.13 История версий

1.13.1 версия 2.0.0.206 Beta

25.07.2026

- **Meshwork (мультидомен)** — несколько серверов объединяются в именованные домены, и домен служит границей изоляции: членство узлов, каталог ресурсов и аварии расходятся только внутри своего домена, между доменами не передаётся ничего. Узел может состоять сразу в нескольких доменах, не смешивая их данные. Подробно — [Meshwork](#).

- **Автообнаружение узлов** — достаточно задать домен и одну точку входа: узлы обмениваются списками соседей и сами достраивают полносвязную сеть прямых связей. Общий доменный ключ вырабатывается автоматически и передаётся только по защищённому каналу — TLS либо доверенный сегмент локальной сети.
- **Распределённая библиотека ресурсов** — выходы UDP, RTP, Pro-MPEG и RIST, multicast-входы UDP, RTP и RIST, а также связи PS1 и SRT публикуются в общем каталоге домена (/data/library): единый список multicast-групп, портов, VLAN, SSM-источников и связей точка-точка с указанием узла и потока. По каталогу видно, какие адреса и порты уже заняты, а новый вход подключается прямо из него — выбором потока, уже отдаваемого на другом узле домена, без ручного ввода адреса.
- **Обзор узлов домена** — по каждому узлу видны статус, время последнего контакта, роль, регион, версия сборки, загрузка CPU, трафик по физическим интерфейсам и число потоков в эфире, а по каждой межузловой связи PS1 или SRT — битрейт, RTT, доля повторов и потерь и вердикт состояния.
- **Работа через NAT** — узел за NAT участвует в домене полноценно и без проброса входящих портов: он подключается сам, публичная точка доступа достраивается по фактически видимому адресу и объявленному порту, а его состояние соседи ретранслируют на один хоп — узел виден и тем, кто не может обратиться к нему напрямую.
- **Автологин связей внутри домена** — для PS1- и SRT-связей между узлами одного домена не нужно заводить логин и пароль на каждую связь: на вызывающей стороне (вход PS1, вход SRT, выход SRT в режиме caller) достаточно включить настройку «Meshwork peer auth», и авторизация выполняется по общему доменному секрету. Подтверждённый узел-партнёр и его поток видны в библиотеке ресурсов и в списке сессий.
- **Low-Latency HLS и MPEG-DASH на CMAF** — новый режим раздачи «OTT / LL-HLS / DASH»: встроенный мультиплексор формирует фрагментированный MP4 (CMAF), поверх которого отдаются MPEG-DASH на /dash и Low-Latency HLS на /llhls; адаптивные наборы собираются и для этих форматов. Подробно — [Режимы раздачи](#).
- **Малая задержка LL-HLS** — медиаплейлист дробится на части, применяются блокирующая перезагрузка плейлиста и подсказка предзагрузки, поэтому плеер начинает воспроизведение, не дожидаясь готовности полного сегмента. Сегменты CMAF несут Producer Reference Time, манифест DASH объявляет UTCTiming и целевую задержку, а настройка «Целевая длительность LL-части (ms)» применяется на лету, без перезапуска потока.
- **HTTP/3 (QUIC)** — HLS, MPEG-DASH, LL-HLS и MPEG-TS over HTTP отдаются поверх QUIC на отдельном UDP-порту, с необязательным 0-RTT; переход на QUIC клиент запрашивает параметром ?h3. Административные маршруты остаются только на TCP.
- **Выравнивание сегментов по IDR** — сегментер различает IDR и обычный I-кадр: на источниках с closed-GOP каждый сегмент начинается с полноценной точки входа, поэтому плеер открывает поток с любого сегмента; на источниках с open-GOP границей служит ближайший I-кадр.
- **Защита контента (DRM)** — OTT-раздачу потока можно шифровать: HLS AES-128 с ключом от самого сервера или от внешнего key-сервера и с необязательной ротацией по временным окнам, либо MPEG Common Encryption (ISO/IEC 23001-7) по схемам cenc и cbcs для CMAF и DASH. Шифрование выполняется один раз, в момент формирования сегмента, поэтому архив DVR хранится зашифрованным и воспроизводится через любое число смен ключа. Подробно — [Защита контента \(DRM\)](#).
- **DVR (сетевой архив)** — каждый OTT-канал пишется на диск параллельно с раздачей, той же сегментацией и без отдельного рекордера; в режиме низкой задержки архив ведётся двумя линиями, MPEG-TS и CMAF, поэтому запись доступна в том же контейнере, что и живое вещание. Подробно — [DVR](#).

- **Воспроизведение архива (VOD)** — архив отдаётся по тем же HLS, DASH и LL-HLS URL, что и живое вещание: `t=<время>` включает режим VOD и задаёт начало окна, `d=<секунды>` — его длительность. Плейлист HLS при этом закрытый, манифест DASH статический, разрывы записи оформлены отдельными периодами; адаптивные наборы воспроизводят архив теми же параметрами.
- **Catch-up по EPG** — вместо `t` и `d` достаточно передать `epg=<время>`: сервер сам находит на привязанном канале EPG передачу, идущую в этот момент, и берёт её начало и длительность как границы окна воспроизведения.
- **Субтитры в архиве** — дорожки WebVTT пишутся на диск вместе с сегментами и воспроизводятся в VOD по тем же URL; окна без реплик места на диске не занимают.
- **Хранилища DVR** — хранилищ может быть несколько, у каждого свой предел заполнения и свой порядок очистки. Глубина архива («Хранение (часы)», до 90 суток) и защищённый минимум задаются отдельно для каждого потока, а очистка по свободному месту не трогает ни защищённый минимум, ни окна открытых VOD-сессий.
- **Монитор DVR** — отдельный экран показывает состояние и заполнение хранилищ, объём и глубину архива по потокам и задержки чтения и записи, а гистограмма покрытия (`/data/dvrstat`) отмечает не только дыры в архиве, но и их причину — обрыв входа, смену PMT, скремблирование, срабатывание очистки. Отсюда же стираются архив отдельного потока и каталоги удалённых потоков.
- **Алертер** — новая служба оповещений: каждая авария становится инцидентом, который поднимается, обновляется и снимается автоматически, а дедуплицированный активный набор показывает только то, что происходит сейчас. Серьёзность задаётся единой шкалой — от информационной до требующей вмешательства администратора; такой инцидент снимается подтверждением оператора. Подробно — [Оповещения \(алертер\)](#).
- **Каталог кодов аварий** — более пятидесяти типов инцидентов в одном списке: потоки и их входы и выходы, нарушения TR 101 290, ресурсы узла, хранилища и здоровье записи DVR, приём DVB и CI/CAM, транскодеры, OTT, сертификаты и узлы домена. Пороги срабатывания настраиваются в разделе оповещений.
- **Внешняя доставка оповещений** — аварии уходят за пределы веб-консоли: письмом по SMTP (STARTTLS, implicit TLS, AUTH), сообщением в Telegram и запуском произвольной команды, которой инцидент передаётся переменными окружения и полным JSON на стандартный вход. У каждого канала свой выключатель и свой порог серьёзности.
- **Аварии всего домена** — дежурный видит аварии любого узла с любого другого: активный набор расходится вместе с `keep-alive` и снимается сразу, как только исчез у источника, а у каждой строки указан узел-источник. Аппаратные и системные аварии по умолчанию остаются локальными и поднимаются до доменных отдельной настройкой.
- **Журнал в базе данных** — сообщения сервиса пишутся в SQLite: типизированные записи с источником и уровнем серьёзности, свёртка идущих подряд повторов в одну строку со счётчиком, ограничение хранения по сроку и по объёму. В отличие от активного набора аварий журнал переживает перезапуск.
- **Монитор TR 101 290** — непрерывный контроль соответствия входного потока стандарту: единый вердикт («Норма», «Проблема», «Проверка...») и структурный отчёт по приоритетам 1, 2 и 3, где для каждого показателя указаны пункт стандарта, измеренное значение и предел. Мультиплекс MPTS оценивается целиком — по всем PID, программам и таблицам SI. Поток автоматически классифицируется как CBR или VBR, и проверки, осмысленные только при постоянном битрейте, на VBR-источнике не оцениваются и не дают ложных срабатываний. Подробно — [Монитор TR 101 290](#).
- **Дрейф опорного генератора** — систематический уход тактовой частоты источника измеряется

линейной регрессией в ppm при допуске ISO/IEC 13818-1 ± 30 ppm и выводится отдельной метрикой. Медленный уход отрабатывается плавными микросдвигами точки синхронизации («Компенсация дрейфа синхронизации», включена по умолчанию), поэтому на выходе нет рывков.

- **Модель буфера T-STD** — анализ видеобуфера эталонного декодера по ISO/IEC 13818-1 со счётчиками переполнений и опустошений для MPEG-2, H.264 и HEVC; отсчёт ведётся по часам PCR, а скорость слива подстраивается под фактический битрейт видео. Включается настройкой «Анализировать видеобуфер T-STD».
- **Анализ для вставки рекламы** — разбор секций SCTE-35 с событиями склейки, точки склейки транспортного уровня с настраиваемым упреждением и метки случайного доступа. Данные выводятся в статистике потока при включённом глубоком анализе.
- **Помощник для рекламаций** — по потоку с устойчивыми нарушениями формируется готовое текстовое задание для AI-чата, по которому тот составляет формальное письмо-рекламацию поставщику потока: перечень устойчивых нарушений, измеренные значения, пункты стандарта и влияние на декодер. Задание отдаётся запросом `GET /data/stream/<id>/ai-complaint-prompt`; название потока и адрес источника в текст не включаются.
- **CI/CAM (EN 50221)** — встроенный хост Common Interface: обнаружение модуля, чтение его имени и списка поддерживаемых CA_system_id, выбор программ для дескремблирования и передача CA_PMT по каждой из них. Модуль, совмещённый с DVB-приёмником, дескремблирует выбранные программы принятого мультиплекса инлайн, несколько одновременно; состояние слотов и вердикт по каждой программе видны в интерфейсе.
- **Программный дескремблер BISS-1 / BISS-E** — применяется не только к приёму с DVB-карты, но и к любому входу потока: ключ задаётся на SPTS целиком или по программам мультиплекса и меняется на лету, без перезапуска входа. На приёме с DVB-карты результат дополнительно проверяется по самому потоку, поэтому неверный ключ не выглядит как успешное дескремблирование, а поднимает отдельную аварию.
- **Очистка условного доступа на приёме DVB** — из мультиплекса, принятого DVB-картой, отдельной настройкой адаптера удаляются PID ECM и EMM, а из PMT — CA-дескрипторы открытых программ: дальше по тракту уходит чистый FTA-поток. При потере ключа разметка условного доступа возвращается, чтобы приёмник ниже по цепочке смог заново запросить доступ.
- **Телеметрия транскодера** — по каждому энкодеру публикуется медиаинформация перекодированного выхода: формат кадра, кодек видео, набор аудиокодексов и текущий битрейт, а по каждому процессу — загрузка CPU и занятая память.
- **RTMP и RTMPS** — канал публикуется на любой RTMP-приёмник, включая YouTube и Facebook: видео H.264 или HEVC (HEVC — через Enhanced RTMP) и одна дорожка AAC, для `rtmps://` узел подключается как TLS-клиент. В обратную сторону вход сам забирает поток с источника `rtmp://` или `rtmps://` и ремультиплексирует FLV в MPEG-TS.
- **Бесшовное переключение источника** — при смене активного входа на передатчике приёмные пиры PS1 не переподключаются: нумерация и метки времени остаются непрерывными, всплеск очереди гасится отбрасыванием самых старых пакетов, а пропуск добирается штатной ретрансмиссией.
- **Адаптивная ретрансмиссия PS1** — приёмник измеряет фактический RTT до передатчика и сам подстраивает под него интервалы повторных запросов; измеренный RTT, текущий интервал перезапроса и признак слишком малой задержки выведены в статистику связи. Задержка приёмного тоннеля задаётся напрямую в миллисекундах вместо прежнего множителя от RTT.
- **Состояния «Переподключение» и «Действие администратора»** — потерявший источник вход или выход переоткрывается на месте и остаётся в состоянии «Переподключение» на всё

время попыток, поднимая одно предупреждение на эпизод вместо сообщения на каждую попытку. Причину, которую повтором не устранить — занятый порт, отсутствующий интерфейс, данные не MPEG-TS, отказ SRT по шифрованию, — узел переводит в состояние «Действие администратора» с удерживаемым оповещением, а правка настроек применяется сразу.

- **MPTS-мультиплексор: исходная идентичность** — программы могут сохранять свои исходные PID, а порядок программ в PAT, SDT и NIT задаётся вручную, поэтому мультиплекс переносится на **Perfect Streamer** без смены идентичности потока для приёмного оборудования. Программы с TS-скремблированием переносятся с исходными метками времени.
- **Интеграция с внешним биллингом** — каждая зрительская сессия авторизуется во внешней биллинг- или CRM-системе, удерживается периодической ре-авторизацией и по завершении сообщается ей же: жизненный цикл Start / Interim / Stop в стиле RADIUS, отзыв подписки сбрасывает сессию посреди просмотра. Охвачены OTT, пиринг PS1 и выходы SRT, а сам биллинг подключается шаблонами запроса и разбора ответа, без доработки ПО.
- **Встроенный клиент ACME (Let's Encrypt)** — выпуск и автоматическое продление сертификатов для веб-сервера, HTTP/OTT-сервера и EPG-сервера: сертификат заказывается на каждое настроенное имя хоста, продление проверяется ежедневно, при неудаче поднимается авария. Поддерживаются произвольный удостоверяющий центр и внешняя привязка учётной записи, а новый сертификат применяется без перезапуска сервиса.
- **Новый веб-интерфейс** — управление узлом переведено на новую консоль, которая открывается по адресу узла (путь /admin): обзор, потоки, мониторинг системы, DVB, DVR, транскодеры, клиенты, журнал, аварии и все настройки сервера. Отдельные схемы показывают топологию домена Meshwork и тракт выбранного потока от входов до выходов. Прежний интерфейс сохранён по адресу /classic. Подробно — [Веб-интерфейс](#).
- **Обновление в реальном времени** — узел отдаёт своё состояние потоком событий на /data/events: при подключении приходит полный снимок, дальше раз в секунду обновляются кадры потоков, системных ресурсов, клиентов и DVB-адаптеров, а аварии и смены состояния приходят событиями. Интерфейс и внешние панели работают без периодического опроса.
- **Контекстная справка** — каждый экран и каждое важное окно нового интерфейса снабжены вызовом справки, который открывает раздел документации именно по тому, что сейчас открыто: общая кнопка справки — для экрана, кнопка «?» в заголовке окна — для этого окна. Раздел открывается на языке интерфейса.
- **Языки, темы и раскладки** — интерфейс переведён на шесть языков (русский, английский, немецкий, французский, испанский, португальский), поддерживает светлое и тёмное оформление и сам подстраивает плотность раскладки под телефон, планшет, рабочую станцию и видеостену.
- Прочие улучшения и исправления ошибок.

1.14 Roadmap

Раздел перечисляет возможности, которые заявлены, но в текущую поставку ещё не вошли. Для каждой указано, что уже работает сегодня и чего пока нет, — чтобы планирование внедрения опиралось на фактические возможности узла. Состав и сроки могут меняться; актуальность уточняйте у поставщика.

1.14.1 Дескремблирование отдельных потоков через CAM

Работает сегодня. Аппаратное дескремблирование CI/CAM (EN 50221) доступно на собственном DVB-приёмнике узла: модуль, установленный в CI-слот тюнерной карты, расшифровывает выбранные программы принятого этим тюнером мультиплекса — несколько программ одновременно, список применяется на лету без перезапуска адаптера. Состояние слота, название модуля, поддерживаемые CA-системы и вердикт модуля о подписке по каждой программе показаны на экране «Оборудование»; предусмотрены оповещения об извлечении модуля, отсутствии подписки и ошибке условного доступа ([Дескремблирование](#)).

Чего пока нет. Пропустить через CAM-модуль поток, не входящий в мультиплекс собственного тюнера, нельзя — тракта «поток → модуль → поток» не существует. SPTS или MPTS, принятый по сети либо из файла, аппаратно не дескремблируется; для таких источников доступно только программное дескремблирование BISS-1 / BISS-E ([SPTS-потоки](#), [Дескремблирование BISS](#)). Модуль, рассчитанный на такой режим работы, узел распознаёт и показывает в инвентаре оборудования, но данные через него не прогоняет.

1.14.2 Промышленные системы DRM для OTT-раздачи

Работает сегодня. OTT-раздачу узел шифрует собственными средствами: HLS AES-128 — с выдачей ключа по сессионному URL либо ссылкой на внешний сервер ключей и с ротацией производных ключей по временному окну, а также MPEG Common Encryption (ISO/IEC 23001-7) в схемах cenc и cbcs поверх CMAF. Шифрование выполняется один раз, при формировании сегмента, поэтому DVR-архив хранится и воспроизводится зашифрованным ([Защита контента \(DRM\)](#)).

Чего пока нет. Готовой интеграции с лицензионными серверами Widevine, PlayReady и FairPlay нет: манифест и сегмент инициализации несут только общую сигнализацию защиты — без системных заголовков этих DRM и без адреса получения лицензии. Автоматического обмена ключами с провайдером ключей (CPIX, SPEKE) тоже нет: пару «ключ — идентификатор ключа» оператор передаёт в свою DRM-инфраструктуру вручную. Схема HLS SAMPLE-AES не поддерживается, а Common Encryption отдаётся только по MPEG-DASH, поэтому защищённой раздачи HLS поверх CMAF и Low-Latency HLS пока нет.

1.14.3 Сигнализация точек врезки рекламы в OTT-раздаче

Работает сегодня. Анализатор распознаёт события SCTE-35 и точки склейки и показывает их в отчёте ([Постоянные измерения](#)). При ремультимплексировании PID с сигнализацией переносится в выходной мультиплекс без изменений.

Чего пока нет. Границы рекламных блоков не выносятся в плейлисты HLS и манифесты DASH: сегменты по точке склейки не нарезаются, плееру и внешней системе подстановки рекламы ничего не сигнализируется; подмены рекламных блоков в самом потоке тоже нет. Через транскодер сигнализация не проходит — в транскодированной копии канала метки склейки теряются.