

Mcaster Manual

Sistema modular profissional para o processamento de sinais de TV

Table of contents

1. Produtos	3
2. Mcaster	4
2.1 Características funcionais	5
3. Instalação	7
3.1 Instalação	7
3.2 Início rápido	9
3.3 Administradores	11
4. Captura de sinal	20
4.1 Captura de multicast e MPTS	20
4.2 Captura por SRT e RIST	22
4.3 Publicação SRT	24
4.4 Recepção de ST 2110	28
4.5 Reserva de entradas	36
4.6 Erros de entrada	38
5. Processamento	40
5.1 Saída MPEG-TS	40
5.2 CBR para DVB	42
5.3 Legendas e teletexto	44
6. Entrega	46
6.1 Entrega à rede	46
6.2 Entrega por SRT	48
6.3 A cópia OTT	50
6.4 Monitoramento de entrega	52
7. Arquivo	57
7.1 O arquivo	57
8. Controle de qualidade	59
8.1 Contadores de transporte (TR 101 290)	59
8.2 Monitoramento	61

1. Produtos

2. Mcaster

2.0.1 Mcaster

Mcaster é a cabeça de rede da Flussonic. O sinal chega a ela de um receptor de satélite, de uma rede vizinha, de um parceiro ou do seu próprio playout; o Mcaster o leva à forma que a rede de entrega aceita e **faz chegar ao destino** — ao multicast de uma rede a cabo, a um modulador, a outra cidade por SRT, a um local de reserva.

A diferença chave em relação ao [Catena](#): o Catena termina no espectador, o Mcaster termina no ponto de entrega do sinal. Além desse ponto há um espectador, mas esse espectador não é o cliente da cabeça de rede. Daí o assunto deste manual: um fluxo de transporte com seus próprios PSI/SI, um bitrate constante, contadores de transporte e comprovação de entrega por destino.

O que você pode colocar no ar

- **Recepção de satélite e de uma rede de terceiros.** O programa é escolhido de um tronco MPTS, e o próprio tronco é recebido uma única vez, não importa quantos programas dele sejam aproveitados.
- **Transporte entre locais.** SRT pela internet aberta com criptografia e recuperação de perdas; UDP/RTP dentro da própria rede.
- **Preparação do sinal para o receptor.** PNR, PIDs e SDT próprios por destino; um bitrate constante para o modulador; teletexto e legendas que chegam vivos à saída.
- **Entrega comprovada.** Por destino: status com um motivo de falha nomeado, taxa, bytes, reconexões, retransmissões.
- **Um arquivo no próprio local.** Gravação direto na cabeça de rede — para o dia em que alguém perguntar o que exatamente saiu às 03:40.

A lista completa está na página [Características funcionais](#).

Como ele é construído

O Mcaster é uma interface e uma licença sobre o motor de streaming [Sapsan](#); o build é o mesmo. Daí seguem-se três consequências práticas.

- **Tudo é configurado pelo painel central.** Streams, máquinas, processamento e destinos de entrega vivem numa única interface web; não é preciso editar arquivos nos servidores.
- **Um processo, uma estação.** A instalação mínima é uma única máquina: motor, painel e API num só processo.
- **Várias máquinas, uma interface.** Quando há mais de uma estação, elas se conectam a um plano de controle e o operador as vê no mesmo painel: distribuição dos streams pelas máquinas, um arquivo compartilhado, estatísticas compartilhadas. Uma máquina conectada não tem configurações próprias — ela as recebe do plano de controle.

Um stream dentro do Mcaster é um único serviço: tem entradas (principal e reserva), processamento e saídas. Tudo o que se descreve neste manual é uma propriedade do stream.

Por onde começar

- **Instalação** — do pacote a um processo em execução.
- **Início rápido** — capturar um multicast e devolvê-lo à rede, com uma verificação em cada passo.
- **Captura de multicast e MPTS** — a primeira coisa que uma estação faz.
- **Monitoramento da entrega** — como se certificar de que o sinal chegou.

2.1 Características funcionais

A lista de funções da cabeça de rede Mcast. Cada função está descrita numa página deste manual, com o link ao lado.

2.1.1 Captura de sinal

- MPEG-TS por UDP — unicast e multicast, IPv4 e IPv6, com assinatura do grupo e seleção de fonte (SSM) — [mais](#).
- Seleção de programa de um tronco MPTS por número (`program=`); o tronco é recebido uma única vez para todos os seus programas, não importa quantos streams o analisem — [mais](#).
- MPEG-TS por HTTP, incluindo transferência em chunks; uma conexão rompida e uma fonte silenciosa são distinguidas — [mais](#).
- Captura por SRT: uma porta dedicada por stream, criptografia AES, `streamid`, um orçamento de recuperação de perdas ajustável — [mais](#).
- Captura por RTMP e HLS — para fontes vindas do mundo OTT.
- Codecs de entrada: vídeo H264, HEVC, MPEG-2; áudio AAC, AC3, EAC3, MPEG-2; teletexto DVB, legendas DVB, legendas CEA-608/708, metadados KLV, marcadores SCTE-35.
- Redundância de entrada: várias fontes por stream, comutação automática pela saúde da fonte, retorno à principal após uma janela limpa — [mais](#).
- Detalhamento causal dos erros de entrada e destaque dos minutos sem sinal — [mais](#).

2.1.2 Processamento

- Formação do fluxo de transporte de saída: PAT/PMT/SDT, número do programa, PIDs explícitos das faixas, nomes do serviço e do provedor, `transport_stream_id` e `original_network_id` — [mais](#).
- Tabelas de serviço próprias por destino: o mesmo sinal vai para receptores diferentes sob PNRs e PIDs diferentes, sem um segundo multiplexador — [mais](#).
- Bitrate constante (CBR) com preenchimento de pacotes nulos, PCR sem deriva e consideração do modelo de buffer do decodificador (HRD) — [mais](#).
- Transcodificação: vídeo H264, HEVC e AV1; áudio AAC e Opus; desentrelaçamento, escalonamento, controle de GOP — [mais](#).
- Teletexto e legendas: passagem para a saída como chegaram, extração para uma faixa de texto, ou as duas representações ao mesmo tempo — [mais](#).

2.1.3 Entrega

- Envio para a rede por UDP: multicast e unicast, controle de TTL, seleção de interface de saída e de endereço de vinculação — [mais](#).
- Encapsulamento RTP conforme RFC 2250 / SMPTE ST 222-2 — a mesma saída, o esquema `rtp://` — [mais](#).
- Envio por SRT a um destino fixo com criptografia e `streamid`; serviço SRT pela porta própria do stream — [mais](#).
- Envio por RTMP para plataformas e CDNs, incluindo HEVC e AV1 (enhanced RTMP) — [mais](#).
- Vários destinos por stream ao mesmo tempo; destinos com configurações idênticas dividem um único fluxo idêntico byte a byte — [mais](#).
- Reanimação automática de um destino morto sem intervenção manual — [mais](#).
- Uma cópia OTT do sinal: HLS, LL-HLS e DASH da mesma estação — [mais](#).
- Criptografia da cópia OTT: Common Encryption `cbs` para HLS e DASH, a chave das configurações ou do seu servidor pelo SPEKE, sinalização para Widevine, PlayReady, FairPlay e ClearKey; os pushes saem abertos — [mais](#).

2.1.4 Arquivo

- Gravação em um armazenamento append-only distribuído em vários discos, retenção por profundidade e por tamanho — [mais](#).
- Reprodução do arquivo por tempo absoluto, rewind, junção perfeita com o ao vivo, prévias JPEG — [mais](#).

2.1.5 Controle de qualidade

- Contadores de transporte por PID: violações de continuity counter, TEI, pacotes cifrados, erros de CRC no PSI, PES quebrados, jitter de PCR contra os limiares do padrão, déficits do buffer do decodificador — [mais](#).
- Contadores de entrega por destino: status com motivo, taxa, bytes, erros, reconexões, retransmissões — [mais](#).
- Detalhamento causal dos erros de entrada: defeitos de mídia e falhas de fonte são separados, cada causa é a sua própria linha no gráfico — [mais](#).
- Um servidor Prometheus embutido como datasource pronto para o Grafana, um endpoint de scrape para a sua própria TSDB, telemetria para o monitoramento em nuvem Retroview — [mais](#).

2.1.6 Gerenciamento

- Um painel central: streams, máquinas, distribuição, arquivo e sessões numa única interface web.
- As edições são aplicadas em tempo real, sem interromper a transmissão.
- Um painel só para o cluster inteiro: uma máquina conectada recebe suas configurações do plano de controle.
- Uma API de gerenciamento para integração com sistemas de provisionamento de terceiros.
- Uma recusa estruturada de uma configuração inválida: 422 com o campo culpado indicado, e não uma distorção silenciosa da configuração.

3. Instalação

3.1 Instalação

O Mcaster se instala numa máquina e nela mesma roda: motor, painel de controle e API num único processo. Esta página é sobre o que o pacote instala, onde vivem os segredos e o que fica aberto para fora. O caminho de uma máquina limpa até um sinal na rede está no [Início rápido](#).

3.1.1 O pacote

O Mcaster é distribuído como pacote deb `mcaster` do repositório da Flussonic:

```
curl -L http://apt.flussonic.com/repo/master/dev.key > /etc/apt/trusted.gpg.d/dev.gpg
echo "deb http://apt.flussonic.com/branch/sapsan/master repo/" > /etc/apt/sources.list.d/flussonic.list
apt update
apt install --install-recommends mcaster
```

O pacote cuida de toda a superfície da instalação:

O quê	Onde
Unidade	<code>mcaster.service</code>
Configurações do processo	<code>/etc/mcaster/</code> — o que o próprio processo precisa ao iniciar: a porta do painel, o endereço do banco, a licença
Segredos	<code>/etc/mcaster/secrets.env</code> (proprietário root, modo 0600)
Licença	<code>/etc/mcaster/license.txt</code>
Estado	<code>/var/lib/mcaster</code> — identidade da máquina, cache de ativação, chaves
Interface	<code>/usr/share/mcaster/ui</code>

O pacote instala automaticamente o motor de streaming exatamente na sua própria versão, junto com o PostgreSQL. A unidade base `sapsan.service` é retirada de uma máquina assim: deve rodar exatamente um processo. Os produtos são mutuamente exclusivos — Mcaster e Catena não dividem a mesma máquina.



Note

A atualização do pacote não reinicia uma unidade em execução. A máquina passa para o novo par «binário + interface» com um `systemctl restart mcaster` manual — assim o reinício continua sendo uma decisão do operador, e não um efeito colateral do `apt upgrade`.

3.1.2 Licença

Sem uma chave válida o processo não inicia. A chave é pedida durante a instalação e gravada em `/etc/mcaster/license.txt`; para implantações não assistidas ela é fornecida por meio de `debconf-set-selections` ou da variável `LICENSE_KEY`.

A chave é emitida **para um produto**: a máquina se declara como `mcaster`, e uma chave emitida para outro produto não serve — nenhuma ativação é concedida, e isso aparece no log imediatamente, e não por meio de uma função que não funciona.

Deixar a pergunta vazia é permitido: o pacote instala, mas o serviço fica em `failed`. Coloque a chave no mesmo arquivo e execute `systemctl start mcaster`.

3.1.3 PostgreSQL

O Mcaster é uma instalação de controle e mantém seu estado (streams, máquinas, estatísticas, registro) no PostgreSQL. O pós-instalação cria a role e o banco de dados `mcaster`, gera as senhas do banco e do administrador e as grava em `/etc/mcaster/secrets.env`:

```
DATABASE_URL=postgres://mcaster:...@127.0.0.1:5432/mcaster
EDIT_AUTH_PASSWORD=...
```

A atualização do pacote não rotaciona os segredos. Para mover o banco de dados para um cluster gerenciado com backups, troque o `DATABASE_URL` e reinicie o serviço — o Mcaster aplica seu esquema sozinho ao iniciar, não há uma etapa de migração separada.

3.1.4 Portas

Logo após a instalação, sobe a porta HTTP 80 — um valor *inicial* que vale só até os listeners serem definidos no painel. Depois disso as portas pertencem às configurações da máquina e são editadas lá.

Porta	Quem define	Para quê
HTTP (inicialmente 80)	<code>INITIAL_HTTP_PORT</code> , depois as configurações da máquina	Painel, API, saída por HTTP
Entradas UDP	A porta de cada fonte multicast	Captura de MPEG-TS
SRT (UDP)	A porta de serviço SRT do stream ou o listener compartilhado	Captura e saída por SRT
RTMP (TCP)	Um listener nas configurações da máquina	Captura e saída por RTMP

Os destinos de saída não ocupam portas: um push abre o próprio socket.

O multicast exige atenção às rotas. A captura é uma assinatura do grupo (IGMP/MLD) na interface por onde o grupo chega; a saída é a [escolha da interface de saída e do TTL](#) no próprio destino. Numa máquina com várias placas de rede, as duas escolhas são feitas explicitamente; caso contrário, os pacotes saem pela rede errada.

3.1.5 Execução em Docker

A imagem de produto é `flussonic/mcaster`: o runtime compartilhado mais a interface do Mcaster e a tag de produto. Ela precisa do PostgreSQL, de uma chave de licença e de um diretório de estado persistente:

```
docker run -d --name mcaster \
-e DATABASE_URL=postgres://mcaster:mcaster@db:5432/mcaster \
-e EDIT_AUTH_LOGIN=admin -e EDIT_AUTH_PASSWORD=secret \
-e LICENSE_KEY="$LICENSE_KEY" \
-p 8081:80 \
-v mcaster-state:/var/lib/sapsan \
flussonic/mcaster:latest
```

As portas de protocolos de mídia que as configurações dos streams ativarem são publicadas aqui também. O diretório de estado precisa sobreviver a reinícios: ele guarda a identidade da máquina e a ativação.

3.1.6 Verificação

```
systemctl status mcaster
journalctl -u mcaster -n 50
```

Em seguida vem o [Início rápido](#): entrar no painel, o primeiro multicast capturado e o primeiro destino de entrega.

3.2 Início rápido

Esta página leva uma máquina limpa até um sinal recebido da rede e devolvido à rede — a razão de existir de uma cabeça de rede. Aqui não se configuram cluster, arquivo nem transcodificação: isso vem depois.

3.2.1 O que você precisa

- Uma máquina com Ubuntu 24.04 e acesso root, no mesmo segmento de rede por onde circula o multicast.
- Acesso à internet — a licença é ativada online.
- Uma chave de licença da sua conta em my.flussonic.com. A própria instalação vai pedi-la.
- Um grupo multicast vivo carregando MPEG-TS — digamos `239.255.1.77:5500`. Se não houver, comece com a fonte sintética: ela exercita todo o caminho, exceto a captura.

3.2.2 Instale

```
curl -L http://apt.flussonic.com/repo/master/dev.key > /etc/apt/trusted.gpg.d/dev.gpg
echo "deb http://apt.flussonic.com/branch/sapsan/master repo/" > /etc/apt/sources.list.d/flussonic.list
apt update
apt install --install-recommends mcaster
```

Um único pacote basta: ele instala automaticamente o motor da sua própria versão e o PostgreSQL, cria o banco de dados, gera as senhas e inicia o `mcaster.service`. O `--install-recommends` mantém o transcodificador funcionando — ele chega como dependência recomendada.

O que exatamente o pacote coloca na máquina está na página [Instalação](#).

3.2.3 Entre no painel

O pacote gravou a senha do administrador no arquivo de segredos:

```
grep EDIT_AUTH_PASSWORD /etc/mcaster/secrets.env
```

Abra `http://<server-address>/` e entre como `admin` com essa senha.

3.2.4 Capture o multicast

Vá até **Streams**, digite o nome `tv1` e clique em **Criar**. O stream abre na aba **Fontes**.

Clique em **+ Adicionar fonte**, escolha o protocolo **UDP MPEGTS** e preencha:

- **Host** — o endereço do grupo, `239.255.1.77`;
- **Porta** — `5500`;
- **Tempo limite do peer (ms)** — quanto silêncio no grupo tolerar antes de reiniciar a captura; campo vazio significa o valor padrão.

Clique em **Salvar**. A estação assina o grupo e em poucos segundos o stream fica verde: aparecem um bitrate e uma lista de faixas.

Se o grupo carrega um multiplex em vez de um único serviço, coloque o número no campo **Programa (PNR)** — veja [Captura de multicast e MPTS](#).

3.2.5 Envie-o para a rede

Abra a aba **Saída** e clique em **+ Adicionar push** no bloco **Pushes**:

- **Nome** — `headend-out`. O nome é a identidade do destino: é por ele que o push é encontrado nas configurações, nas estatísticas e nas métricas.
- **Protocolo** — **MPEG-TS/UDP**.
- **Host e Porta** — o grupo para o qual o sinal é entregue, digamos `239.1.1.10` e `5000`.
- **TTL de multicast e Interface de saída** — preenchidos quando a máquina tem várias placas de rede ou o pacote precisa sobreviver a um roteador.

Clique em **Salvar**. Embaixo das configurações do push aparece uma linha de estatística ao vivo: **Enviando**, a taxa em kbit/s, erros por segundo, reconexões.

3.2.6 Verifique o que chegou

De uma máquina situada na rede de destino, assine o grupo e veja o que roda nele: primeiro se os pacotes chegam ao menos (`tcpdump -i <iface> host 239.1.1.10`), depois se o seu receptor ou analisador de fluxo de transporte decodifica um programa com as mesmas faixas que o cartão do stream mostra.

O que significam o número do programa e os PIDs na sua saída, e como defini-los, está na página [MPEG-TS de saída](#).

A segunda fonte da verdade são as estatísticas da própria estação: a linha sob o push responde «estamos enviando», o receptor responde «chegou». A divergência entre essas duas respostas é exatamente a cara de um problema de rede. Os contadores são explicados na página [Monitoramento da entrega](#).

3.2.7 Se algo deu errado

```
systemctl status mcaster
journalctl -u mcaster -n 50
```

- **O serviço não está rodando e o log fala da licença.** A chave não foi informada na instalação. Coloque-a em `/etc/mcaster/license.txt` e execute `systemctl start mcaster`.
- **O serviço não está rodando e o log fala do banco de dados.** O PostgreSQL está inacessível: escreva um `DATABASE_URL` funcional em `/etc/mcaster/secrets.env` e reinicie o serviço.
- **O stream está vermelho, não há bitrate.** Os pacotes do grupo não chegam à máquina: outra VLAN, um pedido IGMP que o switch não repassa, um firewall cortando o UDP, grupo ou porta errados. Verifique a partir desta máquina — é ela quem assina.
- **Há bitrate, mas não há faixas.** O stream chega, mas não é esse serviço: o grupo carrega um multiplex e é preciso um número de programa.
- **O push diz «Enviando» e o receptor não vê nada.** Os pacotes saem para a rede errada ou morrem num roteador: defina a **Interface de saída** e aumente o **TTL de multicast**.
- **O push está no estado «Erro».** O motivo está escrito ali mesmo, ao lado do status: conexão recusada, timeout, rejeição do receptor.

3.2.8 O que fazer a seguir

- [Adicionar uma entrada reserva](#) — uma segunda fonte e comutação automática.
- [Definir o PNR, os PIDs e o SDT](#) — para que o receptor veja o serviço sob o número certo.
- [Obter um CBR](#) — quando o sinal vai para um modulador.
- [Configurar o monitoramento](#) — alertas para uma entrega que parou.

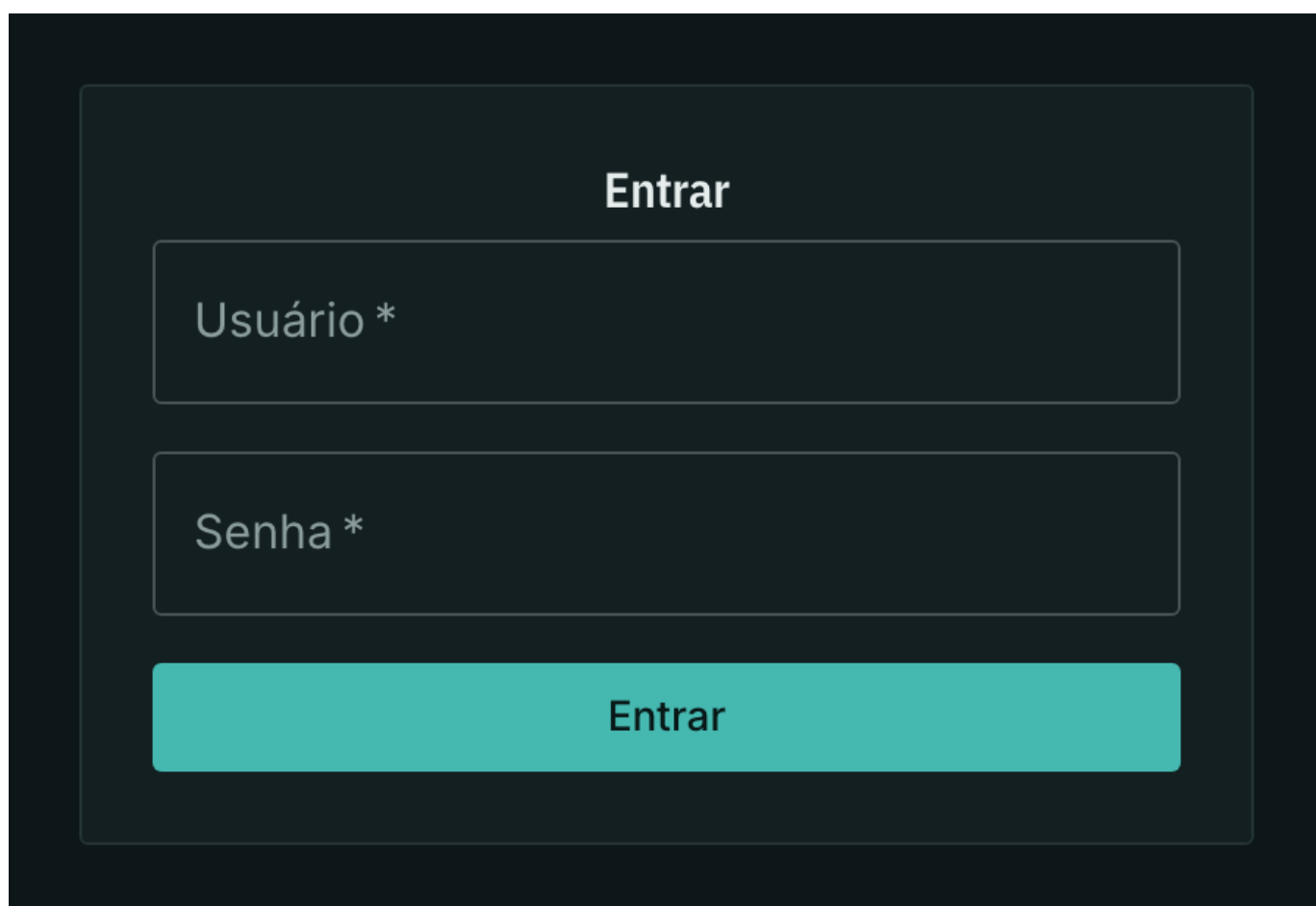
3.3 Administradores

O acesso ao painel do Mcaster funciona por contas nominativas: quem abre o painel tem seu próprio login, sua própria senha e sua própria função. A máquina guarda essa lista sozinha — é a sua própria lista de funcionários e não se sobrepõe a nada mais.

Os administradores, em particular, não são espectadores. As sessões de reprodução em **Sessões** são as pessoas que assistem aos canais, e nada têm a ver com o acesso ao painel. Não se deve misturar as duas listas: elas significam coisas diferentes, vivem por tempos diferentes e um erro numa custa algo completamente diferente do que um erro na outra.

3.3.1 O primeiro acesso

Uma máquina recém-instalada ainda não conhece administradores — não há quem os crie. Existe exatamente uma forma de entrar: as credenciais estáticas que o pacote gerou durante a instalação. O login é `admin`, a senha é o valor de `EDIT_AUTH_PASSWORD` em `/etc/mcaster/secrets.env` (o arquivo é legível apenas pelo root; veja [Instalação](#)).

A screenshot of a login interface. It features a dark background with a central light gray box. Inside this box, the word "Entrar" is displayed at the top. Below it are two input fields: the first is labeled "Usuário *" and the second is labeled "Senha *". At the bottom of the box is a large teal button with the text "Entrar" in white.

O painel aceita essas credenciais e pede de imediato uma conta nominativa:

Crie o primeiro administrador

Você entrou com credenciais estáticas. Crie uma conta nominal para trabalhar com seu próprio login.

Pelo menos 8 caracteres

Criar

[Continuar com credenciais estáticas](#)

A conta criada aqui recebe a função **superadmin** — é a única função que gerencia os demais administradores. De agora em diante, trabalhe sob ela, e não sob as credenciais estáticas. O motivo é simples: as credenciais estáticas são compartilhadas e não respondem à pergunta de quem fez o quê.

O link **Continuar com credenciais estáticas** pula a etapa. A oferta volta no próximo acesso — o painel continua lembrando enquanto a lista de administradores estiver vazia.

3.3.2 As credenciais estáticas são a entrada de emergência

As credenciais estáticas não desaparecem quando já existem administradores. Isso é deliberado: elas permanecem como entrada de emergência para quando as contas nominativas não ajudam — o único superadmin saiu da empresa, esqueceu a senha ou teve a conta desativada por engano. Entrando com elas, sempre é possível criar um novo superadmin.

Trate-as como trata a chave da sala de servidores:

- a senha vive em `/etc/mcaster/secrets.env`, e apenas o root pode ler o arquivo;
- ela é trocada nesse mesmo arquivo, e a máquina pega o novo valor quando o serviço é reiniciado;
- quem precisa de acesso cotidiano recebe uma conta nominativa, em vez de receber esta senha.

3.3.3 A seção «Administradores»

A seção é visível apenas para um superadmin. Para as demais funções ela não está no menu nem é alcançável por link direto — a recusa vem do servidor, e o menu apenas para de oferecer o que a função de qualquer forma não receberia.

LOGIN	FUNÇÃO	ESTADO	RÓTULOS DE CONTEÚDO
anna	superadmin	Ativo	Editar Excluir

A lista tem três colunas, que respondem a três perguntas:

- **Login** — o nome com o qual a pessoa entra.
- **Função** — o que ela pode fazer.
- **Estado** — **Ativo**, **Senha temporária** (a conta existe, mas seu dono ainda não trocou a senha que recebeu) ou **Desativado** (a conta está lá, o acesso não).

3.3.4 Funções

Um administrador tem exatamente uma função, escolhida entre três:

- **superadmin** — tudo, incluindo esta seção e a substituição do documento de licença.
- **admin** — todo o trabalho cotidiano: streams, modelos, máquinas do cluster, o arquivo, a proteção de reprodução. Exatamente duas coisas ficam fechadas: a seção de administradores e a substituição da licença.
- **viewer** — apenas observação: o painel de controle e as sessões. A configuração fica fechada por inteiro, leitura incluída.

O viewer merece uma observação, porque proibir a **leitura** da configuração parece excesso. Não é: os documentos de configuração carregam segredos por construção — senhas de fontes de terceiros, chaves, credenciais de máquinas. Um documento não pode ser entregue com os segredos cortados no caminho, portanto não é entregue a um viewer de forma alguma.

O servidor verifica a função em cada requisição, não apenas no login.

3.3.5 Adicionar um administrador

O botão **Adicionar administrador** abre o formulário:

Adicionar administrador

Login
oleg

Senha
● ● ● ● ● ● ● ● ● ●

Pelo menos 8 caracteres

Repita a senha
● ● ● ● ● ● ● ● ● ●

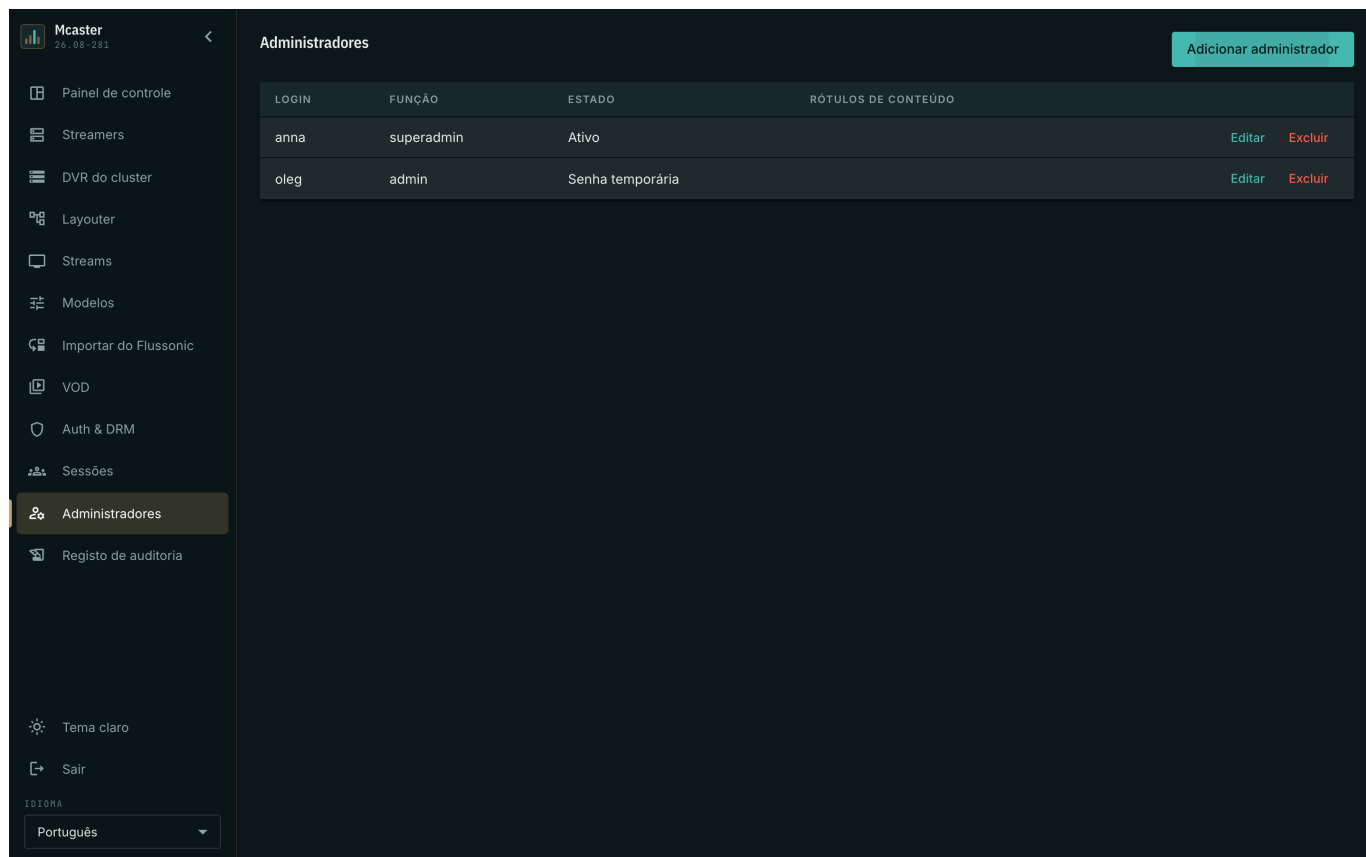
Função
admin ▼

☒ Exigir troca de senha no primeiro acesso

Cancelar Criar

Digite aqui uma senha temporária e entregue-a à pessoa do jeito que você entrega senhas. Marque a caixa **Exigir troca de senha no primeiro acesso**: uma senha inventada no lugar de outra pessoa também é do conhecimento de quem a inventou — ou seja, deixou de ser um segredo só dela.

A conta nova aparece na lista no estado **Senha temporária**:



The screenshot shows the 'Administradores' (Administrators) page in the Mcaster system. The left sidebar contains a menu with options like 'Painel de controle', 'Streamers', 'DVR do cluster', 'Layouter', 'Streams', 'Modelos', 'Importar do Flussonic', 'VOD', 'Auth & DRM', 'Sessões', 'Administradores' (highlighted), and 'Registo de auditoria'. At the bottom of the sidebar are settings for 'Tema claro' and 'Sair', and a language dropdown set to 'Português'. The main content area is titled 'Administradores' and features a table with columns: LOGIN, FUNÇÃO, ESTADO, and RÓTULOS DE CONTEÚDO. There are two rows of administrators: 'anna' (superadmin, Ativo) and 'oleg' (admin, Senha temporária). Each row has 'Editar' and 'Excluir' links. A green button 'Adicionar administrador' is in the top right corner.

LOGIN	FUNÇÃO	ESTADO	RÓTULOS DE CONTEÚDO
anna	superadmin	Ativo	Editar Excluir
oleg	admin	Senha temporária	Editar Excluir

No primeiro acesso, o dono dessa conta cai não no painel, mas no formulário de troca de senha:

Altere sua senha

Esta conta foi criada com uma senha temporária.
Altere-a para continuar.

Senha atual

Nova senha

Pelo menos 8 caracteres

Repita a nova senha

Alterar senha

Nada mais se abre até a senha ser trocada — é o servidor recusando, não uma janela com cara de severa: conhecer os endereços não ajuda a contornar o formulário. Trocada a senha, a pessoa já cai no painel de imediato, sem um segundo login.

3.3.6 Editar, redefinir senha e desativar o acesso

O botão **Editar** numa linha abre tudo o que se pode mudar numa conta existente:



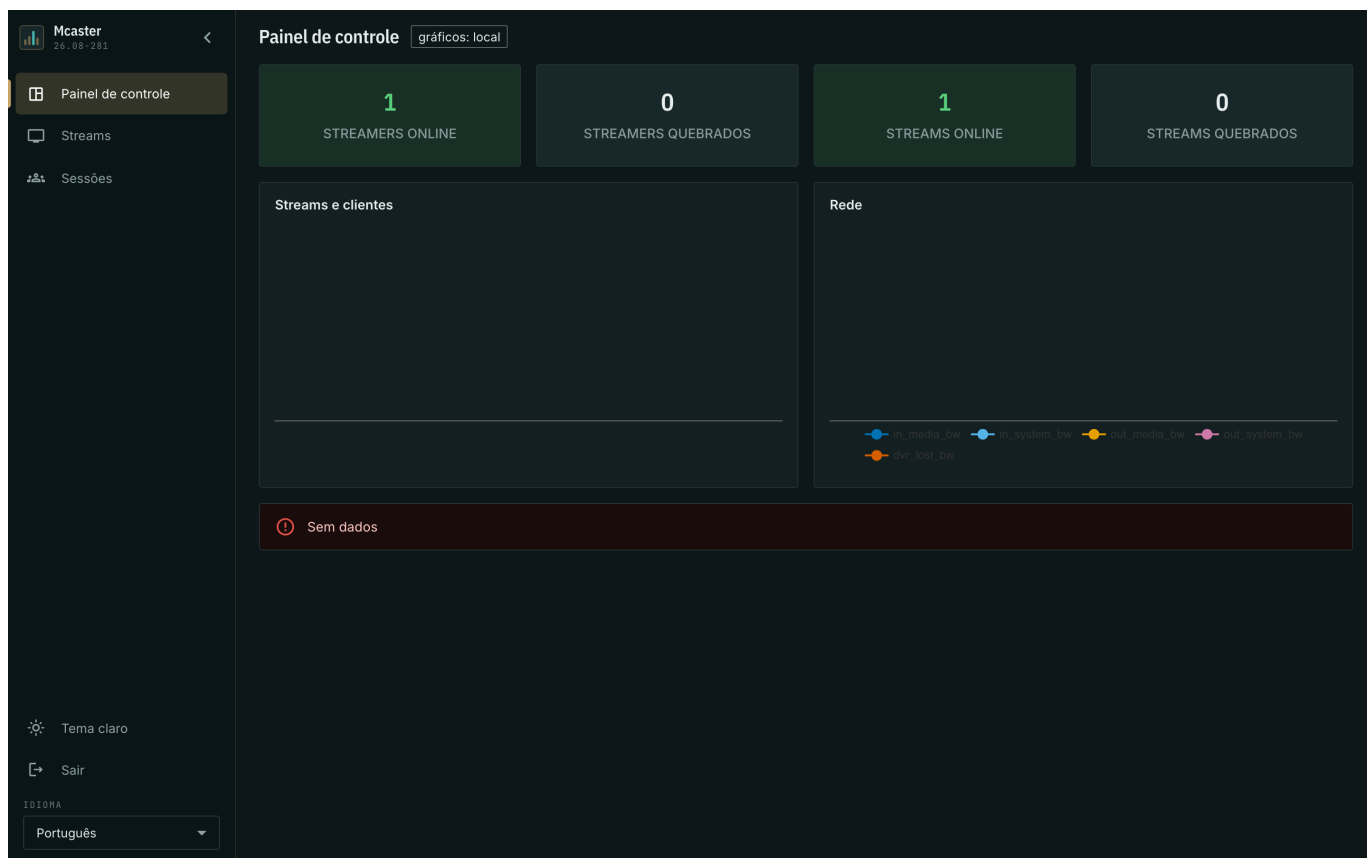
- **Função** — a nova função entra em vigor imediatamente: as sessões da conta são derrubadas e a pessoa entra de novo com os novos direitos.
- **Nova senha** — é assim que se traz de volta quem esqueceu a sua. O painel não vai mostrar a ninguém a própria senha, por isso uma senha esquecida não é recuperada, e sim substituída. A senha emitida aqui também é temporária: seu dono a troca no próximo acesso. Campo vazio significa «deixar a senha como está».
- **Nota** — para que serve esta conta e de quem ela é. Começa a valer a pena quando já há uma dúzia delas.
- **Desativado** — o acesso é fechado na hora, junto com qualquer sessão já aberta, enquanto a conta em si permanece.

LOGIN	FUNÇÃO	ESTADO	RÓTULOS DE CONTEÚDO
anna	superadmin	Ativo	Editar Excluir
nina	viewer	Ativo	Editar Excluir
oleg	admin	Senha temporária	Editar Excluir

Desativar é o que se usa na maior parte do tempo: alguém saiu de férias, mudou de função dentro da empresa, perdeu o notebook. O botão **Excluir** apaga a conta para sempre; exclua quando a conta foi criada por engano.

3.3.7 O que um viewer vê

Assim fica o mesmo painel sob a função **viewer**: no menu restam apenas o painel de controle e as sessões, e as seções de configuração não são mostradas.



É uma função conveniente para o plantão e para quem precisa ver o que está acontecendo sem mudar nada.

3.3.8 A sessão no navegador

O painel não guarda a senha no navegador. Depois do login, ele funciona com um passe emitido pelo servidor, que ele mesmo renova enquanto a aba está aberta — não se pede a senha de novo ao longo do dia de trabalho.

O botão **Sair** derruba a sessão no servidor, e não apenas a aba. Isso importa ao sair da máquina de outra pessoa: fechar uma aba não tira o acesso; sair, tira.

4. Captura de sinal

4.1 Captura de multicast e MPTS

Um fluxo de transporte por UDP é a entrada principal da cabeça de rede. O sinal chega como um serviço único em um grupo (SPTS) ou como um multiplex no qual o programa de que você precisa é um entre uma dúzia (MPTS). O Mcaster aceita os dois casos e, no segundo, recebe o tronco **uma única vez** para todos os seus programas, sejam quantos streams você montar a partir dele.

4.1.1 Conecte o grupo

Crie um stream e pressione **+ Adicionar fonte** na aba **Fontes**. O protocolo é **UDP MPEGTS**:

- **Host** — o endereço do grupo multicast, por exemplo `239.255.1.77`, ou `0.0.0.0` para unicast em uma porta;
- **Porta** — a porta do grupo;
- **Tempo limite do peer (ms)** — quanto silêncio tolerar antes de reiniciar a captura; campo vazio significa o padrão.

Pressione **Salvar**. Se no campo **Host** houver um grupo multicast, a estação se inscreve nele por conta própria (IGMP para IPv4, MLD para IPv6). O unicast difere só no endereço: `0.0.0.0` significa «escutar a porta em todas as interfaces».

4.1.2 Um programa de um tronco MPTS

Um multiplex de um satélite ou de um parceiro carrega vários serviços em um mesmo grupo. O necessário é escolhido pelo campo **Programa (PNR)** da mesma fonte: crie um stream por serviço, todos com o mesmo grupo e porta e com números de programa diferentes.

O campo vazio significa «a primeira com conteúdo» — é o que o formulário diz abaixo dele.

O que importa aqui na prática:

- **O tronco é recebido uma única vez.** Dois streams no mesmo grupo e porta dividem o socket e o demultiplexador: a rede vê uma única inscrição e a máquina faz uma única análise do transporte. Dez programas de um tronco não custam dez capturas.
- **Sem número indicado, toma-se o primeiro programa que realmente carrega mídia.** Não o primeiro pelo PSI: as tabelas nos troncos costumam estar desatualizadas e anunciam serviços cujos PIDs foram cortados há tempos.
- **Um número ilegível é recusado pelo formulário**, e não entendido como «o tronco inteiro»: de outro modo o stream viraria uma salada de todos os programas do multiplex.
- **Os contadores são atribuídos por assinante**, enquanto os erros do tronco como um todo são duplicados para todos: por stream vê-se tanto os seus próprios PIDs quanto o problema ocorrido com o transporte inteiro.

4.1.3 Captura de um remetente específico (SSM)

Quando várias fontes transmitem em um grupo e só uma delas interessa, coloque o endereço dela no campo **Origem (SSM)**.

O kernel se inscreve no grupo indicando o remetente (`IP_ADD_SOURCE_MEMBERSHIP`, IGMPv3) e descarta o tráfego alheio antes que ele chegue à estação. Suportado para IPv4.

4.1.4 Captura por HTTP

O mesmo fluxo de transporte servido por HTTP é a forma habitual de pegar o sinal de um parceiro quando o multicast não circula entre vocês. O protocolo da fonte é **HTTP MPEGTS**, com uma **URL** e, se necessário, um **Tempo limite do peer (ms)**.

A transferência chunked é suportada. Uma conexão rompida e uma conexão viva mas silenciosa são distinguidas e aparecem de forma diferente no status da entrada — são dois diagnósticos diferentes: «fomos desconectados» e «pararam de enviar».

Uma entrada HTTP tem o buffer de entrada ligado por padrão (30 segundos): uma fonte assim quase sempre chega em rajadas e, sem o buffer, as rajadas viajavam para a saída.

4.1.5 O que é analisado do stream

Vídeo H264, HEVC, MPEG-2; áudio AAC, AC3, EAC3, MPEG-2 audio; teletexto DVB e legendas DVB com seus descritores, legendas CEA-608/708, metadados KLV, marcadores SCTE-35, descritores de idioma das trilhas do PMT.

Os marcadores SCTE-35 são recebidos e sobrevivem ao trânsito até a saída. Processá-los — substituir, compensar, inserir — ainda não existe no Mcasterr.

4.1.6 O que deve estar correto na rede

- **O multicast não viaja pela internet.** A estação deve estar no mesmo segmento que a fonte do grupo, ou a rede entre elas deve rotear multicast.
- **Quem se inscreve é a estação.** A requisição IGMP sai dela, e os switches com IGMP snooping entregam o stream apenas aos inscritos. Não há nada a configurar na estação para isso — o importante é que a rede deixe passar a inscrição.
- **Verifique na máquina da estação.** O `tcpdump -i <iface> host 239.255.1.77` executado nela responde à pergunta «os pacotes chegam?», e nenhuma verificação feita de uma máquina vizinha a substitui.

4.1.7 Aplicação das mudanças

A troca de grupo, porta, número de programa ou URL reinicia a fonte. A troca de apenas um tempo limite é aplicada em tempo real, sem interromper a captura.

4.1.8 O que vem a seguir

- [Adicione uma entrada reserva](#) — uma segunda fonte e as regras de comutação.
- [Erros de entrada](#) — o que os contadores significam quando o sinal degrada.
- [Devolva o sinal para a rede.](#)

4.2 Captura por SRT e RIST

SRT e RIST levam o sinal por uma rede que você não controla: pela internet, entre cidades, de uma unidade móvel. As perdas numa rede assim são inevitáveis, e os dois transportes as recuperam por retransmissão enquanto houver tempo para isso.

4.2.1 SRT

Na aba **Fontes** pressione **+ Adicionar fonte** e escolha o protocolo **SRT**. A estação aguarda a conexão em uma porta UDP dedicada e o remetente se conecta a ela.

Campo	O que define
Host, Porta	o endereço e a porta UDP em que se aguarda a conexão
Senha de acesso	a chave de criptografia AES; na divergência, a conexão é recusada no handshake
Stream ID	o identificador esperado do remetente
Latencia de recepcao (ms)	o orçamento de recuperação de perdas (veja abaixo); 120 por padrão
Tempo limite de handshake (ms), Tempo limite do peer (ms)	os tempos limite de estabelecimento e de inatividade

Um endereço `srt://` colado inteiro é dividido nos campos, e a **URL SRT** remontada é mostrada ao lado — é ela que se copia e se entrega ao remetente.

4.2.2 A latência é um orçamento, não um atraso por atrasar

A **Latencia de recepcao** define quanto tempo um pacote espera no buffer de recepção antes de ser entregue. Um pacote perdido só consegue voltar a tempo se o pedido de retransmissão e a resposta couberem nessa janela — daí a regra prática escrita na própria dica do campo: **não menos que três ou quatro RTTs** até a fonte.

Num enlace com RTT de 100 ms, os 120 ms padrão não deixam espaço para nenhuma tentativa de recuperação: uma perda vira um buraco no sinal. O preço de aumentá-la é honesto e previsível — exatamente esse atraso na saída e memória para o buffer de recepção.

O valor é escolhido a partir do RTT medido, e não da sensação: o RTT e a sua variação estão nos [contadores da conexão](#).

4.2.3 O que o transporte faz

O mecanismo ARQ completo está implementado: confirmações (ACK/ACKACK), pedidos de retransmissão (NAK) com período ajustado ao RTT medido, entrega em ordem a partir do buffer de latência, descarte de pacotes irremediavelmente atrasados (too-late drop) e keepalive para conexões ociosas.

A consequência prática para uma cabeça de rede: **as perdas no enlace não precisam virar defeitos no sinal**. Enquanto o orçamento basta, o buraco é fechado por retransmissão e nunca chega à saída. Quando o orçamento não basta, os contadores dizem — crescem os descartes por atraso, e não só as retransmissões.

4.2.4 RIST

O protocolo **RIST** é adicionado com o mesmo **+ Adicionar fonte**. Há duas diferenças em relação ao SRT, e as duas aparecem no formulário:

- **Perfil RIST — Simple (par de portas)**, em que a mídia e o RTCP ocupam portas vizinhas (a porta de mídia deve ser par), ou **Main (tunel GRE)**, em que pelo cabo passa uma única **Porta do tunel**. O tunel também tem um **Papel no tunel** — conectar para fora ou aguardar — e uma **Senha PSK** que criptografa o conteúdo; ela deve coincidir nos dois lados.
- **Buffer de recepcao (ms)** — o mesmo, em significado, orçamento de recuperação da latência do SRT; 1000 por padrão. Ao lado estão **Secao de reordenacao (ms)**, **Numero de tentativas** e a **Interface de multicast** para receber um grupo.

4.2.5 Aplicação das mudanças

A troca de endereço, porta, chave ou identificador reinicia a fonte. A troca apenas dos tempos limite é aplicada em tempo real.

4.2.6 O que vem a seguir

- [Envie o sinal adiante por SRT](#) — para um destino fixo ou para um parceiro puxar.
- [Adicione uma entrada reserva](#) — SRT em par com um multicast.
- [Erros de entrada](#) — como distinguir uma fonte morta de um enlace que degrada.

4.3 Publicação SRT

A **captura por SRT** funciona assim: você diz de antemão à estação onde esperar o sinal, e o remetente deve discar exatamente para lá. A publicação inverte o esquema — um correspondente com OBS ou um codificador móvel disca para o cluster por conta própria, com um único endereço e uma única porta, sem saber nem escolher de quais máquinas o cluster é feito. A captura sobe no nó que atendeu a chamada.

É assim que se conectam autores externos: um repórter com um telefone, o estúdio de um parceiro, uma câmera em um evento. Eles recebem uma única linha de conexão — do resto o cluster cuida sozinho.

4.3.1 Como o cluster atende a chamada

- A porta UDP dedicada do stream é escutada por **todos os nós do cluster** — disque para qual for, a porta é a mesma.
- Quem liga passa pela verificação da **senha de acesso** no próprio handshake; depois o stream fica fixado ao nó que atendeu a chamada e arranca ali.
- Uma publicação caída espera o seu retorno: após alguns segundos de silêncio a conexão se fecha e o stream volta à espera. Um codificador que redisca para outro nó leva o stream consigo — a fixação segue a chamada.
- Enquanto a publicação está no ar, a porta está ocupada: um segundo codificador que discar para a mesma porta recebe uma recusa de «ocupado» e não derruba a transmissão em curso.

As fontes são opcionais para esse stream: ele pode viver só da publicação e arrancar com a primeira chamada. Uma fonte configurada ao lado da publicação atua de reserva — o ao vivo a supera.

4.3.2 Um endereço para todos os codificadores

O codificador precisa de um único endereço, e o cluster tem vários nós. O ajuste **Endereço de publicação** (menu **Configurações do central**) é o nome ou o IP que o console coloca na linha de conexão. O que está por trás desse endereço — um nome DNS com vários registros A, um VIP ou um balanceador UDP — é decisão da instalação; esta página cobre a variante DNS como a que não exige software adicional. O ajuste vazio significa o host do console e só serve para uma máquina única.

Publicação SRT
Um nome público ou IP ao qual os codificadores se conectam. O que está por trás (DNS com vários registros A, VIP, balanceador UDP) é decisão da instalação. Vazio — o host do console (caixa única).

Endereço de publicação
publish.lan

Endereços visíveis dos nós — candidatos aos registros DNS do nome:
192.168.28.240 — node1
192.168.28.242 — node3
192.168.28.241 — node2

Salvar Cancelar

4.3.3 Cenário: um nome DNS com vários registros A

No SRT não há redirecionamentos: o cliente disca por UDP para um endereço concreto, e o protocolo não sabe dizer «pergunte no nó ao lado». O que os clientes SRT — OBS, Larix, VLC — sabem fazer é percorrer os registros A: quando um endereço não responde, eles tentam os registros do nome por conta própria, gastando uns três segundos por cada um morto. Um nome com um registro A por nó dá tolerância a falhas pelo esforço do próprio cliente: um nó morre — o codificador disca para o seguinte.

A ordem de configuração:

1. **Declare os endereços públicos dos nós.** Na ficha de cada nó preencha o **Endereço público de mídia** — o campo oferece a escolha entre os endereços que o nó vê nas suas interfaces; atrás de NAT o correto só você sabe, escreva-o à mão.
2. **Confira a dica.** Sob o ajuste **Endereço de publicação** o console lista os candidatos a registros DNS — um endereço por nó vivo, com o nome do nó. Essa é exatamente a lista do que vai para a zona.
3. **Crie os registros A.** Na sua zona DNS crie um nome — por exemplo, `publish.example.com` — com um registro A para o endereço de cada nó da lista.
4. **Coloque o nome em «Endereço de publicação»** e salve. O cluster resolve o nome periodicamente e confere o resultado com o registro de nós: junto ao ajuste se vê que o nome leva a nós vivos. Um nome que não se resolve, ou que leva só a nós desligados, acende um aviso — o erro clássico de manutenção «nó retirado, registro esquecido» aparece de imediato, não na primeira transmissão quebrada. A partir desse momento as URL do codificador de todos os streams passam a ser montadas com o nome.

Duas ressalvas honestas:

- Vários registros A são **tolerância a falhas, não balanceamento de carga**: os resolvores cacheiam e embaralham os registros do seu jeito, e os codificadores de uma mesma rede podem cair todos no mesmo nó. Um desvio de fixações entre os nós é um quadro normal.
- **Nem todo cliente móvel percorre os registros.** Alguns aplicativos pegam o primeiro endereço e insistem nele. Para esses é mais seguro configurar dois endereços no perfil do codificador — principal e reserva.

4.3.4 Configuração do stream

Abra o stream na aba **Streams** e encontre a seção **Publicação SRT**:

Campo	O que define
Porta UDP	a porta de publicação; uma porta — um stream, em todos os nós ao mesmo tempo
Senha de acesso	a chave de criptografia AES, de 10 a 79 bytes; na divergência, a chamada é recusada no handshake

A senha de acesso é opcional, mas uma porta sem ela é uma porta aberta: poderá publicar qualquer um que disar. Deixe-a vazia só em uma rede de confiança — o formulário avisa sobre isso.

Publicação SRT

Ingestão em uma porta UDP dedicada: o codificador recebe uma única linha — endereço e passphrase. As fontes são opcionais; uma fonte ao lado da publicação atua como reserva.

Porta UDP
9003

Passphrase

Remover a publicação

Opcional, 10-79 bytes; ASCII recomendado

 A porta está aberta sem passphrase: sem criptografia e sem tranca, qualquer um que conectar pode publicar. Apenas para redes confiáveis.

SRT
Larix
Moblin



URL do codificador
srt://publish.lan:9003

Abaixo dos campos monta-se a **URL do codificador** pronta — `srt://endereço:porta?passphrase=...` — com botão de copiar e código QR: o endereço com a chave entram no codificador móvel pela câmera, não pelo teclado. O endereço da linha é o mesmo **Endereço de publicação** da seção acima: configure-o antes de distribuir a linha, senão o codificador leva o endereço de um nó concreto.

Ao lado do QR estão as abas **Larix** e **Moblin**: a mesma porta como link de importação pronto para esses aplicativos — escaneie com a câmera deles e a conexão aparece sozinha nas configurações.

Publicação SRT

Ingestão em uma porta UDP dedicada: o codificador recebe uma única linha — endereço e passphrase. As fontes são opcionais; uma fonte ao lado da publicação atua como reserva.

Porta UDP

9001

Passphrase

korovamoloko-live

Remover a publicação

Opcional, 10-79 bytes; ASCII recomendado

SRT

Larix

Moblin



URL do codificador

srt://publish.lan:9001?passphrase=korovamoloko-live

📄

4.3.5 Conexão do codificador

No OBS: **Configurações → Transmissão → Serviço: Personalizado...**, e a linha inteira da ficha do stream no campo **Servidor**:

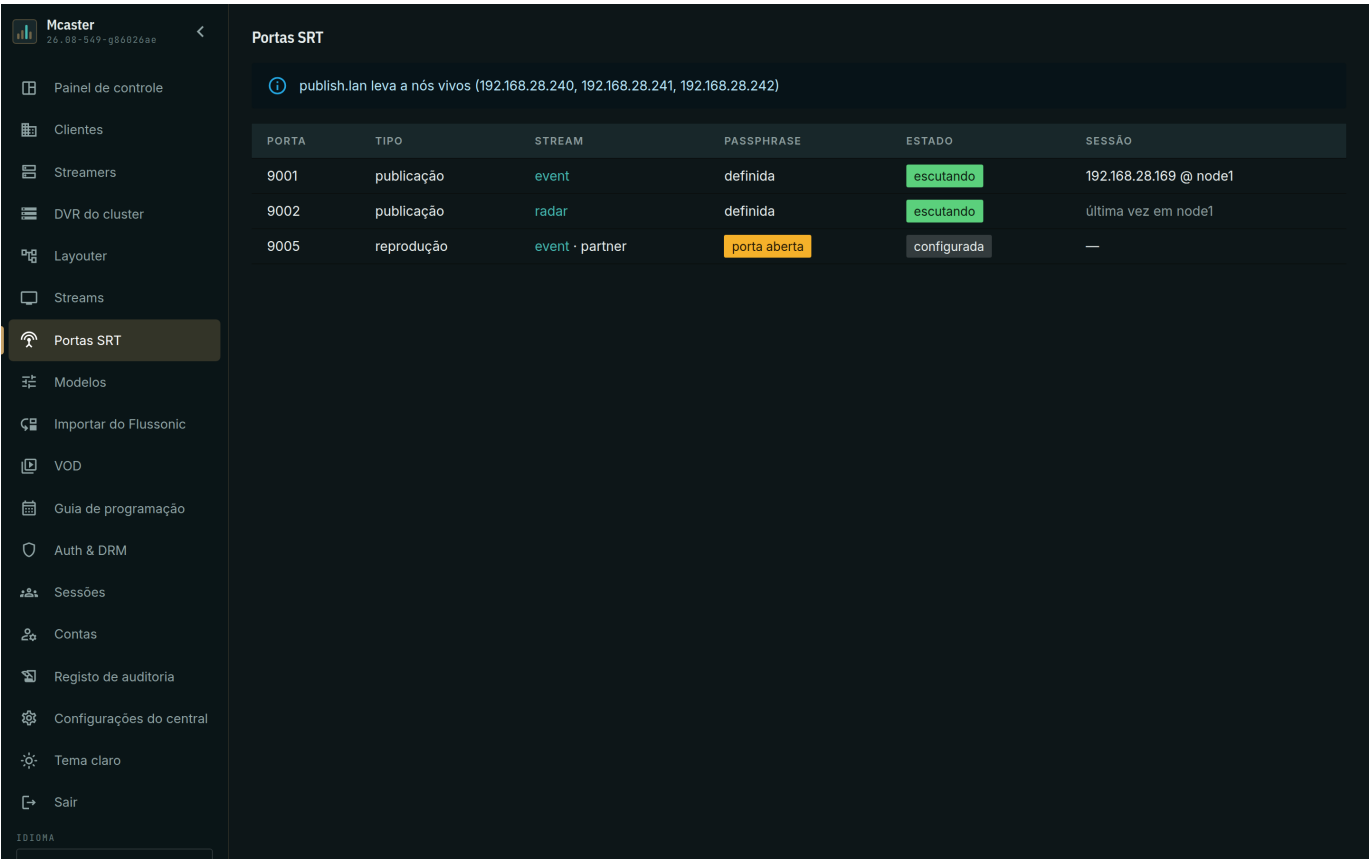
```
srt://publish.example.com:9000?passphrase=sua-chave
```

Ao codificador móvel mostre o QR da mesma seção. Depois de iniciar a transmissão, o stream sobe em um par de segundos — visível na lista de streams e na página de portas.

4.3.6 O resumo de portas SRT

A página **Portas SRT** (no menu, sob **Streams**) é o mapa de todas as portas SRT do cluster: portas de publicação e de [entrega](#), de qual stream é cada uma, se tem senha de acesso, se de fato está escutando. Também mostra:

- **a escolha de porta para um stream novo** — os números ocupados estão à vista, uma porta duplicada fica marcada;
- **uma publicação ao vivo** — o endereço do codificador e o nó que atendeu a chamada;
- **a fixação entre chamadas** — para um stream cujo codificador caiu e ainda não voltou, o nó onde o stream espera: uma chamada para ele retoma sem consultar o central, uma chamada para outro nó leva o stream para lá;
- **a verificação do endereço de publicação** — a mesma linha de estado que junto ao ajuste.



4.3.7 Quando uma chamada é recusada

O que aconteceu	O que o codificador vê
A senha de acesso não coincidiu	uma recusa no handshake, sem conexão
A porta está ocupada por uma publicação em curso	uma recusa de «ocupado»; a transmissão em andamento não é interrompida
A publicação ficou em silêncio	após alguns segundos de silêncio a conexão se fecha e o stream espera a próxima chamada

4.3.8 O que vem depois

- [Captura por SRT e RIST](#) — o esquema clássico quando o endereço da fonte é conhecido de antemão.
- [Entrega por SRT](#) — passar o sinal recebido adiante.
- [Monitoramento](#) — contadores de conexão e qualidade da captura.

4.4 Recepção de ST 2110

ST 2110 é o sinal de estúdio transportado por uma rede IP: vídeo não comprimido, áudio e dados auxiliares viajam em **grupos multicast separados**, amarrados por um relógio PTP comum. O Mcaster recebe esse sinal como uma fonte de stream comum: as trilhas entram na cadeia, são transcodificadas, gravadas e entregues pelos mesmos meios que um sinal de satélite ou uma fonte SRT.

Uma coisa distingue o 2110 de qualquer outra entrada, e a configuração começa por ela: **aqui a fonte não é um endereço, e sim um conjunto de trilhas**. Vídeo, áudio e dados auxiliares são três grupos diferentes com três conjuntos de parâmetros diferentes, e o stream é montado a partir deles, não de uma única URL.

4.4.1 O que a rede precisa cumprir

Um fluxo não comprimido são gigabits por segundo, e a rede é preparada para isso antes de alguém abrir o painel.

- **Banda por fluxo.** 1080p60 com 10 bits são cerca de 2,5 Gbit/s e 260 mil pacotes por segundo. Uma interface de 10 Gbit/s aguenta três fluxos desses, não quatro: o restante é folga para as rajadas.
- **Quem escolhe a porta é o operador, não a estação.** A inscrição sai pela interface indicada nas configurações da entrada, e é ali que o switch entrega o grupo. Uma porta indicada sem banda suficiente é o diagnóstico mais comum: o fluxo «chega» enquanto a imagem se desfaz, porque quem descarta pacotes é o switch — e os contadores da própria máquina continuam limpos.
- **PTP é obrigatório.** O relógio do 2110 vem do domínio PTP, e a estação concilia a própria escala de tempo com ele. Um relógio desviado aparece no status do stream separadamente dos erros de recepção: não são a mesma coisa.
- **Multicast e IGMPv3.** Os grupos são entregues pela inscrição da estação; quando vários emissores alimentam um grupo e você precisa de um deles, o endereço dele vai no campo **Origem (SSM)**.

4.4.2 Conecte as trilhas

Crie um stream, abra a aba **Fontes** e clique em **+ Adicionar fonte**. O protocolo é **ST 2110**.

Protocolo
ST 2110

Interface
dpdk0

Endereçamento
Controlador (IS-05)

Nome (dpdk0) ou o seu endereço; não existe o valor «qualquer»

O controlador traz os endereços por IS-05. À mão é o mesmo feito por você: o endereço e o formato de cada trilha são definidos por você, para uma rede sem controlador e para testes.

☐ Reduzir 10 bits para 8 na recepção

Economiza banda até o transcodificador ao custo dos dois bits baixos; desligado por padrão.

Receptores

Vídeo, áudio e dados auxiliares são anunciados pela própria entrada: o controlador vê três receptores e traz os endereços.

Vídeo
☒ Receber

Comutado pelo controlador: 5e647b64-f5f8-56a8-b9ba-9eb4adccae11

Ajuste desta faixa

Áudio
☒ Receber

Comutado pelo controlador: 9f746893-d917-5944-8f80-bfba7292e210

Ajuste desta faixa

Dados auxiliares
☒ Receber

Comutado pelo controlador: 9784a111-18f4-5b59-a402-0a44f8c58c69

Ajuste desta faixa

Os campos de toda a entrada:

- **Interface** — o nome da porta (`dpdk0`) ou o endereço dela. Aqui não existe o valor «qualquer», e isso é proposital: a interface decide por onde o switch entrega o grupo.
- **Endereçamento** — quem dá os endereços às trilhas. É o assunto da seção seguinte.
- **Reduzir 10 bits para 8 na recepção** — economiza banda até o transcodificador ao custo dos dois bits baixos. Desligado por padrão.

Os **receptores** são anunciados pela própria entrada: vídeo, áudio e dados auxiliares aparecem sempre, e o papel deles não é perguntado. O controlador vê exatamente esses três receptores e traz os endereços para eles. Uma trilha assim não está na configuração enquanto não houver nada a dizer sobre ela.

Pode haver mais trilhas de um mesmo tipo: uma segunda trilha de áudio para outro idioma é criada com **Adicionar faixa** no bloco **Faixas adicionadas**. Ali o papel (`áudio 2`) é escolhido por você, e o tipo precisa ser nomeado explicitamente — de uma string livre ele não se deduz. A edição do papel é aplicada quando o campo perde o foco.

Cada trilha tem a caixa **Receber**: desmarcá-la mantém a trilha na configuração com todos os parâmetros, mas nenhum socket é aberto.

Não esqueça de **Salvar**: a edição da entrada chega ao servidor apenas pelo botão.

4.4.3 Duas formas de endereçar uma trilha

A forma é escolhida no campo **Endereçamento** e vale para a entrada inteira, não para uma trilha isolada: uma rede ou é governada por um controlador ou não é, e uma trilha comutada pelo controlador ao lado de outra digitada à mão não é um cenário, e sim um mal-entendido que alguém teria de depurar.

- **Controlador (IS-05)** — o caminho principal. O operador liga a entrada e nomeia a interface; de onde receber é o controlador que diz, e para todas as trilhas de uma vez.
- **À mão** — o mesmo feito por você: o operador digita o endereço e o formato de cada trilha. Serve para uma rede sem controlador e para testes.

O arquivo de transporte não está nessa lista, e isso é proposital. O SDP é enviado pelo controlador, e descreve uma conexão já feita em vez de ser um jeito de estabelecê-la: ninguém escreve SDP à mão. Uma trilha comutada traz a linha **Comutado pelo controlador** com o identificador do emissor, e abaixo dela o **Arquivo de transporte**, somente leitura. Isso é diagnóstico — a resposta a «de onde isto está chegando agora» — e não um campo de entrada.

O endereçamento à mão fica assim:

Interface
dpdk0

Endereçamento
À mão

Nome (dpdk0) ou o seu endereço; não existe o valor «qualquer»

O controlador traz os endereços por IS-05. À mão é o mesmo feito por você: o endereço e o formato de cada trilha são definidos por você, para uma rede sem controlador e para testes.

☐ Reduzir 10 bits para 8 na recepção

Economiza banda até o transcodificador ao custo dos dois bits baixos; desligado por padrão.

Receptores

Vídeo, áudio e dados auxiliares são anunciados pela própria entrada: o controlador vê três receptores e traz os endereços.

Vídeo

☒ Receber

Grupo multicast
239.255.5.190

Porta
16394

Origem (SSM)
10.77.1.90

Endereço do grupo ao qual a trilha se inscreve

Porta RTP anunciada pelo emissor

Vazio aceita o grupo de qualquer um

O que há no fluxo. Um endereço sem formato é rejeitado: mídia não comprimida não carrega a própria geometria. A taxa de quadros é uma fração como no SDP: 60000/1001, não 59.94.

Largura
1920

Altura
1080

Profundidade de bits
10

Subamostragem
YCbCr 4:2:2

Taxa de quadros
60

☐ Entrelaçado ☐ Transporte por campos (segmented)

Ajuste desta faixa

Um endereço sem formato é rejeitado, e não é implicância: mídia não comprimida não carrega geometria própria dentro do fluxo. Um pacote 2110-20 não declara nem o tamanho do quadro, nem a taxa, nem a subamostragem — o receptor precisa saber disso de antemão, ou não há contra o que analisar as linhas. Daí o conjunto de campos: **Largura**, **Altura**, **Profundidade de bits**, **Subamostragem**, **Taxa de quadros**, **Entrelaçado**, **Transporte por campos (segmented)**.

A taxa é escrita como fração, como no SDP: 60000/1001, não 59.94. O arredondamento aqui não é um detalhe — a partir dele não se reconstrói a grade de marcas de tempo.

Dos esquemas de subamostragem, o que hoje chega à cadeia é **YCbCr 4:2:2** com 8 ou 10 bits. Os demais — 4:2:0, 4:4:4, RGB — são analisados e distribuídos em grupos de pixels, mas deles não se monta quadro algum: a trilha para e o motivo nomeia o esquema e a profundidade.

Os dois campos de um quadro entrelaçado a própria estação junta; o desentrelaçamento continua sendo trabalho do transcodificador.

4.4.4 Comutação por um controlador NMOS

O endereçamento à mão é o caminho reserva. A estação faz o que a infraestrutura de estúdio espera dela: anuncia-se a um controlador por **IS-04** e entrega a ele a comutação dos seus receptores por **IS-05**. O controlador escreve ele mesmo o arquivo de transporte na trilha — a mesma edição de configuração que um operador teria feito.

O anúncio é configurado nas configurações do nó, seção **NMOS**:

NMOS

Como este nó se anuncia a um controlador IS-04. Sem esta seção o nó não aparece no registro, e os endereços dos receptores 2110 são definidos à mão no próprio stream.

Interface dpgk0	Identidade	Rótulo Mcaster head-end
Porta anunciada; vazio significa NMOS desligado	Dela derivam os identificadores; vazio significa o nome do host	Aparece no controlador; vazio significa o nome do host

Onde este nó serve as interfaces NMOS

☒ Porta e interface próprias (recomendado) ☐ Porta HTTP compartilhada do nó

Interface das interfaces...	Porta das interf...
Rede de controle; vazio significa a interface de mídia	Padrão 8090

A ativação IS-05 é servida por quem é proprietário da configuração: o cluster para um nó gerenciado, a própria caixa se for autônoma. Ela escuta na mesma porta que o resto, portanto aqui não há nada a configurar.

- **Interface** — a porta anunciada. Um campo vazio significa que o NMOS está desligado e o nó nem sequer aparece no registro.
- **Identidade** — dela derivam os identificadores dos recursos; vazio significa o nome do host. Se você a mudar, todos os identificadores do nó mudam e o controlador vê um dispositivo novo.
- **Rótulo** — o que aparece no controlador.
- **Onde este nó serve as interfaces NMOS** — **Porta e interface próprias** (recomendado) ou **Porta HTTP compartilhada do nó**. A própria é definida por **Interface das interfaces NMOS** (a rede de controle; vazio significa a interface de mídia) e **Porta das interfaces NMOS** (8090 por padrão).

A escolha é sobre acesso, não sobre conveniência: a ativação IS-05 não é autenticada, e na porta compartilhada as interfaces aparecem em todas as portas HTTP do nó, inclusive as públicas. Só a rede pode limitar quem chega até elas — por isso a recomendação é uma porta separada na rede de controle.

A ativação é servida por quem é proprietário da configuração: para um nó gerenciado por um cluster, o próprio cluster (nas configurações dele é a seção **NMOS (ST 2110)** com a caixa **Comutar entradas pelo central**); para uma caixa autônoma, ela mesma. Do lado do nó não há nada a configurar para isso.

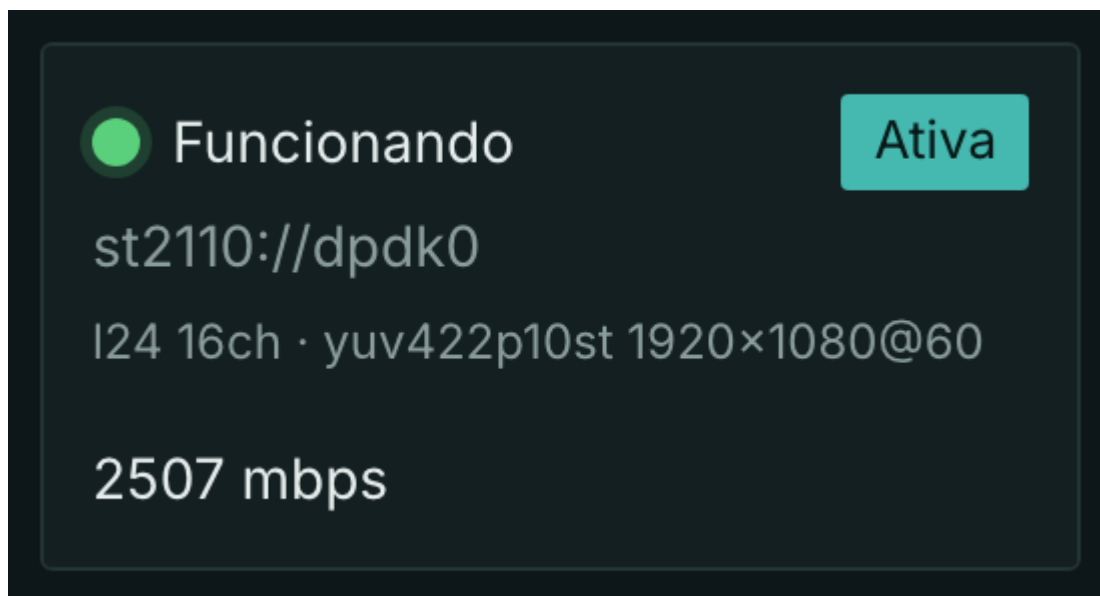
Sobre os receptores importa uma coisa: **um nó nunca tem receptores sobrando**. O controlador vê exatamente aqueles derivados dos streams, e uma ativação não cria stream algum. Por isso o stream a ser comutado é criado de antemão, com uma entrada ST 2110 no modo **Controlador (IS-05)**, onde as trilhas ainda não têm endereços. Enquanto o stream não existir, o controlador não tem o que comutar.

Uma trilha comutada diz isso no formulário: **Comutado pelo controlador** com o identificador do emissor. Essa linha é preenchida pelo controlador e não pode ser editada.

A estação procura um registro por conta própria e, ao encontrá-lo, registra-se nele e mantém a inscrição viva com um batimento. Se não houver registro na rede, ela fala diretamente com o controlador, ponto a ponto.

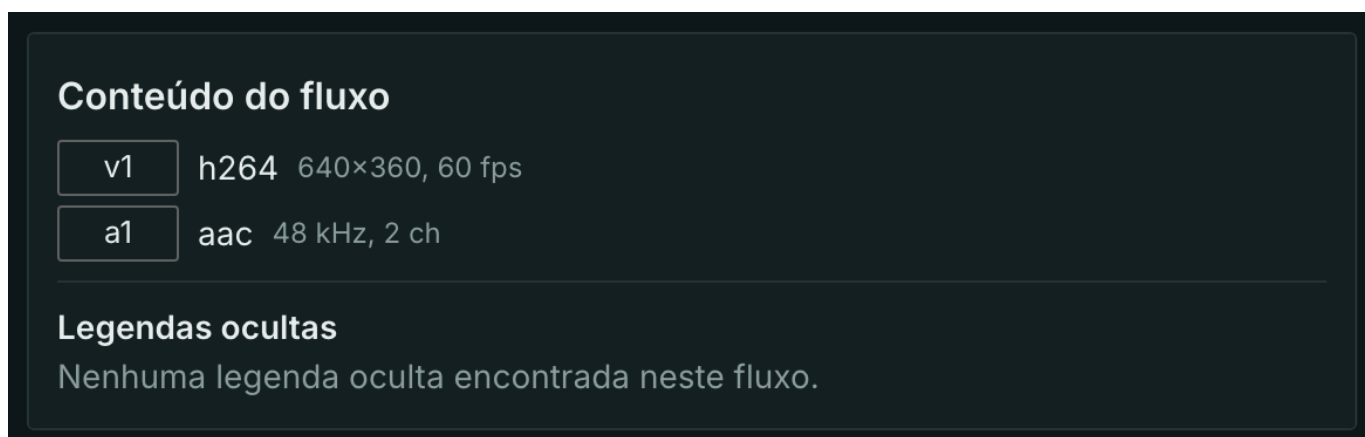
4.4.5 O que chega ao stream

O stream começa assim que a primeira trilha passa a chegar e não espera pelas demais. O estado da recepção fica na aba **Fontes**, no bloco **Saúde das fontes**:



O cartão mostra o status, o endereço de entrada, o conjunto de trilhas e o bitrate de recepção. Os números da captura são reais: 2501 Mbit/s é uma trilha de vídeo 1080p60 de 10 bits mais dezesseis canais de áudio L24.

O conjunto de trilhas fica na aba **Configurações básicas**, no bloco **Conteúdo do fluxo**:



O codec nomeia a disposição no cabo, e não aquilo em que o fluxo vai se tornar: `yuv422p10st` para o vídeo de 10 bits e `l24` para o áudio. A estação não faz downmix de canais — pega tantos quantos o emissor declarar e os entrega como uma única trilha.

4.4.6 Um fluxo não comprimido precisa de transcodificação

Um stream que não tem nada além de uma entrada 2110 é um estado de funcionamento, não uma falha: ele arranca, recebe dados e passa quadros aos assinantes. Mas nele não há segmentação, e **a playlist mestre chega sem uma única trilha**: uma trilha não comprimida não pode entrar nela — um player quebraria analisando o codec dela em vez de receber um honesto «não há nada para reproduzir».

Por isso o painel oferece um link ao lado do conjunto de trilhas — **Transcodificar para reproduzir na web**: ele abre a aba **Transcodificador** com um perfil pronto. Depois disso o stream reproduz, grava no arquivo e alimenta os envios como qualquer outro.

Os indicadores de cadeia não comprimida e de segmentação inativa aparecem separadamente no status do stream, para que esse estado não seja confundido com um defeito.

As miniaturas de um fluxo não comprimido funcionam sem transcodificador: são tiradas diretamente da trilha não comprimida, inclusive para o arquivo.

4.4.7 Dados auxiliares e legendas

Uma trilha do tipo **dados auxiliares** (2110-40) é recebida e analisada em pares DID/SDID. O que acontece depois depende do que sejam esses dados:

- **O teletexto OP-47** chega ao stream como uma trilha de teletexto — a mesma que o teletexto do MPEG-TS produz.
- **As legendas CEA-708/608** são analisadas e o contador do par DID/SDID delas cresce, mas nenhuma trilha de legendas aparece no stream: essas legendas vivem dentro do vídeo codificado, e entregá-las na saída ainda não faz parte da estação.
- **Os pares desconhecidos** são contados e descartados: não existe uma trilha de «dados auxiliares crus» saindo para fora.

Um pacote que não passa na soma de verificação é descartado com um contador; uma paridade quebrada é contada, mas não descarta o pacote.

4.4.8 Ajuste fino

Normalmente desnecessário: um campo vazio significa o padrão da estação, não «desligado». Ele vive em dois blocos recolhidos.

Ajuste desta faixa — o que pertence a um grupo:

- **Núcleo de CPU** — fixa a thread de recepção da trilha em um núcleo. Cada trilha tem a própria thread, por isso cada uma tem o próprio campo. Vazio deixa ao escalonador.
- **Fila AF_XDP** — a fila da placa que a trilha assume, contornando o socket comum. Vazio significa socket comum.

Ajuste da entrada — o que é comum:

- **Buffer do socket (bytes)** — quanto o kernel retém no buffer de recepção de cada socket.
- **Fila de quadros (bytes)** — o teto da fila de uma trilha entre a recepção e a cadeia. No estouro, o excedente é descartado com um contador: a recepção não trava.
- **Quadro de áudio (ms)** — a granularidade das mensagens de áudio. Menos significa menos atraso e mais mensagens.

4.4.9 O que é suportado

Capacidade	Estado
Vídeo 2110-20: YCbCr 4:2:2 com profundidade de 8 e 10 bits	sim
Os 10 bits são mantidos na disposição do cabo, sem redução	sim
Outros esquemas 2110-20: 4:2:0, 4:4:4, RGB	não, são analisados mas não geram quadro — o motivo nomeia o esquema
Entrelaçado: os dois campos são unidos em um quadro	sim
Transporte por campos (segmented)	a caixa é aceita mas ainda não afeta a recepção
Áudio 2110-30: L16 e L24, qualquer número de canais, <i>ptime</i> do SDP	sim
Dados auxiliares 2110-40: despacotamento, soma de verificação, contadores	sim
Teletexto OP-47 como trilha de teletexto	sim
Legendas CEA-708/608	analisadas e contadas, não entregues como trilha
Endereçamento por arquivo de transporte (SDP) enviado por um controlador	sim
Inscrição em um emissor específico (SSM, IGMPv3)	sim
Várias trilhas de um mesmo tipo em uma entrada	sim
NMOS IS-04: anúncio do nó, mDNS, registro em um registry	sim
NMOS IS-05: receptores comutados por um controlador	sim
A estação descobrir emissores alheios por conta própria	não, o endereço vem de um controlador ou de um operador
SDP com várias linhas de mídia em uma trilha	não, um bloco de endereçamento descreve uma trilha
Saída 2110 (a estação como emissora)	não

4.4.10 Aplicação das mudanças

Mudar o grupo, a porta, o emissor ou o arquivo de transporte recria o socket com uma inscrição nova — a recepção daquela trilha é interrompida enquanto as demais seguem. Os campos que não afetam a recepção são aplicados na hora.

4.4.11 Verificação

Verifique nesta ordem, porque cada passo só faz sentido depois que o anterior passou:

- **Os pacotes chegam** — `tcpdump -i <interface> host <grupo>` na própria máquina da estação. Uma verificação de uma máquina vizinha não responde a esta pergunta: a entrega do grupo segue a inscrição, e quem a fez foi a estação.
- **As trilhas subiram** — o cartão do bloco **Saúde das fontes** mostra o conjunto de trilhas e o bitrate de recepção.
- **O relógio não se desviou** — o desvio do relógio no status do stream. Sem sincronização o arquivo e o rebobinar param de funcionar, e a cura está no domínio PTP, não nas configurações da entrada.
- **A imagem é a esperada** — a prévia do stream: a geometria e a subamostragem são definidas à mão no endereçamento manual, e um erro ali dá linhas tortas em vez de uma recusa a receber.

4.4.12 Um script para uma bancada sem controlador

Uma bancada de testes normalmente não tem controlador NMOS, e ainda assim os endereços dos grupos são necessários: na maioria dos emissores eles mudam de uma ativação para outra, então anotá-los uma vez não funciona.

Para esse caso vai anexado um pequeno script — [nmosctl.py](#). Ele precisa apenas de Python 3; não tem dependências, e para encontrar nós é usado o `avahi-browse`.

```
python3 nmosctl.py discover           # nós na rede (mDNS)
python3 nmosctl.py list 10.0.0.90:8090 # dispositivos, emissores, receptores
python3 nmosctl.py enable 10.0.0.90:8090 <sender> # habilitar um emissor
python3 nmosctl.py sdp 10.0.0.90:8090 <sender>    # o arquivo de transporte dele
```

O texto que o `sdp` devolve é o que um controlador traz ao comutar uma trilha: na bancada ele serve para conferir o endereçamento à mão, e dele se copiam o grupo, a porta e o formato.

Os dois tropeços mais frequentes:

- **Enquanto um emissor não estiver habilitado não há arquivo de transporte** — o `sdp` responde com erro, porque o endereço do grupo aparece só na ativação.
- **O endereço muda a cada ativação.** Pegue-o de novo em vez de tirá-lo das suas anotações: uma inscrição no grupo anterior dá zero pacotes em silêncio.

4.4.13 O que vem a seguir

- [Transcodifique o fluxo](#) — o perfil depois do qual um sinal não comprimido reproduz e grava.
- [Adicione uma entrada reserva](#) — uma segunda fonte e as regras de comutação.
- [Erros de entrada](#) — o que os contadores significam quando um sinal se degrada.
- [Devolva o sinal à rede.](#)

4.5 Reserva de entradas

Uma cabeça de rede precisa permanecer no ar quando uma fonte morre. Para isso o stream recebe várias entradas: a estação conduz o ar com a viva de maior prioridade, vigia as demais e volta à principal quando ela reaparece.

O par típico numa estação é a recepção de satélite como entrada principal e o multicast de um parceiro como reserva; ou dois grupos de duas cabeças de rede.

4.5.1 Configure várias entradas

Na aba **Fontes** adicione uma segunda entrada com **+ Adicionar fonte**.

- **A ordem da lista é a prioridade:** quanto mais alta a entrada, mais importante ela é; a ordem muda com as setas **Mover para cima** e **Mover para baixo**. O campo **Prioridade** nas configurações da entrada sobrepõe a ordem da lista — o número menor é mais importante.
- **Defina o «Intervalo de reverificação da reserva (s)»** no bloco **Tempos limite de entrada**. Enquanto uma reserva não está no ar, a estação a sonda periodicamente, e na lista ela aparece como **Pronta** — verificada, só que não ativa. Sem reverificação, a saúde da reserva continua questão de fé até a própria falha, que é exatamente o momento em que descobrir já é tarde; 0 a desliga.

Os estados da entrada na lista são **Funcionando**, **Pronta**, **Iniciando**, **Degradada**, **Morta** e **Não verificada**.

4.5.2 Regras de morte e degradação

Cada entrada tem uma placa **Opções da fonte**; campos vazios herdam as regras comuns do stream do bloco **Tempos limite de entrada**. Assim os limiares são definidos uma vez por stream e sobrepostos um a um, enquanto a placa recolhida mostra com chips o que exatamente está definido.

Os campos estão reunidos em três grupos, pelo que a estação faz quando uma regra é violada.

- **Quando a fonte está morta** — falhas duras, comutação imediata: **Tempo limite da fonte (s)** (nenhum quadro), **Tempo limite de vídeo** (sem vídeo), **Bitrate máximo** (bitrate acima do teto — proteção contra lixo na entrada).
- **Quando a fonte está degradada** — falhas suaves, comutação só se existir uma fonte melhor: **Tempo limite de áudio** (o áudio sumiu enquanto o vídeo vive), **Bitrate mínimo (kbit)** (bitrate abaixo do piso), **Limite de erros CC** (violações de continuidade MPEG-TS dentro da janela limpa; o campo existe só para protocolos de transporte — UDP, HTTP MPEGTS, SRT, RIST).
- **Recuperação e novas tentativas** — **Janela limpa (s)**, **Tempo máximo de nova tentativa** e **Prioridade**.

A degradação é sobre a qualidade do sinal, e sair de uma entrada degradada para uma morta não faz sentido. Por isso uma regra suave não derruba o ar, apenas rebaixa a entrada na fila. O motivo de uma marca a estação nomeia direto: «erros CC acima do limite», «bitrate abaixo do piso», «sem áudio», «sem quadros».

4.5.3 A janela limpa

A **janela limpa** é a duração de um trecho sem nenhuma violação. Uma entrada conta como funcionando se e somente se não tiver violado nenhuma regra por esses segundos, e a mesma janela é exigida de uma entrada revivida antes de voltar ao ar.

É a proteção contra o pêndulo: uma fonte que sobe por dez segundos e cai de novo não sacode mais o ar. O outro lado é honesto e previsível — o retorno à principal fica sempre adiado pelo comprimento da janela.

4.5.4 O que acontece no momento da comutação

- **Uma entrada fica ativa apenas quando está pronta.** Ela precisa entregar os parâmetros do stream e o primeiro quadro-chave; até lá a atual continua no ar. É exatamente por isso que a comutação não dá tela preta.
- **A emenda não é perfeita.** O quadro-chave da nova entrada é o ponto de emenda, e o receptor vai vê-lo: um quadro pode se repetir ou sumir. Não espere deste mecanismo uma comutação hitless de nível ST 2022-7.
- **A divergência de parâmetros é visível com antecedência.** A estação compara os parâmetros das entradas e mostra a divergência antes da falha — uma reserva com resolução diferente ou conjunto de trilhas diferente é notada cedo, e não no momento da comutação.
- **Uma entrada caída reconecta com uma pausa crescente,** e quando todas as entradas se esgotam, a rotação recomeça da primeira.

4.5.5 A vinheta

O bloco **Cartela quando todas as fontes caem** dá ao stream uma última linha de defesa: no campo **Caminho do arquivo no streamer** entra o caminho de um arquivo preparado, e o **Tempo limite (s)** decide após quantos segundos da queda da última entrada ele liga.

Para uma cabeça de rede isto não é cosmética: um multicast vazio na saída faz receptores e moduladores se comportarem de forma imprevisível, enquanto um sinal com vinheta continua sendo um sinal.

O arquivo deve existir nesse caminho em **cada** máquina para onde o stream possa migrar: a vinheta é tocada pela máquina em que o stream roda naquele momento.

4.5.6 O que olhar depois

Por stream vê-se quantos segundos ele viveu na entrada principal e quantos nas reservas, quanto tempo ficou sem dado algum, quantas comutações e tentativas houve. Uma fatia crescente de tempo na reserva significa que a entrada principal é sistematicamente ruim, mesmo que o ar nunca tenha sido interrompido.

Os erros detalhados por causa estão na página [Erros de entrada](#).

4.5.7 Se a reserva se comporta mal

- **O ar pula entre entradas.** Aumente a janela limpa: a entrada terá que provar saúde por mais tempo.
- **A reserva está no estado «Não verificada».** O intervalo de reavaliação não está definido — a estação não a visita.
- **A comutação parece lenta.** Reduza o tempo limite da fonte. Um limiar pequeno demais dispara em falso a qualquer solução de rede.
- **Não volta para a principal.** A principal ou ainda não reapareceu, ou não consegue cumprir a janela limpa — ou seja, está de fato doente.

4.6 Erros de entrada

Um único contador de erros responde só à pergunta «há um problema na entrada». Para uma cabeça de rede isso não basta: a causa decide a quem ligar. Pacotes perdidos são a rede, conexão recusada é o parceiro, tempo limite de handshake é o codificador dele ou o firewall entre vocês.

Por isso os erros de entrada são contados **por causa**, e cada causa é uma linha própria no gráfico e uma série própria nas métricas.

4.6.1 Duas famílias de causas

- **Defeitos de mídia de um stream vivo** — os dados chegam, mas danificados: `lost_packets`, `broken_payload`, `desync`, `ts_pat`.
- **Falhas da fonte** — não há dados, e a causa da morte é nomeada: `connection_closed`, `connection_refused`, `timeout`, `http_request_error`, `not_found`, `denied`, `decode_error`, `protocol_error`, `io_error`, `other`.

A diferença é prática. A primeira família significa que o sinal está correndo e precisa ser consertado no nível de transporte — o próximo lugar a olhar são os [contadores TR 101 290](#). A segunda significa que não há sinal algum, e o que precisa ser consertado é a conexão.

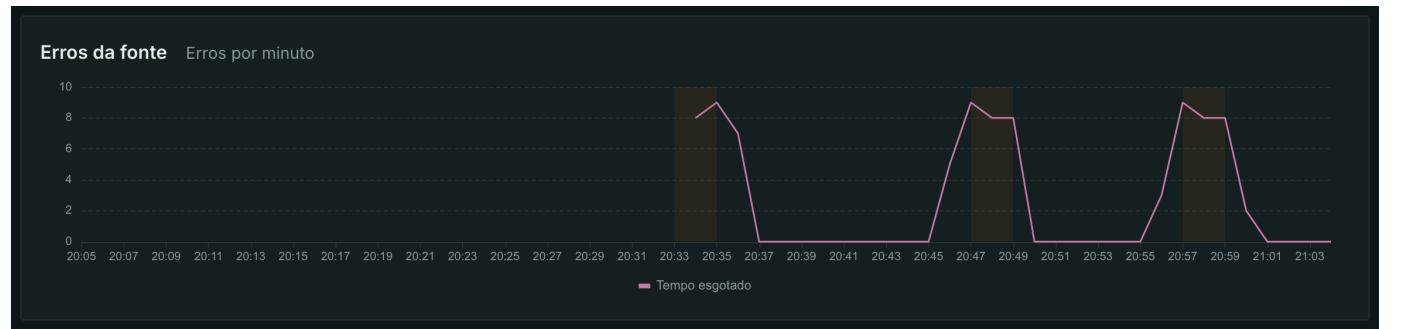
Como isto é contado:

- a classificação vem da categoria tipada do erro, e não do texto do log — reformular uma mensagem não quebra o contador;
- uma falha ao abrir uma fonte também entra no detalhe, não só a falha de uma conexão já em funcionamento;
- cada falha entra também no contador agregado `errors`, de modo que as somas conferem;
- as falhas repetidas nas tentativas são contadas a cada vez: a frequência de eventos é exatamente o sinal de «a fonte ainda está morta»;
- as paradas normais — reconfiguração, fim do stream, desalojo por uma entrada de maior prioridade — não contam como erros.

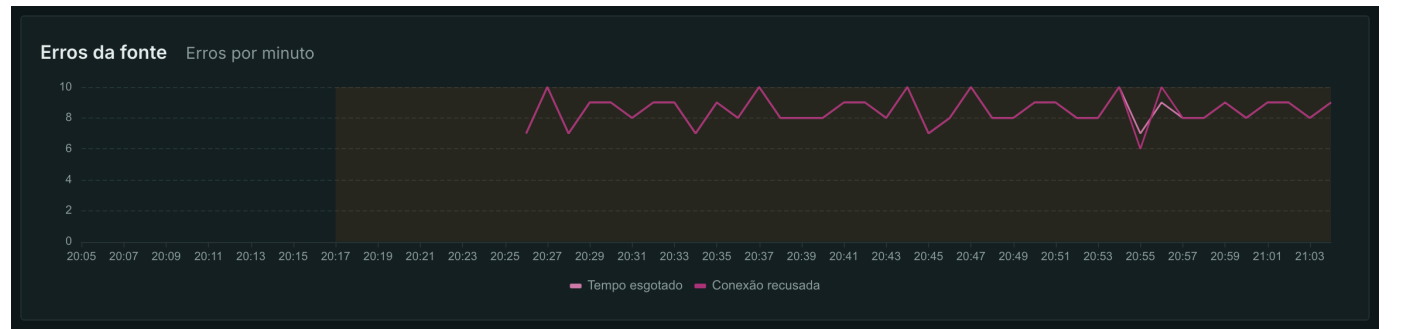
4.6.2 O gráfico no cartão do stream

O painel **Erros da fonte** na aba **Fontes** desenha os erros por minuto como linhas separadas por causa. Causas sem eventos não ocupam lugar na legenda — o gráfico mostra exatamente o que está acontecendo.

Os minutos em que o ritmo de quadros de entrada foi zero são destacados como outage. Isso responde a uma pergunta que um único contador de erros nunca fecha: se aqueles erros eram com o sinal vivo, ou se naquele momento não havia sinal algum.



Uma fonte HLS ou HTTP MPEG-TS inacessível dá `http_request_error` em cada tentativa, e um handshake SRT que falha por tempo limite dá `timeout`:



Sem a opção de contadores estendidos o painel mostra uma única linha agregada; as linhas detalhadas aparecem junto com a opção.

4.6.3 A série de métricas

No servidor Prometheus incorporado o detalhe viaja como uma série com o rótulo de causa — `errors_detail_total{name, cause}`; quando há mais de uma estação, as séries levam também `node`. Causas sem eventos não criam séries.

```
# stream errors over 5 minutes broken down by cause
sum by (cause) (increase(errors_detail_total{name="tv1"}[5m]))

# every input currently seeing connection failures
sum by (name) (rate(errors_detail_total{cause="connection_closed"}[5m])) > 0
```

4.6.4 Como ler

O que se vê	O que significa
<code>lost_packets</code> com bitrate vivo	quem perde é a rede entre a fonte e a estação; use os contadores de transporte para achar o PID afetado
<code>timeout</code> em rajadas, faixas de outage	a fonte não responde; cada rajada é mais uma tentativa
<code>http_request_error</code> como pente constante	a fonte HTTP está inacessível: endereço, DNS, firewall ou uma origem morta
<code>connection_closed</code> com o stream vivo	o parceiro mesmo derruba a conexão; olhe o lado dele
<code>denied</code> , <code>not_found</code>	acesso ou caminho: a chave, o <code>streamid</code> , o nome do recurso
<code>desync</code> , <code>broken_payload</code>	os dados chegam quebrados — o transporte ou o codificador do outro lado

Um erro isolado no gráfico é o normal numa rede. O diagnóstico vem da **forma**: um pente constante de uma causa, rajadas ao redor das faixas de outage, crescimento com o bitrate parado no mesmo nível.

4.6.5 O que vem a seguir

- [Reserva de entradas](#) — como transformar o diagnóstico numa comutação.
- [TR 101 290](#) — contadores de transporte por PID.
- [Monitoramento](#) — para onde esses contadores vão e como alertar sobre eles.

5. Processamento

5.1 Saída MPEG-TS

O receptor do outro lado não é um reproduztor: ele procura o serviço pelo número do programa, as trilhas pelos PIDs e o nome do serviço na SDT. Enquanto essas tabelas não forem as que ele espera, o sinal não existe para ele, por mais correto que seja o vídeo lá dentro.

Todas as saídas MPEG-TS de um stream — a entrega por HTTP, SRT, os destinos por UDP, SRT e RIST — são produzidas por **um único multiplexador**. A seção **Saída MPEG-TS** na aba **Saída** define as suas tabelas de serviço; a mesma seção num destino individual as sobrepõe apenas para esse destino.

5.1.1 As tabelas do stream

A seção diz isso mesmo no painel: ela vale para toda saída MPEG-TS do stream até que um push ou um SRT play a sobreponha. Campos vazios mostram os padrões como dicas.

Campo	O que define	Padrão
Número do programa (PNR)	a entrada na PAT, <code>program_number</code> na PMT e <code>service_id</code> na SDT	1
PID da tabela PMT	o PID da tabela PMT	auto
PID das trilhas	PIDs explícitos das trilhas: uma linha «Trilha» + «PID» por trilha	auto
Nome do serviço (SDT)	o nome do serviço na SDT	o nome do stream
Nome do provedor (SDT)	o nome do provedor na SDT	vazio
Transport stream ID	<code>transport_stream_id</code> na PAT e no cabeçalho da SDT	1
Original network ID	<code>original_network_id</code> no cabeçalho da SDT	1

As regras são verificadas ao salvar, não no ar:

- os **PIDs** estão na faixa 32–8190: abaixo disso é reservado às tabelas de serviço, acima, aos pacotes nulos;
- **PIDs duplicados** dentro da seção são proibidos, inclusive um PID de trilha igual ao PID da tabela PMT;
- o **número do programa** nunca é zero — o número zero na PAT é reservado à NIT;
- os **nomes do serviço e do provedor** juntos são limitados para que a SDT caiba em um pacote TS.

Um PID pode ser definido para uma trilha que agora não existe: ele vale quando a trilha aparece. Se um PID explícito colide com o PID atribuído automaticamente a outra trilha, o explícito vence e a trilha desalojada migra de forma determinística para um PID livre — com um aviso no log.

MPEG-TS output

Applies to every MPEG-TS output of the stream unless a push or SRT play overrides it

Program number (PNR)
1030

PMT PID
1000

Transport stream ID

Original network ID

Service name (SDT)
TV One

Provider name (SDT)
Example Media

Track PIDs

Track
a1

PID
1012

×

Track
v1

PID
1011

×

Track

PID

Add PID

5.1.2 SDT

Junto com a PAT e a PMT, a saída leva uma tabela SDT (PID 0x11): um serviço do tipo digital television, `service_id` igual ao número do programa, e os nomes do serviço e do provedor tirados da mesma seção.

As strings são codificadas conforme o DVB: o latino vai como está e todo o resto como UTF-8 com a codificação declarada, por isso nomes cirílicos são lidos corretamente pelos receptores.

Quando as configurações mudam, as versões das tabelas PAT/PMT/SDT afetadas são incrementadas e os receptores pegam as novas como devem. A edição é aplicada em tempo real: destinos de entrega e clientes conectados não reconectam.

5.1.3 Tabelas próprias por destino

Um destino individual tem uma seção **Substituição da saída MPEG-TS** com os mesmos campos — é a resposta a uma tarefa típica de cabeça de rede, em que o mesmo sinal é entregue a receptores diferentes sob números diferentes. Ela está disponível para:

- um **push** por UDP, RTP, SRT ou RIST — o RTMP não tem fluxo de transporte, por isso a seção não existe lá;
- a **distribuição por SRT** da porta dedicada do stream.

Campos vazios são herdados da seção do stream — os valores herdados aparecem como placeholders. Um valor definido no destino sobrepõe o do stream. A exceção é **PID das trilhas**: é um campo único, e um mapa não vazio do destino substitui inteiro o do stream — não há fusão por trilha.

A saída MPEG-TS por HTTP e o listener SRT compartilhado do servidor não têm substituição: eles sempre servem a saída compartilhada do stream.

MPEG-TS output override

Empty fields inherit the stream MPEG-TS section

Program number (PNR)

300

PMT PID

1000

Transport stream ID

Original network ID

Service name (SDT)

Provider name (SDT)

Track PIDs

Track	PID	
a1	221	×
v1	211	×
Track	PID	Add PID

5.1.4 Por que isto é barato

Para um destino com tabelas próprias não se inicia um segundo multiplexador. O destino recebe uma cópia da saída compartilhada na qual só as tabelas de serviço e os PIDs dos pacotes são reescritos: o arranjo dos pacotes no tempo, o PCR e os contadores de continuidade do multiplex compartilhado são preservados.

Daí seguem duas consequências práticas:

- as **propriedades CBR da saída** valem também para os destinos sobrepostos — reescrever as tabelas não quebra o bitrate constante;
- destinos com configurações efetivas idênticas dividem um stream byte a byte idêntico: cem destinos iguais custam tanto quanto um.

A versão de tabelas de uma saída reescrita é mantida pela sua própria história de mudanças, e não copiada da saída compartilhada.

5.1.5 O que vem a seguir

- **Entrega à rede** — para onde este stream vai.
- **CBR para DVB** — os requisitos do modulador quanto à taxa e ao PCR.
- **TR 101 290** — como verificar que as tabelas e a continuidade estão em ordem.

5.2 CBR para DVB

Um modulador não precisa de uma taxa média, precisa de **uniformidade no nível de pacotes TS**: a banda está alocada e o stream deve ocupá-la exatamente, sem quedas nem picos. O mesmo requisito vem dos receptores profissionais e das redes onde o transporte é contado por banda, e não por uma média por minuto.

O CBR no Mcast é uma propriedade do multiplexador de saída, não um botão separado de «ligar CBR».

5.2.1 O que o multiplexador faz

- **A taxa-alvo é derivada dos bitrates das trilhas** — com uma folga de cerca de 5% para o crescimento e um orçamento para as tabelas de serviço.
- **Os vazios são preenchidos com pacotes nulos (PID 0x1FFF)**: a banda fica ocupada de modo uniforme mesmo quando os quadros chegam em rajadas.
- **O PCR é posicionado a partir de um relógio sem deriva**, por isso a divergência entre PCR e DTS não se acumula nem em muitas horas de transmissão.
- **O modelo de buffer do decodificador (HRD) é levado em conta** — estouros e esvaziamentos são acompanhados em vez de descobertos no receptor.

A uniformidade é medida contra o relógio interno do stream (27 MHz), não contra o relógio de parede: o arranjo não depende de quão irregularmente os quadros chegam da fonte.

5.2.2 O que a uniformidade custa

As três exigências de um multiplex — arranjar os pacotes de modo uniforme, nunca travar, nunca perder nada — são satisfeitas duas a duas, mas não as três juntas. A uniformidade é uma afirmação sobre o futuro, enquanto os dados são um fato sobre o passado: para arranjar os pacotes uniformemente, o cronograma precisa conhecer o ritmo com antecedência, e o ritmo só pode ser medido pelo que já chegou.

Na prática, isso significa:

- **Um PID sem nada para enviar cede o seu lugar** em vez de parar o multiplex. Uma trilha de áudio anunciada na PMT mas silenciosa não deve travar o vídeo — e não trava.
- **Um PID atrasado se recupera dentro da janela** em vez de ser descartado. Um mux de broadcast clássico perde dados neste ponto; aqui a troca é feita ao contrário.
- **O ritmo é medido pelo PES empacotado**, não inferido da banda declarada multiplicada por uma constante.

O que daí segue para o operador: defina o bitrate de saída com honestidade, com folga para os picos, e não no limite do bit. Uma banda subestimada não é «qualidade um pouco pior», é um PID sistematicamente atrasado.

5.2.3 Configure as trilhas de saída

Abra o stream, a aba **Transcodificador**. Você pode começar de um preset — por exemplo **Escada padrão 1080/720/360 + AAC** — e editar os degraus a partir daí.

Num degrau do bloco **Vídeo (escada)**:

- **Bitrate (kbps)** — a taxa-alvo da trilha. A soma das trilhas mais as tabelas de serviço é a banda que o fluxo de transporte vai ocupar.
- **Comprimento do GOP e Estrutura do GOP** — para broadcast, o GOP costuma ser de exatamente um segundo: um ponto de entrada para o receptor e para a emenda. O botão **Unificar GOP** alinha em todos os degraus.
- **Codec** — com o valor **Como está**, o degrau não é recodificado. Se a fonte já chegou na forma certa, o multiplexador não precisa de transcodificador para arranjá-la de modo uniforme.
- **Altura, Rate control e Preset do codificador** — as demais propriedades do degrau.

O bloco **Codificação** escolhe quem faz o trabalho: **CPU** ou **GPU NVIDIA** (com uma placa escolhida, ou automaticamente a menos carregada).

Warning

Um degrau passthrough ao lado de degraus codificados degrada a escada: seu GOP e seus quadros-chave não estão alinhados com os demais degraus. O painel marca esse degrau à parte.

5.2.4 Desentrelaçamento

Uma entrada entrelaçada — SD MPEG-2, 1080i de broadcast — codificada em vídeo progressivo aparece como serrilhado. O **Desentrelaçamento** vive no bloco **Decodificador**, porque é uma propriedade da **entrada**, e não de um degrau: a decodificação acontece uma única vez e a imagem resolvida vai para todas as trilhas de saída.

Modo	O que faz
Automático (apenas quadros entrelaçados)	o padrão: o filtro é aplicado aos quadros que o decodificador marcou como entrelaçados; uma entrada progressiva passa direto sem custo
Forçado (todos os quadros)	para fontes que mentem sobre o entrelaçamento
Desativado	nenhum filtro é construído, os quadros vão para o codificador como estão

O desentrelaçamento não muda a taxa de quadros: um quadro progressivo por quadro de entrada.

5.2.5 Verificação

O que olhar na saída:

- **a taxa** no receptor — ela não deve flutuar; uma queda significa que o multiplex não tinha nada a enviar;
- **o jitter do PCR** e os **contadores de continuidade** — veja [TR 101 290](#); um stream uniforme com PCR incorreto não ajuda um modulador;
- **as tabelas de serviço** — o receptor deve ver o serviço sob o número esperado, veja [Saída MPEG-TS](#).

Sobrepôr as tabelas num destino individual **não quebra o CBR**: o destino recebe uma cópia da saída compartilhada com PIDs e tabelas reescritos, enquanto o arranjo dos pacotes e o PCR permanecem os do multiplex compartilhado.

5.2.6 O que vem a seguir

- [Saída MPEG-TS](#) — PNR, PIDs e SDT.
- [Entrega à rede](#) — como este stream chega ao modulador.
- [TR 101 290](#) — como verificar que a saída serve.

5.3 Legendas e teletexto

Um sinal de broadcast leva texto em duas formas: o teletexto europeu (legendas numa página, normalmente 888) e as legendas ocultas americanas CEA-608/708 dentro do vídeo. Uma cabeça de rede faz sobre elas duas perguntas diferentes: **elas sobrevivem até a saída e é precisa uma faixa separada.**

As duas vivem na aba **Processamento** do cartão do stream.

5.3.1 Trânsito por padrão

Por padrão, a estação não faz nada com o texto: o PID de teletexto e as legendas ocultas dentro do vídeo saem exatamente como chegaram. No formulário isso se lê assim:

- **Teletexto — Em trânsito dentro do MPEG-TS (sem faixa de legendas);**
- **Legendas ocultas, modo de entrega — Dentro do vídeo (passthrough).**

Para a entrega a uma rede, esse é o comportamento certo: o receptor e o televisor decodificam o texto por conta própria, e qualquer interferência seria um custo e um risco adicionais. O trânsito sobrevive ao remux e ao reempacotamento em outro transporte.

5.3.2 Quando a extração é necessária

A extração é necessária quando atrás da estação não está um receptor, e sim um reproduzidor: **nenhum reproduzidor OTT decodifica teletexto**. Se da estação sai uma **cópia OTT**, as legendas chegarão ao espectador apenas como faixa de texto.

- **Teletexto — Extrair as páginas de legendas como faixas:** cada página de legendas vira uma faixa WebVTT. Dentro do MPEG-TS o teletexto continua viajando: a extração adiciona uma faixa, não tira nenhuma.
- **Legendas ocultas — três modos: Dentro do vídeo (passthrough), Faixa de legendas no lugar (extract) e Dentro do vídeo e como faixa (both).**

O multilinguagem sai de graça: as emissoras costumam pôr uma página de legendas por idioma, e cada uma vira uma faixa separada, sem segunda entrada e sem trabalho manual.

O modo extract corta as legendas ocultas do MPEG-TS

No modo **Faixa de legendas no lugar (extract)**, as legendas ocultas são cortadas do vídeo em todas as saídas, MPEG-TS incluso: pushes por UDP, RTP e SRT, segmentos TS e saída por HTTP. Uma faixa de texto não tem como viajar dentro do MPEG-TS, então um consumidor do fluxo de transporte fica sem legendas de todo. Se o sinal é retirado por MPEG-TS — e numa cabeça de rede esse é o caso comum —, escolha **both**.

A extração de teletexto custa a baixa latência

Uma legenda só se conhece quando fecha: enquanto a página não muda, a duração da linha fica indecisa. Por isso um stream com faixa de texto é servido como segmentos comuns, sem segmentos parciais de LL-HLS. Para a entrega a uma rede isso não muda nada, mas alonga a borda ao vivo da cópia OTT em alguns segundos — mais um motivo para não ligar a extração onde o sinal apenas atravessa a estação.

5.3.3 O que se vê por stream

Sob o seletor, o painel lista as páginas de teletexto encontradas no stream. De cada uma são conhecidas duas coisas independentes, e não se deve confundi-las:

- **declarada** — a emissora declarou a página nas tabelas de serviço, com seu idioma e sua finalidade;
- **no ar** — a página chegou de fato no stream.

Daí os estados que você verá ao lado de uma página: **declarada mas nunca vista** (a emissora silencia), **no ar mas não declarada** (a emissora esqueceu de atualizar as tabelas — essas páginas são extraídas em igualdade de condições), ou as duas ao mesmo tempo.

O idioma é mostrado junto com a sua procedência, porque não merecem a mesma confiança: **declarado no PMT** é um fato, **herdado da faixa** é um palpite razoável a partir de uma faixa de idioma único.

A lista se preenche a partir do ar e só faz sentido num stream em execução.

5.3.4 O arquivo

As legendas ocultas são decodificadas e gravadas no [arquivo](#) em todos os modos, por isso o modo de entrega pode ser trocado depois — as legendas aparecerão também no arquivo já gravado. As faixas de teletexto extraídas vão parar no arquivo junto com o vídeo.

5.3.5 O que vem a seguir

- [A cópia OTT](#) — quem precisa da faixa de texto.
- [Saída MPEG-TS](#) — o que vai no fluxo de transporte.

6. Entrega

6.1 Entrega à rede

A saída principal de uma cabeça de rede é um fluxo de transporte enviado para a rede: para o grupo multicast de um operador de cabo, para um modulador, para um endereço unicast de um receptor. Aqui a estação é o lado ativo: ela envia sozinha, ninguém se conecta a ela.

Um destino de entrega se chama **push**, e todo push tem um **Nome**. O nome é a identidade dele: pelo nome o destino é encontrado nas configurações, nas estatísticas e nas métricas. O painel avisa disso bem abaixo do campo: trocar o nome cria um push novo e os contadores começam do zero.

6.1.1 Envie o stream para um grupo

A aba **Saída**, o bloco **Pushes**, o botão **+ Adicionar push**. O protocolo é **MPEG-TS/UDP**.

Campo	O que define	Padrão
Nome	a identidade do destino	—
Host	o grupo multicast ou o endereço unicast do receptor	—
Porta	a porta do receptor	—
TTL de multicast	por quantos roteadores o datagrama sobrevive	10
Interface de saída	por qual placa de rede o multicast sai	escolhe o sistema
Endereço de vinculação	o endereço local ao qual o socket está vinculado	escolhe o sistema

A chave **Ativado** ao lado do nome desliga um destino sem apagá-lo. O unicast difere do multicast só no endereço: TTL e escolha de interface funcionam do mesmo jeito.

6.1.2 Multicast numa rede real

Três coisas fazem a entrega tropeçar, e as três se resolvem com os campos acima.

- **A placa de rede errada.** Numa máquina com várias interfaces, o sistema escolhe a interface de saída pela tabela de rotas, e para multicast essa escolha muitas vezes não é a pretendida. Se o sinal «está sendo enviado» mas o receptor não o vê, defina primeiro a **Interface de saída** explicitamente.
- **TTL.** O padrão 10 deixa o pacote sobreviver ao roteamento — o 1 padrão do kernel não deixaria o stream sair nem da sub-rede. Através de vários roteadores o TTL sobe; para um segmento estritamente isolado, desce.
- **O receptor tem que se inscrever.** A estação envia para o grupo independente de alguém estar escutando: os pacotes saem sem nenhum inscrito. «Não chega nada» do lado do receptor costuma ser a inscrição dele mesmo ou o IGMP snooping no switch, e não a estação.

6.1.3 RTP em vez de TS puro

Alguns receptores esperam o MPEG-TS embalado em RTP conforme RFC 2250 / SMPTE ST 2022-2, e não um fluxo puro sobre UDP. É a mesma saída com outra encapsulação: escolha o protocolo **MPEG-TS/RTP** em vez de **MPEG-TS/UDP**.

O datagrama ganha um cabeçalho RTP; todo o resto — endereçamento, TTL, interface, contadores — é compartilhado com um push UDP comum.

A opção vizinha na mesma lista é **RIST**, para quando o lado que recebe espera exatamente isso.

6.1.4 Vários destinos de uma vez

Um stream tem quantos pushes precisar, e eles funcionam de forma independente: um receptor caído num destino não toca os demais.

Destinos com configurações de saída efetivas idênticas dividem **um único stream byte a byte idêntico**: cem pushes iguais custam tanto quanto um. O que os torna diferentes são apenas as suas próprias [tabelas de serviço](#) — o PNR e os PIDs próprios do destino; aí para ele as tabelas e os PIDs são reescritos, enquanto o arranjo dos pacotes, o PCR e os contadores de continuidade permanecem os da saída compartilhada.

6.1.5 Comportamento na falha

O pusher não tem política de novas tentativas própria — quem o reergue é o stream. Uma vez a cada cinco segundos o stream confere os destinos em execução com as configurações e troca um pusher morto por um novo; o contador de reconexões conta exatamente essas trocas.

Daí a consequência prática: **o envio a um destino inalcançável nunca desiste e nunca precisa de reinício manual**. Ele continuará voltando enquanto o destino estiver ativado.

Não contam como reconexões:

- **editar as configurações** — o pusher também reinicia, mas isso é decisão do operador, não uma falha do enlace;
- **desligar e ligar** — desligar não é apagar: as configurações e os contadores acumulados ficam, só o envio para.

Apagar um destino apaga também as suas estatísticas: os contadores estão ligados ao nome.

6.1.6 O que verificar

Sob cada push corre uma linha de estatística ao vivo: o status — **Enviando**, **Erro** ou **Desativado** —, a taxa em kbit/s, os erros por segundo, as reconexões e, num push falho, também o motivo. Todos os campos e as receitas de alertas estão na página [Monitoramento de entrega](#).

O status **Enviando** com taxa zero significa que não há nada a enviar: o problema está na entrada, não na saída.

6.1.7 O que vem a seguir

- [Saída MPEG-TS](#) — PNR, PIDs e SDT, inclusive os próprios de cada destino.
- [CBR para DVB](#) — quando o sinal vai para um modulador.
- [Monitoramento de entrega](#) — contadores e alertas.

6.2 Entrega por SRT

O SRT leva o sinal para onde o multicast não chega: para outra cidade, para um parceiro, pela internet. Há duas maneiras de entregá-lo, e elas diferem em **quem estabelece a conexão**:

- **Push** — a estação disca por conta própria até o destino e faz push do stream. É assim que se entrega um sinal a um site que espera uma conexão de entrada.
- **Distribuição (play)** — a estação levanta um listener e espera o parceiro puxar o stream. É cômodo quando a contraparte tem o seu próprio horário e o seu próprio equipamento: você dá o endereço e a chave, o resto é com ela.

6.2.1 Envio para um destino

A aba **Saída, + Adicionar push**, o protocolo **SRT**:

Campo	O que define
Nome	a identidade do destino
Host, Porta	o listener SRT do lado que recebe
Senha de acesso	a chave de criptografia AES; na divergência, o lado receptor recusa no handshake
Stream ID	o identificador pelo qual o receptor entende o que lhe foi enviado
Tempo limite de handshake (ms), Tempo limite do peer (ms)	os tempos limite de estabelecimento e de inatividade

Um endereço colado inteiro como `srt://203.0.113.10:9000` é dividido nos campos, e a **URL SRT** remontada é mostrada ao lado — é ela que se copia e se entrega ao lado receptor.

Um push morto se levanta sozinho, uma vez a cada cinco segundos, como qualquer outro [destino de entrega](#).

6.2.2 Distribuição da porta do stream

O bloco **Reprodução SRT**: ligue **Permitir pull SRT (reprodução)**, defina a **Porta** e, de preferência, uma **Senha de acesso**.

O parceiro puxa o stream conectando a `srt://<station-address>:<port>` com a chave dele. A porta pertence ao stream por inteiro — uma porta por stream; um conflito de portas entre streams é rejeitado ao salvar, e não descoberto no ar.

A distribuição de uma porta dedicada pode ter [a sua própria seção de saída MPEG-TS](#): o parceiro recebe o sinal sob o número de programa de que precisa, sem tocar os demais destinos.

6.2.3 O que convém saber

- **A saída não pinta o stream.** O status do stream reflete a entrada. Um destino de push caído não deixa o stream vermelho — a entrega é verificada pelos [contadores do destino](#).
- **A latência na saída é o orçamento do receptor.** Quanto tempo o lado receptor tem para recuperar perdas é ele quem define; a regra dos «três ou quatro RTTs» é a mesma nos dois lados — veja [Captura por SRT](#).
- **Retransmissões são normais enquanto são poucas.** O contador de retransmissões de um destino cresce quando o caminho perde pacotes: o transporte está fazendo o trabalho dele. O que preocupa não é o fato, mas a tendência e a companhia dela — reconexões ao lado.

6.2.4 O que vem a seguir

- [Monitoramento de entrega](#) — status, taxa, erros e retransmissões por destino.
- [Saída MPEG-TS](#) — um PNR e PIDs próprios para um parceiro específico.

- [Entrega à rede](#) — multicast e RTP.

6.3 A cópia OTT

Uma cabeça de rede não serve espectadores — esse é o trabalho do [Catena](#) ou da sua CDN. Mas muitas vezes o sinal precisa ser entregue ao mundo OTT direto da cabeça de rede: mostrar ao operador a imagem num navegador, dar uma cópia à transmissão pela internet, fazer push do ar para uma plataforma.

Esta página é sobre essa cópia e sobre os limites dela.

6.3.1 HLS e DASH da cabeça de rede

Qualquer stream da cabeça de rede está disponível em:

```
http://<station-address>/streaming/v/<stream>/index.m3u8
http://<station-address>/streaming/v/<stream>/Manifest.mpd
```

São servidos HLS (incluindo LL-HLS com segmentos parciais), DASH e MPEG-TS por HTTP. De fábrica, isso basta para ver o sinal num reprodutor e se certificar de que da cabeça de rede sai exatamente o que deve.

O que ter em mente:

- **Um reprodutor precisa dos seus próprios codecs.** Um navegador não lê um MPEG-2 de broadcast — é preciso [transcodificação](#) para H.264/AAC.
- **Um reprodutor precisa de legendas como pista.** Nenhum reprodutor OTT decodifica teletexto, veja [Legendas e teletexto](#).
- **Entrega em massa não é tarefa da cabeça de rede.** Balanceamento, zonas, autorização de espectadores e contabilidade de sessões começam onde a cabeça de rede termina.

6.3.2 Push por RTMP

Entregar o sinal a uma plataforma ou a um sistema de terceiros que espera uma conexão RTMP de entrada se faz com o mesmo push.

A aba **Saída, + Adicionar push**, o protocolo **RTMP**. Ele pede só um **Nome** e uma única **URL** inteira, com a chave de stream incluída: `rtmp://a.rtmp.example.com/live2/<key>`.

- São suportados H264+AAC, e também HEVC e AV1 em enhanced RTMP; streams só de áudio também saem.
- Os timestamps num push começam de zero — do jeito que as plataformas esperam.
- Os erros se distinguem no status: conexão recusada, tempo de conexão esgotado, publicação recusada (a chave está ocupada ou errada), um receptor travado.
- Não existe seção de [MPEG-TS de saída](#) para RTMP: esse protocolo não tem fluxo de transporte.

Como qualquer outro [destino de entrega](#), um push RTMP se levanta sozinho e mantém seus próprios [contadores](#): ele tem quadros, mas não tem datagramas nem retransmissões.

6.3.3 Criptografia da cópia OTT

A cópia OTT pode ser entregue criptografada. O bloco **Criptografia da entrega (DRM)** na aba **Entrega** do stream ativa o Common Encryption pelo esquema `cbs`: HLS e DASH saem como texto cifrado, e a playlist e o manifesto levam a sinalização para **Widevine**, **PlayReady**, **FairPlay** e **ClearKey**. A chave vem das configurações ou do seu servidor de chaves pelo SPEKE.

Quem criptografa é a estação — nesse arranjo ela é o origin. Os streamers que repassam a cópia dela recebem o texto cifrado já pronto e o entregam como está: não têm a chave e não têm a quem pedi-la.

O que isso muda na estação:

- **HLS e DASH são criptografados**, incluindo LL-HLS e a saída do arquivo.
- **MPEG-TS por HTTP recusa**. Não há criptografia para ele, e entregar esse stream aberto seria contornar a proteção: o pedido recebe um `403`.
- **Os pushes não são afetados**. Multicast, RTP, SRT e RTMP saem da estação abertos — criptografa-se a entrega aos reprodutores, não o transporte até o seu próprio equipamento.

Todo o caminho pode ser verificado sem um DRM comercial: deixe **ClearKey** como único sistema, informe uma chave nas configurações e abra a página do reprodutor — a licença é emitida pela própria estação. Esse modo não protege nada e serve apenas para verificar.

6.3.4 O que vem depois

- **Monitoramento de entrega** — se o push chegou até a plataforma.
- **Transcodificação** — levar o sinal ao que um reprodutor toca.
- **Catena** — quando atrás da cabeça de rede é preciso um serviço de televisão completo com espectadores.

6.4 Monitoramento de entrega

Uma cabeça de rede não pode perguntar ao receptor o que ele mediu. Tudo o que ela tem são seus próprios contadores de envio, e eles são coletados **por destino**.

A chave da estatística é o par «nome do stream, nome do destino», o mesmo das [configurações do destino](#). Isso não é formalidade: um stream sai para cinco destinos, e a frase «o stream está entregando 40 Mbit/s» não responde se ele chega a algum deles em particular.



A opção «estatísticas de pushes»

A vista de entrega é uma opção de licença à parte. Ela corta apenas os canais de dados: a seção `pushes` e os campos planos `push_*` da API, as séries `stream_push_*` do scrape e do servidor Prometheus embutido. A coleta em si nunca para, e os contadores sempre vão para a telemetria do Retroview — a análise no Retroview está disponível também sem a opção.

6.4.1 A vista no painel

Na aba **Saída**, abaixo das configurações de cada destino corre uma linha de estatísticas ao vivo: status, velocidade, erros por segundo, reconexões e, num destino que falhou, a causa ali mesmo na linha.

Nome
backup-dc

Protocolo
SRT

Ativado

Identifica o push: um nome novo cria outro push e os contadores começam do zero

Host
203.0.113.10

Porta
9000

Stream ID

Senha de acesso

Nome do host ou IP. Uma URL srt:// colada inteira é distribuída pelos campos.

URL SRT
srt://203.0.113.10:9000

Montada a partir dos campos acima: copie para um player ou um codificador.

Substituição da saída MPEG-TS

Campos vazios herdam a seção MPEG-TS do stream

Número do programa (PNR)

PID da tabela PMT

Transport stream ID

Original network ID

Nome do serviço (SDT)

Nome do provedor (SDT)

PID das trilhas

Trilha

PID

Adicionar PID

Desativado 0 kbit/s erros/s: 0.00 reconexões: 0

Nome
cdn-main

Protocolo
MPEG-TS/UDP

Ativado

Identifica o push: um nome novo cria outro push e os contadores começam do zero

Host
127.0.0.1

Porta
5500

TTL de multicast

Interface de saída

Endereço de vinculação

Substituição da saída MPEG-TS

Campos vazios herdam a seção MPEG-TS do stream

Número do programa (PNR)

PID da tabela PMT

Transport stream ID

Original network ID

Nome do serviço (SDT)

Nome do provedor (SDT)

PID das trilhas

Trilha

PID

Adicionar PID

Enviando 4230 kbit/s erros/s: 0.00 reconexões: 0

Nome
youtube

Protocolo
RTMP

Ativado

Identifica o push: um nome novo cria outro push e os contadores começam do zero

URL
rtmp://198.51.100.7/live/secret-key

Erro 0 kbit/s erros/s: 0.00 reconexões: 11 rtmp push youtube init failed: rtmp connect failed: timeout after 5s

Na lista de streams, um destino quebrado acende uma marca ao lado do status do stream: dá para ver que a entrega está quebrada sem abrir o stream.

McMaster

26-08-201

<

Painel de controle

Streamers

DVR do cluster

Layouter

Streams

Modelos

Importar do Flussonic

VOD

Auth & DRM

Sessões

Administradores

Registo de auditoria

Tema claro

Sair

Idioma

Português

Streams

Pesquisar por nome

Nome do novo stream *

Modelo

Criar

NOME DO STREAM	FONTES	STATUS	MODELO	TRANSCODIFICADOR	RESOLUÇÃO DE ENTRADA	TAXA DE BITS DE ENTRADA	DVR	ESPECTADORES
tv1		push		h264/aac	HD	167 mbps	-	—

1 na página

Anterior

Próxima

6.4.2 Contadores do destino

GET /streamer/api-v4/streams/stats/{name} traz um objeto indexado pelo nome do destino:

```

"pushes": {
  "cable-net": {
    "status": "running",
    "proto": "udp",
    "url": "udp://239.1.10.5000",
    "opened_at": "2026-08-13T11:17:53Z",
    "bitrate_kbit": 2113.4,
    "bytes_total": 19053048,
    "datagrams_total": 14478
  },
  "partner-dc": {
    "status": "error",
    "proto": "srt",
    "url": "srt://203.0.113.10:9000",
    "error": "srt handshake rejected: bad passphrase",
    "reconnects_total": 22
  },

```

```
"backup-site": {"status": "disabled", "proto": "srt", "url": "srt://198.51.100.4:9000"}
}
```

Campo	O que significa	Como usar
<code>status</code>	<code>running</code> , <code>error</code> ou <code>disabled</code>	fora de <code>running</code> por mais de N minutos — alerta; <code>disabled</code> é normal, é um destino que o operador desligou
<code>error</code>	a causa do <code>status error</code>	o primeiro a ler: conexão recusada, tempo esgotado, handshake rejeitado
<code>proto</code> , <code>url</code>	o transporte e o destino exatamente como o operador os definiu	dá para ver para onde o sinal vai de fato
<code>opened_at</code>	o momento em que o pusher atual começou	fica mais novo a cada reconexão — o canal está caindo e voltando
<code>bitrate_kbit</code>	a velocidade de envio medida	bem abaixo do bitrate do stream — o envio não dá conta
<code>errors_rate</code>	erros de envio por segundo	cresce com a conexão viva — o receptor ou o canal estão degradando
<code>bytes_total</code>	bytes enviados, acumulado	conciliação de volumes por destino
<code>errors_total</code>	erros de envio, acumulado	—
<code>reconnects_total</code>	ressurreições de um pusher morto	cresce com <code>status: running</code> — o canal rompe e se recupera
<code>retransmitted_packets_total</code>	retransmissões do transporte (SRT, RIST)	crescem — perdas no caminho até o receptor
<code>datagrams_total</code> , <code>frames_total</code>	as unidades enviadas: datagramas nos transportes de pacotes, quadros no RTMP	—

Campos zerados são omitidos da resposta: um destino RTMP não tem datagramas, um UDP não tem quadros nem retransmissões.

Ao lado dos contadores do stream há três somas sobre todos os destinos — `push_bytes_total`, `push_errors_total`, `push_reconnects_total`. Elas respondem «quanto a cabeça de rede enviou no total» sem percorrer o mapa.

6.4.3 Séries do Prometheus

Cada contador de destino é uma série própria com o rótulo `push`:

Série	Rótulos
<code>stream_push_bytes_total</code>	<code>stream</code> , <code>push</code> , <code>proto</code>
<code>stream_push_errors_total</code>	<code>stream</code> , <code>push</code> , <code>proto</code>
<code>stream_push_reconnects_total</code>	<code>stream</code> , <code>push</code> , <code>proto</code>
<code>stream_push_retransmitted_packets_total</code>	<code>stream</code> , <code>push</code> , <code>proto</code>
<code>stream_push_datagrams_total</code> , <code>stream_push_frames_total</code>	<code>stream</code> , <code>push</code> , <code>proto</code>

No servidor Prometheus embutido os mesmos contadores estão disponíveis com os rótulos `name` (stream) e `push`; quando há mais de uma cabeça de rede, soma-se `node`. A profundidade de protocolo — `stream_push_srt_*` e `stream_push_rist_*` — chega junto com a opção de contadores estendidos.

```
# velocidade de envio por destino
rate(stream_push_bytes_total{stream="tv1"}[1m])

# a saída total do stream por todos os destinos
sum by (name) (rate(stream_push_bytes_total{name="tv1"}[1m]))
```

```
# canais que rompem: reconexões em cinco minutos
increase(stream_push_reconnects_total[5m]) > 0

# perdas no caminho até o receptor por SRT/RIST
rate(stream_push_retransmitted_packets_total[1m])
```

6.4.4 Como ler o estado

- **status: error com error não vazio** — nada está sendo enviado, e a causa está nomeada. O destino continua se levantando sozinho, uma vez a cada cinco segundos.
- **status: running, mas bitrate_kbit zerado** — a conexão está lá, os dados não: confira se o próprio sinal está chegando na entrada.
- **status: running com reconnects_total crescente** — o canal rompe e se recupera; `opened_at` mostra a idade da conexão atual.
- **status: disabled** — o destino está desligado pelo interruptor. Os contadores ficam congelados nos últimos valores; isso não é perda de dados.

6.4.5 Os alertas para começar

Pergunta	Onde olhar	O sinal do problema
O sinal chega?	<code>pushes[].status</code> , <code>bitrate_kbit</code> ; <code>rate(stream_push_bytes_total[1m])</code>	o status não é <code>running</code> ; velocidade zerada num stream vivo
Por que não chega?	<code>pushes[].error</code>	o texto da causa: conexão recusada, tempo esgotado, handshake rejeitado
O canal até o receptor rompe?	<code>reconnects_total</code> , <code>opened_at</code>	as reconexões crescem; <code>opened_at</code> fica mais novo a cada poucos minutos
O caminho perde pacotes?	<code>retransmitted_packets_total</code>	cresce com o status estável
A saída dá conta?	o <code>bitrate_kbit</code> do destino contra o bitrate do stream	a velocidade do destino é sistematicamente menor

6.4.6 O que vem depois

- [Entrega para a rede](#) e [entrega por SRT](#) — com o que um destino é configurado.
- [Erros de entrada](#) — a outra metade do quadro: o que chega na cabeça de rede.
- [Monitoramento](#) — os canais de estatística como um todo.

7. Arquivo

7.1 O arquivo

Uma cabeça de rede não precisa do arquivo por causa do espectador, e sim como prova: o que exatamente enviamos às 03:40, se havia sinal na entrada no momento da reclamação, que publicidade foi ao ar na hora disputada. A gravação acontece na mesma máquina que recebe e entrega.

O arquivo é um armazenamento próprio: o stream é anexado em arquivos por hora, separadamente por pista, em vários discos ao mesmo tempo. Ler o arquivo não atrapalha gravá-lo.

7.1.1 Os discos da máquina

Os discos são configurados uma única vez nas configurações da máquina, a seção **DVR**. O painel avisa ali mesmo: estas configurações mudam onde o arquivo é gravado na máquina.

- **Raiz do arquivo** — o ponto de montagem, a casa do catálogo e a base das rotas relativas.
- **Disco 1, Disco 2, ...** — com o botão **Adicionar disco**. A rota de um disco é o nome de um subdiretório dentro da raiz, não uma rota absoluta.
- **Caminho do catálogo** — o diretório de serviço dos metadados do arquivo; campo vazio significa `<root>/catalog`. Não é aqui que o arquivo fica guardado.
- **Verificar montagem antes de gravar** — vale ligar sempre: sem isso, um disco que caiu vira um diretório vazio comum, e a cabeça de rede ficará gravando o arquivo no sistema de arquivos raiz até ele se esgotar.

Sem um único disco as configurações não salvam: o DVR precisa de pelo menos um.

Ao lado deles vivem os **discos de cache** — neles assenta o arquivo alheio já lido; a gravação ao vivo nunca os escolhe, e o espaço é liberado pela idade da leitura de um fragmento. Uma cabeça de rede precisa deles só se serve o arquivo de máquinas vizinhas.

7.1.2 A gravação de um stream

Abra o stream, a aba **DVR**, ligue **Gravar no DVR** e defina os limites:

- **Retenção (horas)** — quantas horas manter;
- **Tamanho máximo (bytes)** — o teto de tamanho.

Com os dois definidos, vence o mais rígido. A limpeza roda uma vez por hora: a profundidade apaga apenas as horas plenamente vencidas, o limite de tamanho conta das gravações mais novas para as mais velhas.

A gravação se equilibra sozinha entre os discos ativos: horas consecutivas de um mesmo stream alternam entre discos, streams vizinhos se espalham de forma parelha e as pistas de um stream multipista são gravadas em paralelo. Um disco posto em somente leitura continua sendo lido, mas não recebe dados novos — é assim que um disco é aposentado sem perder o arquivo.

7.1.3 A leitura

O arquivo é lido por tempo absoluto: um intervalo, um recuo a partir do ao vivo (retrocesso) com emenda perfeita de volta ao ao vivo, um preview JPEG do quadro num momento dado. Isso basta tanto para investigar um incidente quanto para exportar uma gravação sob demanda.

Se o stream tem a [extração de legendas](#) ligada, as pistas extraídas são gravadas junto com o vídeo.

7.1.4 A verificação da gravação

O estado da gravação aparece na mesma aba: a janela do arquivo, as horas com dados e os buracos entre elas, o tamanho do arquivo, as velocidades de gravação dos fragmentos. O que significam os desvios:

Sinal	O que significa
DVR configurado, mas não está gravando	o stream está offline ou a gravação nunca começou
a borda do arquivo fica atrás da hora atual	a gravação parou
o limite inferior parou de avançar	a limpeza não está rodando — adiante há um disco cheio
lacunas: N h com uma janela larga	há buracos no arquivo: o stream caiu ou a gravação estava desligada
Erros de gravação do arquivo — dados estão sendo perdidos	o arquivo está perdendo dados agora mesmo, verifique o disco
Gravação degradada (fragmentos collapsed/skipped/delayed)	os fragmentos são gravados devagar, com atraso ou colapsam — o armazenamento está degradando

A última linha é um sinal precoce, não uma constatação de perda: a fatia de gravações lentas cresce antes de os erros aparecerem.

7.1.5 O que vem depois

- [Monitoramento](#) — para onde vão os contadores de gravação.
- [Legendas e teletexto](#) — o que cai no arquivo junto com o vídeo.

8. Controle de qualidade

8.1 Contadores de transporte (TR 101 290)

A ETSI TR 101 290 é a lista do setor de verificações do fluxo de transporte: o que precisa ser verdadeiro para que um fluxo conte como apto e em que ordem essas violações importam. O Mcaster mantém esses contadores **por cada PID**, tanto na entrada quanto sobre o fluxo analisado, por isso a pergunta «o sinal degradou ou não» deixa de ser questão de opinião.

Isso difere dos [erros de entrada](#): lá está a causa pela qual uma fonte morreu ou estragou; aqui, uma medição do transporte em si.

8.1.1 O que é contado por cada PID

Contador	O que significa	O que fazer
<code>errors_ts_cc</code>	uma violação do continuity counter — uma porção do transporte se perdeu ou se repetiu	o principal indicador de perda; procure-a na rede ou na fonte
<code>errors_ts_tei</code>	o flag transport error indicator está posto: o erro foi marcado por um nó anterior	o problema chegou antes de nós — olhe para cima na cadeia
<code>errors_ts_scrambled</code>	os pacotes continuaram cifrados	a criptografia não foi retirada: CAM, chaves, configuração do descrambler
<code>errors_ts_psi_checksum</code>	a soma de verificação de uma tabela de serviço não bateu	a tabela chegou quebrada; se isso não para de crescer, os receptores perdem o serviço
<code>errors_ts_pmt</code> , <code>errors_ts_pat</code>	problemas com as tabelas de programas	o serviço não se monta; o programa anunciado pode não estar no tronco
<code>errors_pid_lost</code>	o PID sumiu do fluxo	uma pista foi embora — na saída isso é silêncio ou tela preta
<code>broken_pes_count</code>	pacotes PES quebrados	carga útil danificada
<code>empty_packets</code>	pacotes vazios no PID	em geral é normal, mas junto com uma queda do bitrate é sinal de inanição

Os erros do fluxo inteiro — `errors_ts_pat` (PAT ausente ou quebrada) e `errors_ts_service_lost` (um serviço anunciado sumiu) — são contados à parte dos PID: eles significam que o que desmoronou não foram os dados, e sim a descrição deles.

8.1.2 PCR: exatidão e jitter

O PCR é o relógio do receptor. Se ele treme, o decodificador não segura o buffer, e a imagem se despedaça mesmo com o fluxo intacto.

O jitter é contado em buckets, com o `PCR_AC` = ±500 ns do padrão como fronteira:

- **exatamente zero e até 500 ns** — dentro da tolerância;
- **de 500 ns a 500 µs** — uma violação de exatidão;
- **acima de 500 µs** — uma violação grosseira;
- `pcr_resync` — foi preciso reancorar o relógio: uma ruptura da escala, não um tremor.

Não existe de propósito um campo único que some tudo: um «jitter médio» esconde exatamente aquilo para o que a medição existe. O que se deve olhar é a distribuição — uma fatia crescente dos buckets `fail` com o bitrate inalterado significa que a fonte ou o transporte parou de segurar o ritmo.

8.1.3 O buffer do decodificador (HRD)

`hrd_buffer_min`, `hrd_buffer_max` e `hrd_buffer_underrun` descrevem o modelo de buffer do decodificador hipotético: se o fluxo chega ao receptor de um jeito que deixe o buffer nem vazio nem transbordando.

Isso responde a uma pergunta que assistir ao stream num computador não pode: **se o fluxo sobreviverá a um decodificador de hardware**. Um reprodutor de software perdoa o que um set-top box ou um modulador não perdoarão.

8.1.4 A linha do tempo: saltos e não monotonicidade

Um grupo à parte de contadores é sobre tempo: `dts_goes_backwards`, `dts_jump_forward`, `too_large_dts_jump`, `time_corrections`, `corrected_backward_pts`.

Para uma cabeça de rede esse é um sinal precoce de uma fonte ruim: um fluxo cuja escala salta chega ao receptor, mas a emenda, a gravação e a multiplexação sofrerão com isso constantemente.

8.1.5 Como usar isso

- **Um alerta sobre** `rate(errors_ts_cc) > 0` por cada PID captura a degradação do transporte antes que um espectador a veja.
- **Pacotes cifrados** são um sinal binário: ou não há nenhum, ou a configuração está errada.
- **O jitter do PCR e o HRD** são o critério de aceitação da saída antes de entregá-la a um modulador.
- **O contador CC na entrada** pode comandar o ar: o limiar `cc_errors_limit` marca uma fonte como degradada e a cabeça de rede passa para a reserva — veja [Redundância de entradas](#).

A profundidade por cada PID é aberta pela opção de licença de contadores estendidos; o agregado de erros está sempre disponível. Para onde os contadores vão e com o que olhá-los está na página [Monitoramento](#).

8.1.6 Os limites

O Mcaster mede o que é visível a partir da análise do transporte. As verificações que exigem a parte de radiofrequência — nível de sinal, MER, BER antes e depois da correção — pertencem ao hardware de recepção, não à cabeça de rede, e aqui não são contadas.

8.2 Monitoramento

O Mcaster sempre coleta o conjunto completo de contadores. A licença e a tarefa só decidem por qual canal você os lê.

Canal	O que dá	Disponibilidade
O painel e a API	um retrato instantâneo: o que o stream e os destinos dele estão fazendo agora	todos
O servidor Prometheus embutido	um datasource Grafana pronto, com algumas horas de histórico	todos
O scrape do Prometheus	métricas cruas para a sua própria TSDB e o seu alertmanager	a opção de contadores estendidos
Retroview	o monitoramento em nuvem do fabricante: meses de histórico, tendências, alertas prontos	assinatura

A regra de escolha é simples:

- «O que está acontecendo **agora**?» — o painel ou a API.
- «O que aconteceu nas **últimas horas**?» — o servidor Prometheus embutido.
- «O que aconteceu **na semana passada** e por que caiu de madrugada?» — Retroview.
- «Quero **meu próprio** Prometheus, Grafana e alertmanager» — o scrape.

8.2.1 O que olhar exatamente

Os contadores em si estão explicados nas páginas de cada tarefa, não aqui:

- [Monitoramento de entrega](#) — destinos: status, velocidade, erros, reconexões, retransmissões.
- [Erros de entrada](#) — causas das falhas e minutos sem sinal.
- [TR 101 290](#) — qualidade do transporte por cada PID.
- [Redundância de entradas](#) — tempo na principal e na reserva.

8.2.2 O servidor Prometheus embutido

Dentro do Mcaster roda um servidor Prometheus: a HTTP API `/api/v1/query`, `/api/v1/query_range`, `/api/v1/series`, `/api/v1/labels`, `/api/v1/label/{name}/values` sobre um armazenamento próprio em memória. Para o Grafana é um datasource pronto — adicione um datasource do tipo Prometheus e aponte para a URL da cabeça de rede. Os gráficos do painel se alimentam dessas mesmas consultas.

Limites de construção:

- o histórico é de algumas horas, em memória; depois de um restart o armazenamento fica vazio. É o canal do «o que está acontecendo agora», não um arquivo de métricas;
- o PromQL é suportado como um subconjunto: seletores com matchers de rótulos, `rate` / `irate` / `increase`, as agregações `sum` / `avg` / `min` / `max` / `count` com `by()` / `without()`, aritmética, `offset`. Uma construção não suportada devolve um erro explícito, não um resultado distorcido;
- o conjunto de séries é o básico; a profundidade por PID, SRT e RTP chega com a opção de contadores estendidos.

```
# a velocidade de captura do stream
rate(stream_input_bytes_total{stream="tv1"}[1m])

# erros de entrada em 5 minutos por causa
sum by (cause) (increase(errors_detail_total{name="tv1"}[5m]))

# a saída por cada destino
rate(stream_push_bytes_total{stream="tv1"}[1m])
```

Quando há mais de uma cabeça de rede, cada série carrega um rótulo `node`, e um mesmo dashboard funciona tanto contra uma cabeça de rede única quanto contra o plano de controle: agregue com `sum without (node)` — numa máquina única o rótulo não existe e a agregação não muda nada.

8.2.3 O scrape para o seu próprio monitoramento



A opção de contadores estendidos

Os endpoints do scrape estão disponíveis só com esta opção de licença. Sem ela, use o servidor Prometheus embutido e o Retroview.

Métricas no formato de texto do Prometheus:

- GET `/streamer/api-v4/live-metrics` — streams e entradas;
- GET `/streamer/api-v4/sessions-metrics` — sessões;
- GET `/streamer/api-v4/dvr/metrics` — discos do arquivo e catálogo;
- GET `/streamer/api-v4/runtime/metrics` — o processo e o alocador de memória.

O histórico fica com você e é limitado só pelo retention da sua TSDB; a profundidade completa está disponível, incluindo métricas por PID, SRT e RTP; o alerting é o seu alertmanager. Os contadores são monótonos e sobrevivem à observação de restarts.

8.2.4 Retroview

O Retroview é o monitoramento em nuvem do fabricante. A cabeça de rede envia telemetria uma vez por minuto — o catálogo completo de contadores, incluindo todo o detalhe estendido — e o Retroview o guarda por meses. Nada precisa ser configurado na cabeça de rede: o canal funciona junto com a licença.

Quando ir lá, e não ao servidor Prometheus embutido:

- **histórico e tendências** — crescimento de carga ao longo de meses, planejamento de capacidade;
- **post-mortems** — o que o stream estava fazendo de madrugada: a telemetria sobrevive a restarts;
- **conciliação de consumo** — os contadores são amostrados no tempo por um sistema externo;
- **análise de causas sem a opção de contadores estendidos** — a telemetria sempre carrega o detalhe;
- **alertas prontos** — regras de produto em vez de regras caseiras.

8.2.5 Os logs

Os logs são estruturados e o nível é definido pela variável `RUST_LOG`:

```
journalctl -u mcaster -f
```

Os registros carregam o nome do stream e o módulo de destino, então filtrar por um stream específico funciona sem um indexador externo. Os traces são exportados por OpenTelemetry (OTLP).

8.2.6 A saúde da cabeça de rede

- Sonda de readiness: GET `/streamer/api-v3/monitoring/readiness` — para balanceadores e orquestradores.
- Memória por parte do pipeline: `sum by (part) (stream_memory_bytes)` — crescimento monótono sem crescimento de carga merece investigação.