← Visão Geral

Configuração do OmniSCP

Todas as configurações ajustáveis pelo operador são fornecidas através de `config/runtime.exs`, cada uma respaldada por uma variável de ambiente para que os valores possam ser substituídos no momento da implantação sem editar o arquivo. Os valores mostrados abaixo são os padrões aplicados quando a variável de ambiente correspondente não está definida.

```
import Config

config :omniscp,
  sbi_scheme: System.get_env("SBI_SCHEME", "http"),
  sbi_addr: System.get_env("SBI_ADDR", "127.0.0.17"),
  sbi_port: String.to_integer(System.get_env("SBI_PORT", "7777")),
  nrf_uri: System.get_env("NRF_URI", "http://127.0.0.1:7777"),
  mcc: System.get_env("MCC", "999"),
  mnc: System.get_env("MNC", "70"),
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000")),
  prometheus_metrics_port:
String.to_integer(System.get_env("PROMETHEUS_PORT", "9568")),
  discovery_cache_ttl:
String.to_integer(System.get_env("DISCOVERY_CACHE_TTL", "60000")),
  lb_strategy: String.to_atom(System.get_env("LB_STRATEGY",
"round_robin")),
  max_retries: String.to_integer(System.get_env("MAX_RETRIES",
"2")),
  upstream_timeout:
String.to_integer(System.get_env("UPSTREAM_TIMEOUT", "5000"))

config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all
```

Parâmetros `:omniscp`

Parâmetro	Tipo	Requerido	Padrão
`sbi_scheme`	String	Não	`"http"`
`sbi_addr`	String	Não	`"127.0.0.17"`
`sbi_port`	Inteiro	Não	`7777`
`sbi_certfile`	String	Não	*(não definido)*

Parâmetro	Tipo	Requerido	Padrão
`sbi_keyfile`	String	Não	*(não definido)*
`nrf_uri`	String	Não	`"http://127.0.0.1:777`
`mcc`	String	Não	`"999"`
`mnc`	String	Não	`"70"`

Parâmetro	Tipo	Requerido	Padrão
`heartbeat_interval`	Inteiro (ms)	Não	`10000`
`prometheus_metrics_port`	Inteiro	Não	`9568`
`discovery_cache_ttl`	Inteiro (ms)	Não	`60000`

Parâmetro	Tipo	Requerido	Padrão
`lb_strategy`	Átomo	Não	`round_robin`
`max_retries`	Inteiro	Não	`2`

Parâmetro	Tipo	Requerido	Padrão
`upstream_timeout`	Inteiro (ms)	Não	`5000`

Capacidade/prioridade do perfil NRF

Esses dois valores fazem parte do próprio perfil NRF do SCP. Eles **não** são definidos em `runtime.exs` e não têm uma variável de ambiente dedicada, portanto normalmente assumem seus padrões, mas podem ser ajustados em tempo de execução através da API de gerenciamento (`PATCH /api/oam/config/{id}`), que também aciona uma nova re-registro no NRF para que o novo valor seja anunciado.

Parâmetro	Tipo	Requerido	Padrão	Var. de Ambiente	Desc
`nf_capacity`	Inteiro (0-65535)	Não	`100`	(nenhuma, apenas OAM)	Valor de cap anunciado n do SCP. Consumidore podem usá-l ponderar o S
`nf_priority`	Inteiro (0-65535)	Não	`0`	(nenhuma, apenas OAM)	Valor de prio anunciado n do SCP (mer prioridade).

> **Padrões de fallback compilados.** A aplicação também possui padrões de fallback compilados usados apenas se a chave `:omniscp` estiver completamente ausente, e estes diferem dos padrões de ambiente `runtime.exs` acima (por exemplo, `sbi_addr` volta para `127.0.0.200`, `nrf_uri` para `http://127.0.0.10:7777`, e `max_retries` para `1`). Em operação normal, `runtime.exs` fornece cada chave, então os valores nas tabelas acima são os padrões efetivos.

Ajustes de balanceador de carga e sobrecarga

Essas chaves de configuração da aplicação `:omniscp` governam o rastreamento de saúde baseado em falhas e o controle de sobrecarga. Elas **não têm variável de ambiente** e não são expostas através da API OAM, portanto são definidas apenas via configuração da aplicação. Cada uma é lida ao vivo em tempo de execução, então uma mudança na configuração da aplicação em execução entra em vigor na próxima utilização do caminho de solicitação sem uma recompilação. Os padrões mostrados se aplicam quando a chave não está definida.

Parâmetro	Tipo	Requerido	Padrão	Descriçã
`health_check_interval`	Inteiro (ms)	Não	`15000`	Intervalo da varredura de saúde/recupe em segundo p sobre instânc de produtores
`failure_threshold`	Inteiro	Não	`3`	Falhas consecutivas encaminhame antes que um instância seja marcada com não saudável retirada da rotação.
`recovery_cooldown_ms`	Inteiro (ms)	Não	`30000`	Período de es após o qual u instância não saudável é elegível para retornar à rot
`oci_reduction_threshold`	Inteiro	Não	`50`	Limite de con de sobrecarga governa quar um produtor sobrecarrega (por seu métr de redução `3` `Sbi-Oci`) é removido do conjunto de candidatos ar

Parâmetro	Tipo	Requerido	Padrão	Descriçã
				da seleção do balanceador carga. Veja Operações → Roteamento e Descoberta.

Ajuste do listener SBI

O proxy SBI é servido sobre HTTP/2. Os seguintes parâmetros de transporte são fixos na aplicação (não ajustáveis por ambiente) e estão documentados aqui porque governam o comportamento de concorrência do SCP sob carga:

Configuração	Valor	Descrição
`max_concurrent_streams`	`5000`	Máximo de streams HTTP/2 concorrentes por conexão de entrada. Limita quantas solicitações SBI em andamento uma única conexão de consumidor pode multiplexar através do proxy.
`num_acceptors`	`100`	Tamanho do pool de aceitadores do listener SBI para aceitar novas conexões TCP de entrada.

Solicitações de saída (upstream) para produtores usam um pool de conexões Finch compartilhado fornecido pelo runtime comum Omni5G, reutilizado em todo o tráfego encaminhado e indexado pelo host do produtor. O dimensionamento desse pool não é exposto como uma configuração do OmniSCP, portanto, a concorrência upstream é governada por esse pool compartilhado juntamente com `upstream_timeout` (que limita cada solicitação encaminhada individual) e o limite de `max_concurrent_streams` de entrada acima. Para escalar um SCP ocupado, as alavancas que você controla são

`upstream_timeout` (timeouts mais curtos liberam conexões de volta ao pool mais cedo) e `discovery_cache_ttl` (uma maior taxa de acerto remove viagens de ida e volta ao NRF do caminho de solicitação); `max_concurrent_streams` limita quanto uma única conexão de consumidor pode multiplexar através do proxy.

API de Gerenciamento / OAM

O OmniSCP expõe uma API de gerenciamento/OAM sobre **HTTPS na porta 8443** para status somente leitura e controle operacional ao vivo. Esses endpoints não transportam tráfego de assinantes; restrinja o acesso à rede de operações. Todos os caminhos abaixo são servidos sob o caminho base `/api` do listener de gerenciamento.

Endpoints de Status (somente leitura)

Método	Caminho	Descrição
`GET`	`/api/status/api`	Status do serviço API.
`GET`	`/api/status/license`	Status da licença.
`GET`	`/api/status/nrf`	Status de registro NRF para o SCP.
`GET`	`/api/status/nf`	Status do NF SCP: endereço/porta SBI, tamanho do cache de descoberta, estratégia `lb_strategy` ativa e saúde do balanceador de carga por instância.
`GET`	`/api/api/cache`	Conteúdo do cache de descoberta: por entrada, o tipo de NF alvo, nome do serviço, contagem de instâncias e TTL restante.
`GET`	`/api/api/health`	Saúde do balanceador de carga: por instância rastreada, o indicador de saúde e contagem de falhas, além de totais saudáveis/não saudáveis.
`GET`	`/api/api/statistics`	Estatísticas do SCP: tamanho do cache de descoberta e contagens de instâncias rastreadas/saudáveis/não saudáveis.
`GET`	`/api/oam/config`	Configuração de tempo de execução efetiva atual (SBI, NRF, TTL de cache, `lb_strategy`,

Método	Caminho	Descrição
		`max_retries`, `upstream_timeout`, `nf_capacity`, `nf_priority`, nível de log).
`GET`	`/api/oam/cache`	Visualização do cache de descoberta (como `/api/api/cache`).
`GET`	`/api/oam/upstream/drain`	Lista as instâncias de NF upstream atualmente drenadas.

Endpoints de Controle

Método	Caminho	Descrição
`PATCH`	`/api/oam/config/{id}`	Atualiza as configurações de tempo de execução selecionadas ao vivo: `discovery_cache_ttl`, `lb_strategy`, `max_retries`, `upstream_timeout`, `nf_capacity`, `nf_priority`, `log_level`. Os valores são validados (`lb_strategy` deve ser um dos `round_robin`, `weighted`, `priority`; as chaves inteiras devem ser inteiros). Alterar `nf_capacity`/`nf_priority` também aciona uma nova re-registro no NRF.
`DELETE`	`/api/oam/cache/{id}`	Limpa o cache de descoberta para um tipo específico de NF, ou passe `all` para limpar todo o cache.
`POST`	`/api/oam/upstream/drain`	Drena um upstream para que o proxy pare de selecioná-lo. Corpo: `{"nf_instance_id": "..."}` (alias `"target"`), opcional `"reason"`.
`DELETE`	`/api/oam/upstream/drain/{id}`	Remove o dreno de um upstream pelo ID da instância NF, tornando-o elegível para seleção novamente.

Método	Caminho	Descrição
`POST`	`/api/oam/log_level`	Altera o nível de log ativo em tempo de execução. Corpo: `{"level": "..."}` (`emergency`, `alert`, `critical`, `error`, `warning`/`warn`, `notice`, `info`, `debug`).
`POST`	`/api/oam/nrf/reregister`	Força uma re-registro imediata do SCP com o NRF.

Drenagem upstream é um estado administrativo, separado do rastreamento de saúde baseado em falhas automático. Instâncias drenadas são filtradas da lista de candidatos *antes* da seleção do balanceador de carga, então um produtor drenado nunca recebe tráfego encaminhado. Se **cada** candidato para um serviço estiver drenado, o proxy não tem próximo salto utilizável e solicitações para esse serviço retornam `504 NF_DISCOVERY_FAILURE` (veja Operações → Respostas de erro do proxy e Solução de Problemas).

Registro

Parâmetro	Tipo	Requerido	Padrão	Des
`format`	Tupla	Não	`{OmniLogger.JsonFormatter, :format}`	Forma linhas como estrut atravé format compa `OmniL` adequ para ir por un pipelir logs.
`metadata`	Átomo / Lista	Não	`:all`	Metad log a s emitid `:all` i todas chave metad anexa uma e de log

O nível de log ativo pode ser alterado em tempo de execução sem uma reinicialização através da API de gerenciamento, seja `POST /api/oam/log_level` ou `PATCH /api/oam/config/{id}` com um valor `log_level`.

← Visão Geral

Métricas do OmniSCP

As métricas são expostas no listener do Prometheus (`prometheus_metrics_port`, padrão `9568`) em `/metrics`. Os valores das tags de resultado nas métricas de proxy são: `success` (2xx/3xx), `client_error` (4xx), `server_error` (5xx), `error` (conexão/timeout).

Métricas de Proxy

Métrica: `omni_scp_proxy_requests_count`
Tipo: Contador
Descrição: Contagem de resultados de proxy por solicitação.
Labels: `target_nf_type`, `result`

Métrica: `omni_scp_proxy_requests_duration_ms`
Tipo: Resumo
Descrição: Duração do proxy por solicitação em milissegundos.
Labels: `target_nf_type`

Métrica: `omni_scp_proxy_requests_total`
Tipo: Contador
Descrição: Total de solicitações proxy por tipo de NF alvo e resultado.
Labels: `target_nf_type`, `result`

Métrica: `omni_scp_proxy_request_duration_ms`
Tipo: Histograma (distribuição)
Descrição: Duração da solicitação de proxy. Baldes: 1, 5, 10, 25, 50, 100, 250, 500, 1000 ms.
Labels: `target_nf_type`

O SCP é um proxy de encaminhamento sem estado e não mantém estado por associação, portanto, não expõe nenhuma métrica de "associações ativas". A atividade ao vivo é observada através do contador de solicitações de proxy

acima (taxa, razão de erro) e os números de saúde do balanceador de carga na API OAM.

```
# Taxa de solicitações proxy por tipo de NF alvo
sum by (target_nf_type) (rate(omni_scp_proxy_requests_total[5m]))

# Razão de erro do servidor entre todos os produtores
sum(rate(omni_scp_proxy_requests_total{result="server_error"}
[5m]))
  / sum(rate(omni_scp_proxy_requests_total[5m]))

# Latência p95 do proxy por tipo de NF
histogram_quantile(0.95, sum by (le, target_nf_type)
(rate(omni_scp_proxy_request_duration_ms_bucket[5m])))
```

Métricas do Cache de Descoberta

Métrica: `omni_scp_discovery_cache_hits`
Tipo: Contador
Descrição: Acertos no cache de descoberta, por serviço.
Labels: `target_nf_type`, `service_name`

Métrica: `omni_scp_discovery_cache_misses`
Tipo: Contador
Descrição: Erros no cache de descoberta, por serviço.
Labels: `target_nf_type`, `service_name`

Métrica: `omni_scp_cache_hits_total`
Tipo: Contador
Descrição: Acertos agregados no cache de descoberta.
Labels: nenhum

Métrica: `omni_scp_cache_misses_total`
Tipo: Contador
Descrição: Erros agregados no cache de descoberta.
Labels: nenhum

```
# Razão geral de acertos no cache (alvo > 0.9 em uma topologia
estável)
sum(rate(omni_scp_cache_hits_total[5m]))
  / (sum(rate(omni_scp_cache_hits_total[5m])) +
sum(rate(omni_scp_cache_misses_total[5m])))

# Taxa de erros por serviço (valores altos significam redescoberta
frequente do NRF)
sum by (target_nf_type, service_name)
(rate(omni_scp_discovery_cache_misses[5m]))
```

Métrica de Registro do NRF

Métrica: `omni_scp_nrf_registration_status`
Tipo: Gauge
Descrição: Status de registro do SCP com o NRF (`1` = registrado, `0` = não registrado).
Labels: `nf_type`

```
# Alerta se o SCP perder seu registro no NRF
min(omni_scp_nrf_registration_status) < 1
```

Métricas da VM BEAM

O runtime também exporta gauges de host/VM: `beam_memory_total`, `beam_memory_processes`, `beam_memory_processes_used`, `beam_memory_system`, `beam_memory_atom`, `beam_memory_atom_used`, `beam_memory_binary`, `beam_memory_code`, `beam_memory_ets` (todos em bytes), e `beam_processes_count`, `beam_ports_count`, `beam_atom_count`, `beam_vm_uptime` (segundos). Estes são úteis para monitoramento de capacidade e vazamentos do próprio processo SCP.

> Os nomes das métricas acima usam a normalização de ponto para sublinhado do Prometheus; as séries de histograma expõem os sufixos usuais `_bucket`, `_sum` e `_count`.

← Visão Geral

Operações do OmniSCP

Este guia cobre o papel do SCP no 3GPP, as interfaces e endpoints que ele expõe, seu comportamento de roteamento e descoberta de requisições, e os principais procedimentos em tempo de execução com diagramas de sequência. Para cada configuração ajustável, consulte Configuração; para métricas, veja Métricas; para tratamento de falhas, consulte Solução de Problemas.

Papel do 3GPP e Referências de Especificação

O SCP fornece **comunicação indireta** conforme definido na TS 23.501. No **Modelo C**, o consumidor realiza a descoberta por conta própria (via NRF), mas delega o encaminhamento de requisições ao SCP. No **Modelo D**, o consumidor delega tanto a descoberta quanto o encaminhamento ao SCP: ele envia a requisição com parâmetros `3gpp-Sbi-Discovery-*` e o SCP realiza a descoberta NRF em seu nome. O OmniSCP suporta ambos os modelos, além do encaminhamento direto quando o consumidor já conhece o destino.

Especificação	Seção	Relevância
TS 23.501	§6.3.1, Anexo E	Papel do SCP e os modelos de comunicação indireta (C e D) na 5G SBA.
TS 23.501	§7.3	Comportamento de manuseio de requisições do SCP dentro da SBA.
TS 29.500	§6.10	Roteamento SBI via SCP; os cabeçalhos de roteamento/descoberta `3gpp-Sbi-*`.
TS 29.500	§6.10.3	Conjunto de cabeçalhos `3gpp-Sbi-Discovery-*` transportados em requisições de descoberta delegada.
TS 29.500	§6.10.3.2	Cabeçalho `3gpp-Sbi-Target-apiRoot` (destino de roteamento direto); seleção do nome do serviço primário.
TS 29.500	§6.10.3.3	Cabeçalho de resposta `3gpp-Sbi-Producer-Id` (vinculação consumidor-produtor).
TS 29.500	§5.2.2.2	Cabeçalho `User-Agent` transmite o tipo de NF do solicitante.
TS 29.510	§6.2	Nnrf_NFDiscovery, o serviço de descoberta NRF que o SCP consome.
TS 29.510	§6.1	Nnrf_NFManagement, auto-registro do SCP e notificações de status do NF.

Referência: TS 23.501, TS 29.500, TS 29.510.

Interfaces e Ouvintes

O OmniSCP é um proxy reverso HTTP abrangente. Consumidores de NF direcionam seu tráfego SBI para o ouvinte SBI do SCP em vez de diretamente para os produtores. Para cada requisição, o SCP resolve um produtor alvo (de um cabeçalho de requisição, de um resultado de descoberta NRF em cache ou consultando o NRF), seleciona uma instância via a estratégia de balanceamento de carga configurada, encaminha a requisição e, em caso de falha, tenta novamente contra uma instância diferente.

Ouvinte	Porta Padrão	Esquema	Pro
Proxy SBI	`7777` (`sbi_port`)	HTTP / HTTPS (`sbi_scheme`)	Aceita requi consumidor notificações Todos os ca notificações encaminhac produtores. `sbi_scheme` `sbi_certfi` para termin ouvinte.
Métricas Prometheus	`9568` (`prometheus_metrics_port`)	HTTP	Serve o end `/metrics`. C Métricas.
API de Gerenciamento / OAM	`8443`	HTTPS	Endpoints d somente lei OAM. Consu Configuraçã Gerenciame

O ouvinte SBI é servido sobre HTTP/2 com um limite de stream por conexão e um pool de aceitadores fixo; esses valores de ajuste de transporte são descritos em Configuração → Ajuste do ouvinte SBI.

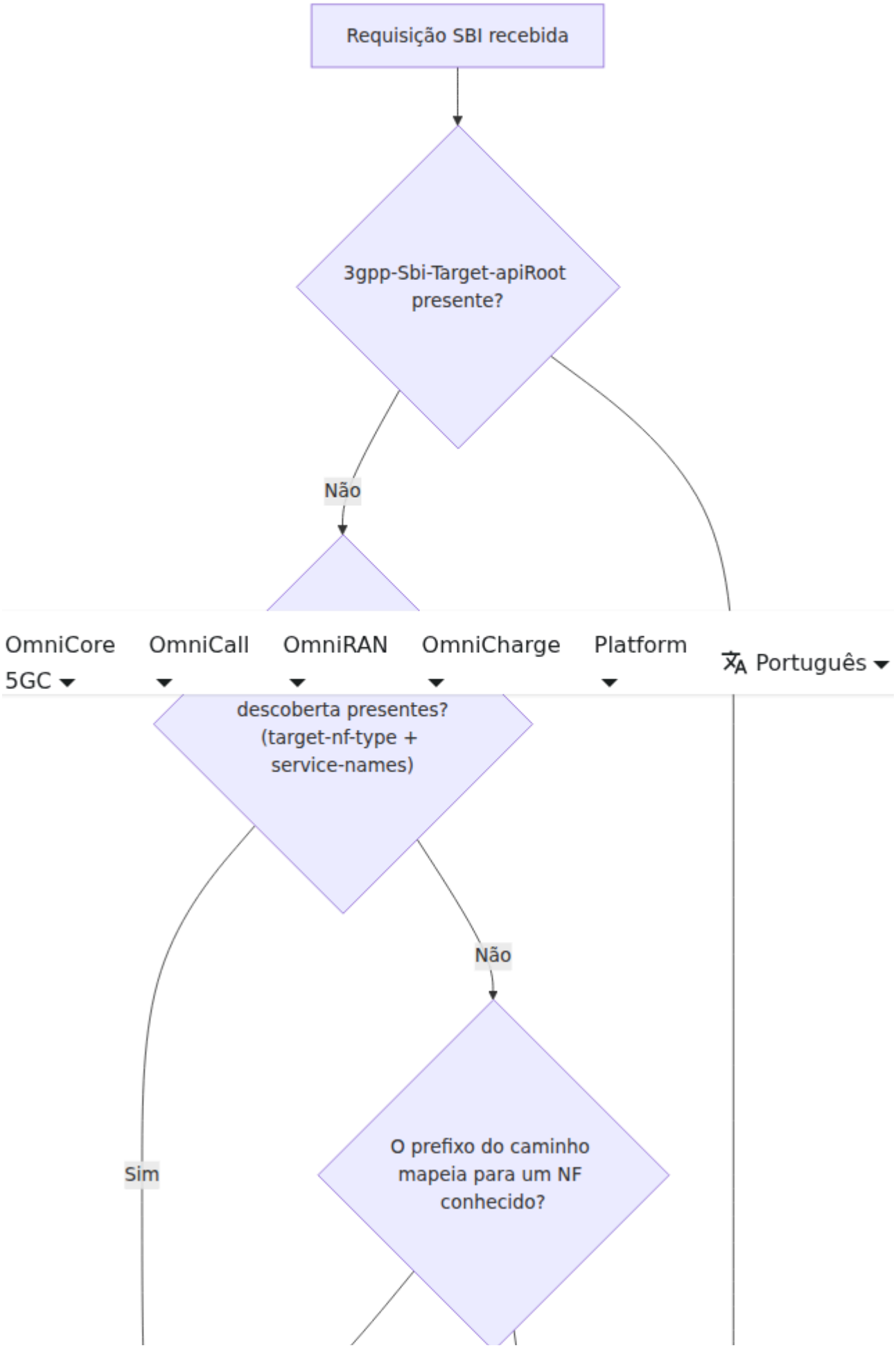
Tabela de endpoints SBI

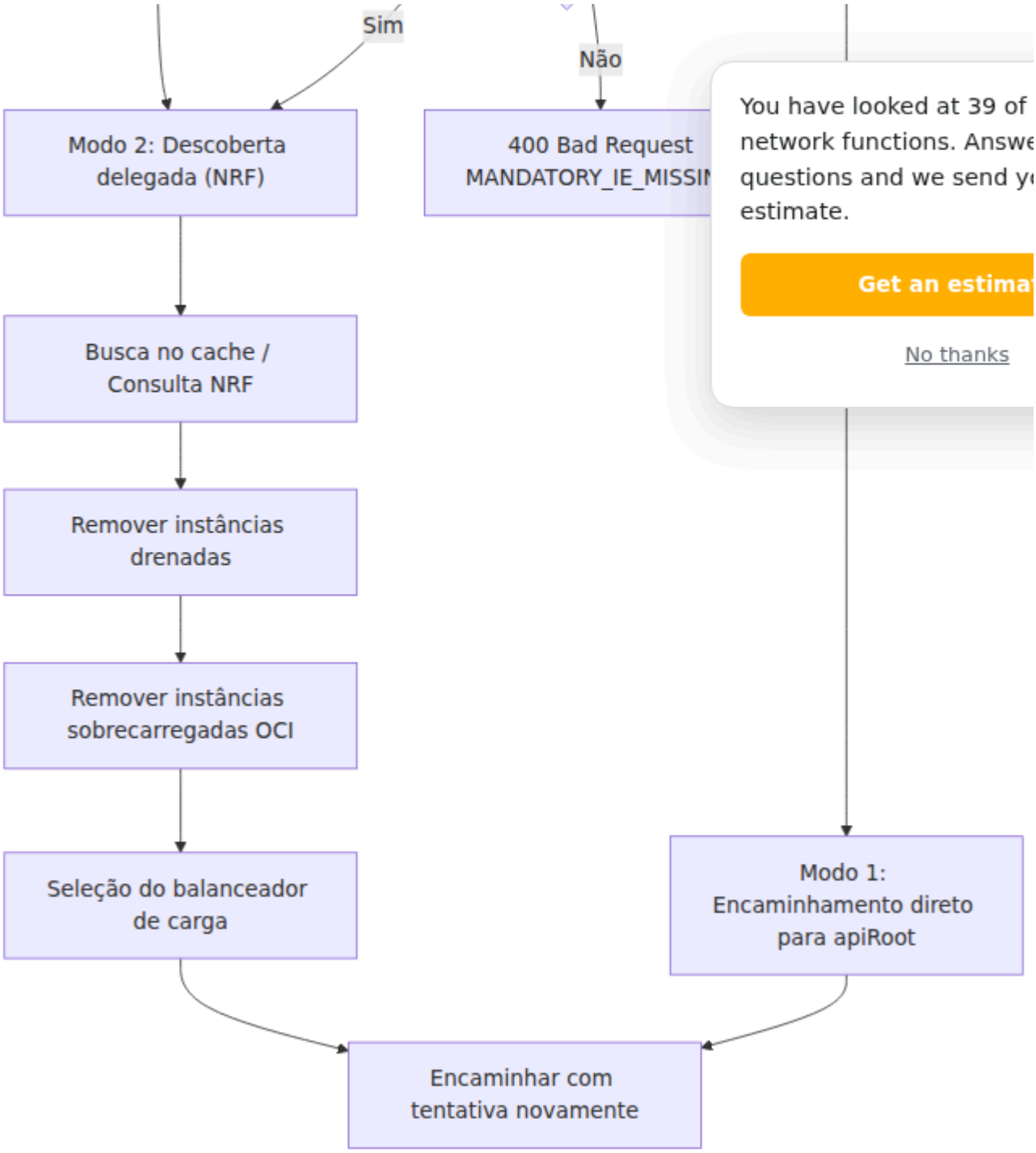
O OmniSCP é transparente: exatamente um caminho é tratado localmente, e todas as outras requisi��ões são encaminhadas.

Método	Caminho	Tratado localmente	Descrição
`POST`	`/nnrf-nfm/v1/nf-status-notify`	Sim	Recebe notificações de status do NF NRF. Em `NF_DEREGISTERED` ou `NF_PROFILE_CHANGED`, o cache de descoberta é invalidado para a única instância afetada identificada por seu `nfInstanceId` (retornando à remoção de um único tipo de NF apenas se nenhum ID de instância estiver presente, e não fazendo nada se nenhum dos dois estiver presente). O cache inteiro nunca é limpo. Retorna `204 No Content`.
`*`	todos os outros caminhos	Não (encaminhado)	Encaminhado para o produtor resolvido de acordo com o modo de roteamento ativo.

Roteamento e Descoberta

Em cada requisição recebida, o SCP inspeciona os cabeçalhos e o caminho da requisição para decidir **como** resolver o produtor alvo. Três modos de roteamento são avaliados em ordem de prioridade:


Requisição SBI recebida
3gpp-Sbi-Target-apiRoot presente?
Não
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
descoberta presentes? (target-nf-type + service-names)
Não
Sim
O prefixo do caminho mapeia para um NF conhecido?



Modo 1: Encaminhamento direto

Quando o consumidor envia `3gpp-Sbi-Target-apiRoot`, ele já conhece a raiz da API do produtor. O SCP remove os cabeçalhos específicos do SCP e encaminha a requisição diretamente para aquele URI base sem consulta NRF, cache ou balanceamento de carga.

Modo 2: Descoberta delegada (Modelos C/D)

Quando o consumidor envia `3gpp-Sbi-Discovery-target-nf-type` e `3gpp-Sbi-Discovery-service-names`, o SCP resolve o conjunto de produtores por conta própria:

1. **Busca no cache**: indexada por `{tipo de NF alvo, nome do serviço primário}`. O primeiro nome em `3gpp-Sbi-Discovery-service-names` é tratado como o serviço primário (TS 29.500 §6.10.3.2). Em um acerto dentro do TTL, a lista de instâncias em cache é utilizada e nenhuma consulta NRF é feita.
2. **Descoberta NRF**: em um erro, o SCP consulta o NRF (Nnrf_NFDiscovery), passando os parâmetros de descoberta que recebeu: tipo de NF do solicitante, lista de PLMN alvo, lista de S-NSSAI do solicitante, ID do conjunto NF e ID da instância NF alvo. O resultado é armazenado sob a mesma chave para `discovery_cache_ttl`.
3. **Filtragem de drenagem**: qualquer instância cujo `nfInstanceId` esteja administrativamente drenado (veja Configuração → API de Gerenciamento) é removida do conjunto de candidatos antes da seleção.

4. **Controle de sobrecarga**: qualquer instância que esteja sinalizando sobrecarga via seu cabeçalho de resposta `3gpp-Sbi-Oci` (Informação de Controle de Sobrecarga) em ou acima do `oci_reduction_threshold` durante seu período de validade é removida do conjunto de candidatos (TS 29.500 §6.3). Isso é feito com o melhor esforço: se todos os candidatos estiverem sobrecarregados, o conjunto completo é mantido em vez de descartar a requisição.
5. **Seleção do balanceador de carga**: uma instância é escolhida entre os candidatos restantes usando `lb_strategy`.
6. **Encaminhar com tentativa novamente**: a requisição é encaminhada; em caso de falha, uma instância diferente é tentada, até `max_retries`. Em cada resposta, o `3gpp-Sbi-Oci` anunciado do produtor é registrado para que requisições subsequentes possam reduzir a carga dele.

Modo 3: Inferência baseada em caminho

Quando nenhum cabeçalho de roteamento está presente, o SCP infere o tipo de NF alvo a partir do prefixo do nome do serviço do caminho da requisição (formato do caminho `/<nome-do-serviço>/<versão>/...`) e, em seguida, procede exatamente como no Modo 2.

Prefixo do caminho	Tipo de NF
`nudm-`	UDM
`nausf-`	AUSF
`namf-`	AMF
`nsmf-`	SMF
`npcf-`	PCF
`nudr-`	UDR
`nnssf-`	NSSF
`nbsf-`	BSF
`nnrf-`	NRF
`nchf-`	CHF
`nnef-`	NEF
`naf-`	AF

Se o prefixo do caminho não estiver nesta tabela, o SCP retorna `400 Bad Request` com a causa `MANDATORY_IE_MISSING`.

Estratégias de balanceamento de carga

A estratégia de seleção é definida por `lb_strategy` e se aplica aos Modos 2/3 (conjuntos de instâncias descobertas):

Estratégia	Lógica de seleção
`round_robin`	Cicla através das instâncias candidatas saudáveis em ordem. Um contador de rotação separado é mantido por grupo `{nfType, serviceName}` para que serviços diferentes não interfiram. Este é o padrão.
`weighted`	Seleciona a instância com a melhor (menor) pontuação `load - capacity` de seu perfil NRF, preferindo produtores de alta capacidade e baixa carga. `load` ausente é padrão para 50 e `capacity` ausente para 100.
`priority`	Seleciona a instância com o menor valor de `priority` (maior prioridade) de seu perfil NRF. `priority` ausente é padrão para 65535. Adequado para implantações ativas/em espera.

> **De onde vêm os pesos.** As estratégias `weighted` e `priority` leem `load`, `capacity` e `priority` do **próprio perfil NRF de cada produtor**; o SCP não possui configuração de peso por produtor. Para influenciar a seleção, você ajusta esses valores nos produtores (ou no NRF), não no SCP. As configurações `nf_capacity`/`nf_priority` do SCP (Configuração → Capacidade/Prioridade do perfil NRF) são um conceito separado: descrevem *o próprio SCP* em seu próprio perfil para que pares possam pesar o SCP, e não têm efeito sobre como o SCP pesa os produtores.

Rastreamento de saúde baseado em falhas. Independentemente da estratégia, o balanceador de carga rastreia os resultados por instância. Uma instância é marcada como não saudável após **3 falhas consecutivas** e é excluída da seleção; uma varredura de saúde em segundo plano é executada a cada **15 segundos** e retorna uma instância ao serviço uma vez que um **período de espera de 30 segundos** tenha decorrido desde sua última falha. Se todos os candidatos estiverem não saudáveis, o SCP recorre à lista completa em vez de falhar completamente (favorecendo a disponibilidade em vez da exclusão estrita). Este rastreamento automático de saúde é distinto do estado de drenagem controlado pelo operador.

Esses três limites (contagem de falhas, intervalo de varredura e período de espera) não são expostos como variáveis de ambiente e têm os valores padrão conforme mostrado acima, mas são chaves de configuração reais `:omniscp` (`failure_threshold`, `health_check_interval`, `recovery_cooldown_ms`) que são lidas ao vivo em tempo de execução. Nenhum endpoint OAM altera esses valores em tempo de execução. Consulte Configuração → Ajustes do balanceador de carga e sobrecarga. Para retirar um produtor da rotação em um cronograma que você controla (em vez de esperar pelo limite de falhas), use drenagem administrativa em vez disso. Uma falha para este propósito é uma resposta 5xx ou um erro de conexão/timeout; qualquer resposta com status abaixo de 500 (incluindo 4xx) conta como um sucesso e redefine o contador de falhas da instância.

Comportamento SBI / Proxy

Cabeçalhos consumidos (e removidos antes do encaminhamento)

Estes controlam o roteamento e nunca são encaminhados para o produtor.

Cabeçalho	Propósito
`3gpp-Sbi-Target-apiRoot`	Destino de roteamento direto do Modo 1.
`3gpp-Sbi-Discovery-target-nf-type`	Tipo de NF a ser descoberto (ex: `UDM`).
`3gpp-Sbi-Discovery-service-names`	Nomes de serviços separados por vírgula; o primeiro é o primário (usado como a chave de cache e serviço de consulta NRF).
`3gpp-Sbi-Discovery-requester-nf-type`	Tipo de NF do solicitante; delimita a consulta NRF.
`3gpp-Sbi-Discovery-target-plmn-list`	Lista de PLMN alvo; passada para a consulta NRF.
`3gpp-Sbi-Discovery-requester-snssai-list`	Lista de S-NSSAI do solicitante; passada para a consulta NRF.
`3gpp-Sbi-Discovery-nf-set-id`	Filtro de ID do conjunto NF; passado para a consulta NRF.
`3gpp-Sbi-Discovery-target-nf-instance-id`	ID da instância NF alvo específica; passado para a consulta NRF.
`3gpp-Sbi-Discovery-requester-nf-instance-id`	ID da instância do solicitante. Consumido e removido, mas não está atualmente incluído na consulta de descoberta NRF.

O cabeçalho `User-Agent` também é inspecionado: conforme TS 29.500 §5.2.2.2, ele começa com o tipo de NF do solicitante (ex: `AMF-...`), que é usado como o tipo de NF do solicitante se nenhum cabeçalho de descoberta explícito fornecer um. Se nenhum dos dois estiver presente, o SCP define o tipo de NF do solicitante como `SCP` para a consulta NRF.

Cabeçalhos produzidos

Cabeçalho	Propósito
`3gpp-Sbi-Producer-Id`	Adicionado à resposta, carregando o `nfInstanceId` do produtor que atendeu à requisição, para que o consumidor possa vincular requisições subsequentes ao mesmo produtor (TS 29.500 §6.10.3.3).
`3gpp-Sbi-Lci`	Informação de Controle de Carga do Produtor, retransmitida de ponta a ponta da resposta do produtor para que o consumidor receba sua dica de carga através do SCP (TS 29.500 §6.3). O SCP retransmite o valor do produtor e não sintetiza o seu próprio.
`3gpp-Sbi-Load`	O cabeçalho companheiro de carga `3gpp-Sbi-Lci`, retransmitido de ponta a ponta da resposta do produtor junto com `3gpp-Sbi-Lci` (TS 29.500 §6.3).

Cabeçalhos de resposta hop-by-hop (`transfer-encoding`, `connection`, `content-length`) não são encaminhados de volta ao consumidor.

Respostas de erro do proxy

Quando o SCP não consegue completar uma operação, ele retorna um corpo `ProblemDetails` conforme RFC 7807 por TS 29.500.

Status HTTP	Causa	Condição
`400 Bad Request`	`MANDATORY_IE_MISSING`	Sem informações de roteamento: sem target-apiRoot, sem cabeçalhos de descoberta, e o prefixo do caminho não mapeia para nenhum serviço conhecido.
`502 Bad Gateway`	`TARGET_NF_NOT_REACHABLE`	Nenhum URI SBI pôde ser resolvido a partir de um perfil NF descoberto.
`504 Gateway Timeout`	`NF_DISCOVERY_FAILURE`	A descoberta (NRF + cache) não gerou instâncias NF utilizáveis, incluindo quando **todas** as instâncias candidatas para o serviço estão administrativamente drenadas.
`500 Internal Server Error`	`SYSTEM_FAILURE`	Erro interno inesperado do proxy que não está coberto pelas causas mais específicas acima.

> **Nota sobre o caso de todos drenados.** Se todos os candidatos para um serviço estiverem drenados, o proxy não tem próximo salto utilizável e retorna `504 NF_DISCOVERY_FAILURE` (não `500`). Desdrene pelo menos uma instância para restaurar o serviço. Consulte Solução de Problemas.

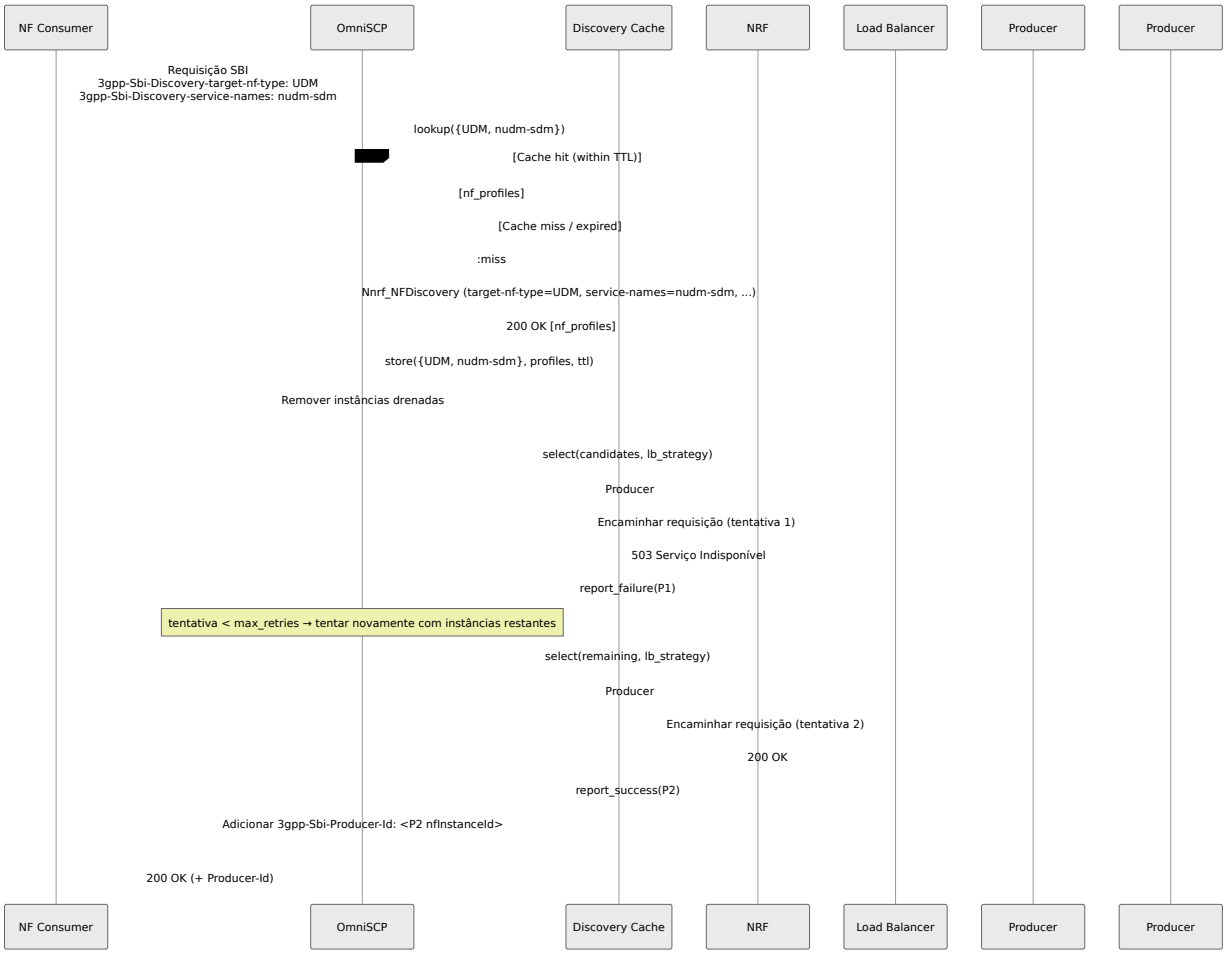
Semântica de tentativa novamente. Uma resposta de produtor com status **< 500** (incluindo 4xx) é tratada como uma resposta válida e retornada ao consumidor como está. Uma resposta **5xx** ou um **erro de conexão/timeout** conta como uma falha: a instância que falhou é reportada ao balanceador de carga e removida do conjunto de candidatos, e uma instância diferente é tentada. As tentativas continuam até que uma resposta não falha seja obtida,

`max_retries` seja alcançado ou a lista de candidatos seja esgotada, momento em que a última resposta (ou erro) é retornada.

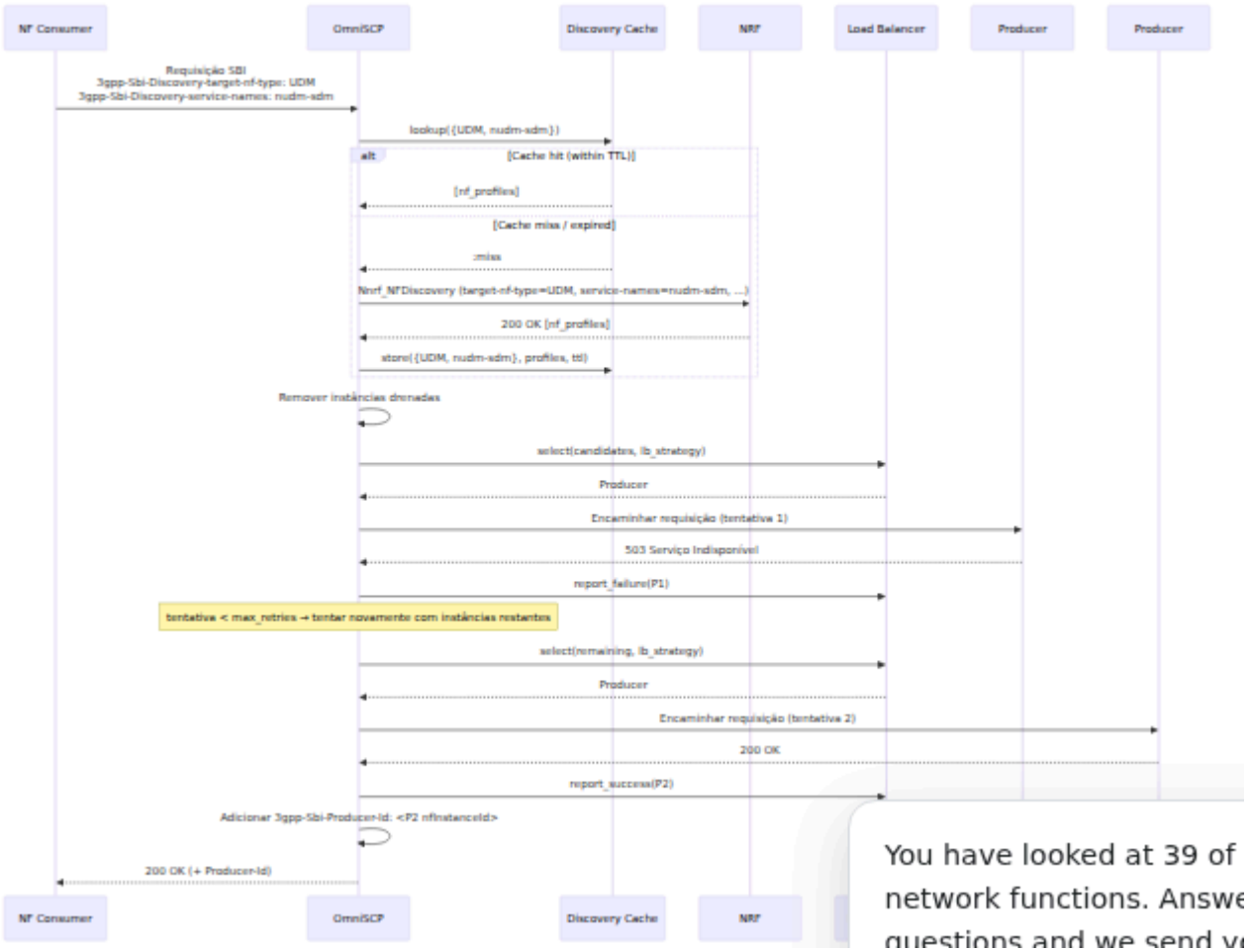
Procedimentos Chave

Requisição encaminhada, balanceada com uma tentativa novamente (Modo 2/3)

A sequência abaixo mostra a descoberta delegada, a seleção do balanceador de carga, uma primeira tentativa que falha com um 5xx e uma tentativa bem-sucedida contra uma segunda instância.

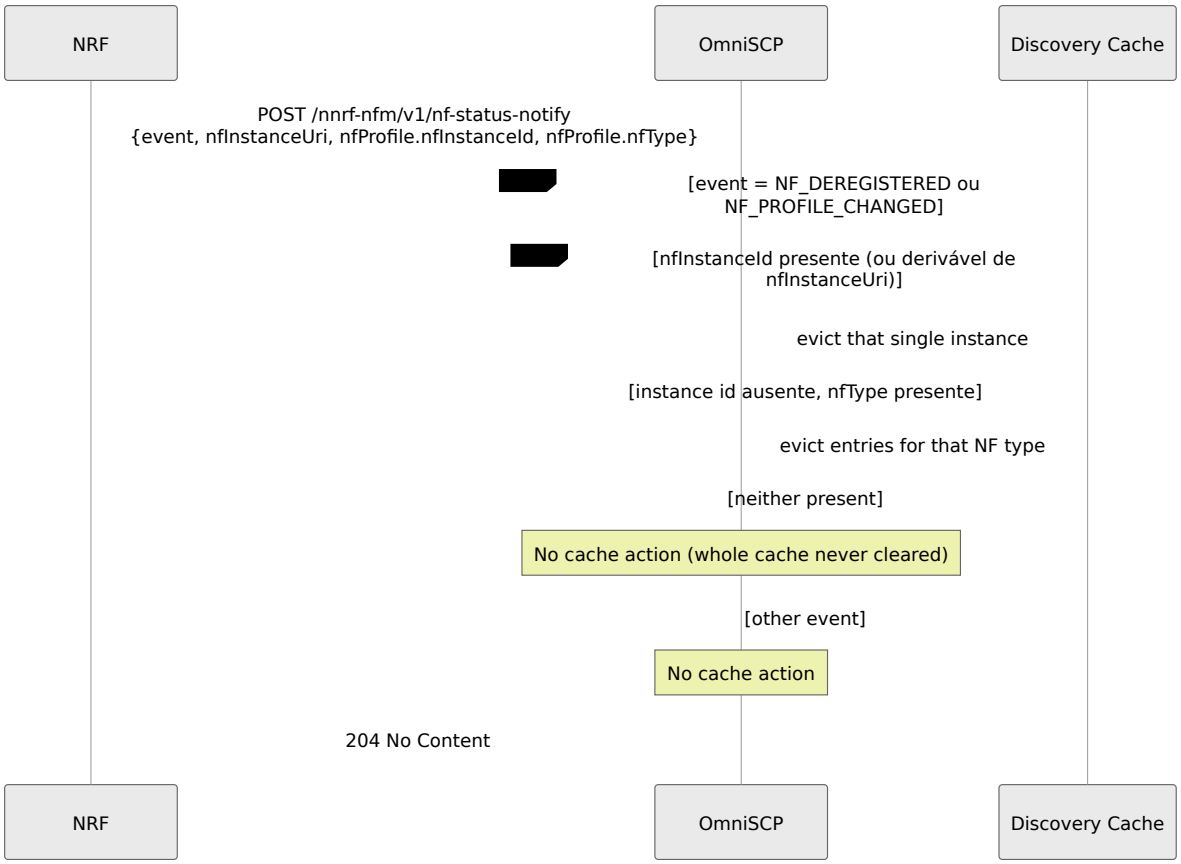


Encaminhamento direto (Modo 1)



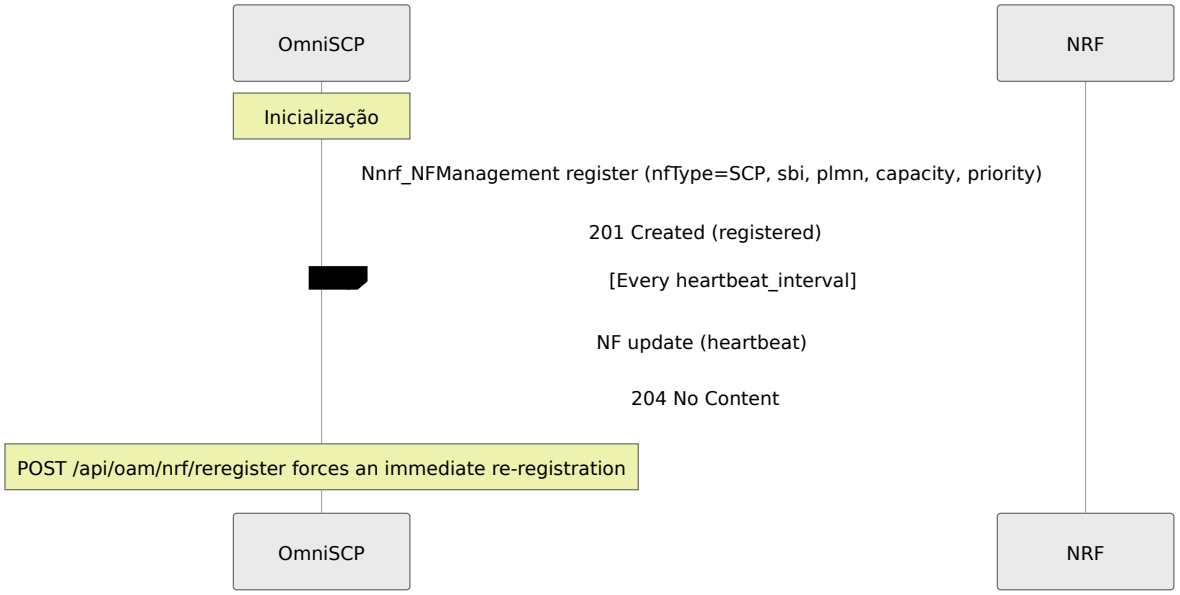
You have looked at 39 of
network functions. Answe
questions and we send yo

Notificação de status NRF (invalidação de cache)



Auto-registro e heartbeat NRF

O SCP se registra com o NRF como um NF `SCP` na inicialização e mantém o registro ativo com batimentos periódicos. O perfil anunciado carrega o endereço/porta/esquema SBI, o PLMN (`mcc`/`mnc`) e a `capacity`/`priority` do NF. O registro pode ser forçado imediatamente via a API OAM.

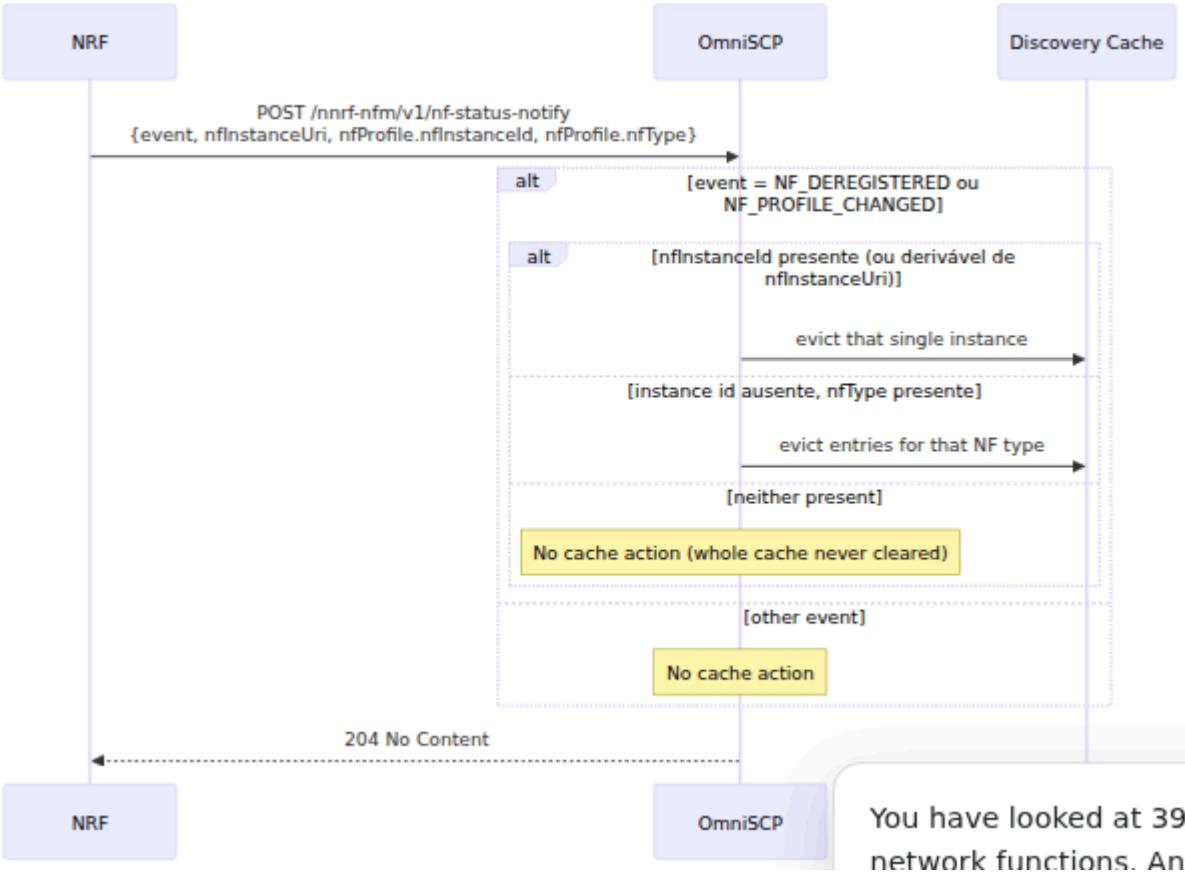


Drenagem administrativa upstream

A drenagem é um estado administrativo, separado do rastreamento de saúde baseado em falhas automático. Instâncias drenadas são filtradas da lista de candidatos *antes* da seleção do balanceador de carga, de modo que um produtor drenado nunca receba tráfego encaminhado. Use isso para retirar um produtor da rotação de forma graciosa (para manutenção) sem desregistrá-lo do NRF.

Fluxo de trabalho de manutenção. Obtenha o `nfInstanceId` do produtor alvo do NRF (o valor com o qual o produtor se registrou), em seguida, `POST /api/oam/upstream/drain` com esse ID e um `reason` opcional. Requisições em andamento já encaminhadas para o produtor não são interrompidas; apenas *novas* seleções o ignoram, então permita que requisições existentes sejam concluídas antes de atender o produtor. Confirme que ele está fora da rotação com `GET /api/oam/upstream/drain`, e quando a manutenção estiver concluída, `DELETE /api/oam/upstream/drain/{id}` para retorná-lo ao serviço.

> **O estado de drenagem é na memória.** As entradas de drenagem são mantidas em uma tabela ETS e **não são persistidas entre reinicializações do SCP**: se o processo SCP reiniciar enquanto um produtor estiver drenado, esse produtor se torna selecionável novamente e você deve reemitir a drenagem. Planeje janelas de manutenção de acordo e verifique novamente `GET /api/oam/upstream/drain` após qualquer reinicialização do SCP.



You have looked at 39 of
network functions. Answe

← Visão Geral

Solução de Problemas do OmniSCP

As causas de erro e os códigos de status referenciados aqui estão descritos em Operações → Respostas de erro do proxy. As chaves de configuração e os endpoints OAM estão em Configuração; métricas em Métricas.

400 Solicitação Inválida: MANDATORY_IE_MISSING

Sintomas: O consumidor recebe `400` com a causa `MANDATORY_IE_MISSING`.

Possíveis causas:

- A solicitação não continha `3gpp-Sbi-Target-apiRoot` nem o par `3gpp-Sbi-Discovery-target-nf-type` + `3gpp-Sbi-Discovery-service-names`.
- Dependendo da inferência baseada em caminho (Modo 3) para um prefixo de serviço que não está no mapa embutido.

Resolução:

1. Confirme se o consumidor envia um target-apiRoot ou ambos os cabeçalhos de descoberta.
2. Se depender da inferência de caminho, verifique se o prefixo do caminho está na tabela do Modo 3. Para um prefixo que não está nessa tabela, envie cabeçalhos de descoberta explícitos.

504 Timeout de Gateway: NF_DISCOVERY_FAILURE

Sintomas: O consumidor recebe `504` com a causa `NF_DISCOVERY_FAILURE`.

Possíveis causas:

- O NRF não retornou instâncias para o tipo de NF/serviço solicitado.
- O NRF está inacessível.

Resolução:

1. Verifique `nrf_uri` e a conectividade de rede com o NRF.
2. Confirme se o tipo de NF alvo está registrado no NRF.
3. Verifique se uma notificação de status recente do NRF invalidou o cache para o tipo de NF e o produtor não foi re-registrado.

504 Timeout de Gateway: NF_DISCOVERY_FAILURE (todas as instâncias drenadas)

Sintomas: O consumidor recebe `504` com a causa `NF_DISCOVERY_FAILURE` para um serviço que tem produtores registrados.

Possíveis causas:

- Cada instância candidata para o serviço está administrativamente drenada, então o proxy não tem um próximo salto utilizável. Esta condição mapeia para `504 NF_DISCOVERY_FAILURE`, o mesmo status que a descoberta que não resulta em instâncias.

Resolução:

1. Verifique `GET /api/oam/upstream/drain`. Se todas as instâncias para o serviço estiverem drenadas, desdrene as necessárias via `DELETE /api/oam/upstream/drain/{id}`.
2. Se nenhuma instância estiver drenada e o serviço estiver registrado, trate como um caso de falha de descoberta (veja a seção NF_DISCOVERY_FAILURE acima).

500 Erro Interno do Servidor: SYSTEM_FAILURE

Sintomas: O consumidor recebe `500` com a causa `SYSTEM_FAILURE`.

Possíveis causas:

- Um erro interno inesperado do proxy que não é coberto pelas causas mais específicas (`400`, `502`, `504`) acima.

Resolução:

1. Revise os logs do SCP para detalhes do erro interno.

502 Bad Gateway: TARGET_NF_NOT_REACHABLE

Sintomas: O consumidor recebe `502` com a causa `TARGET_NF_NOT_REACHABLE`.

Possíveis causas:

- Nenhum URI SBI pôde ser resolvido a partir de um perfil NF descoberto.
- Todos os produtores selecionados retornaram erros de conexão após tentativas.

Resolução:

1. Confirme se os produtores estão em execução e acessíveis nos endereços SBI em seus perfis NRF.
2. Se os produtores estiverem lentos, aumente `upstream_timeout`.
3. Revise `max_retries`; `0` transforma uma única falha em um erro imediato.
4. Verifique `GET /api/api/health` para instâncias marcadas como não saudáveis após falhas repetidas.

Timeouts de upstream / alta latência do proxy

Sintomas: `omni_scp_proxy_request_duration_ms` elevado; solicitações falham intermitentemente com resultado `error`.

Possíveis causas:

- Produtores respondendo mais lentamente que `upstream_timeout`.
- Baixa taxa de acerto de cache forçando uma viagem de ida e volta ao NRF em muitas solicitações.

Resolução:

1. Inspecione o histograma de latência e a taxa de erro por tipo de NF.
2. Compare `omni_scp_cache_hits_total` vs `omni_scp_cache_misses_total`; uma alta taxa de misses significa descoberta frequente do NRF, então considere aumentar `discovery_cache_ttl`.
3. Ajuste `upstream_timeout` para corresponder aos tempos de resposta realistas dos produtores. Lembre-se de que uma tentativa com timeout adiciona o timeout total à latência de ponta a ponta antes de uma nova tentativa.

Cache de roteamento / descoberta obsoleto

Sintomas: Solicitações são encaminhadas para produtores que se mudaram ou reiniciaram, causando erros até que as entradas expirem.

Possíveis causas:

- Um produtor mudou de endereço ou reiniciou sem se desregistrar do NRF, então o SCP mantém um perfil obsoleto até a expiração do TTL.

Resolução:

1. Reduza `discovery_cache_ttl` para encurtar a janela de obsolescência.
2. Certifique-se de que os produtores se desregistrem do NRF ao desligar, para que o SCP receba uma notificação `NF_DEREGISTERED` e invalide o tipo de NF afetado.
3. Limpe explicitamente via `DELETE /api/oam/cache/{nf_type}` (ou `DELETE /api/oam/cache/all`).

Registro do NRF não mantido

Sintomas: `omni_scp_nrf_registration_status` lê `0`.

Possíveis causas:

- `nrf_uri` incorreto ou NRF inacessível.
- Desvio de PLMN (`mcc`/`mnc`) em relação à configuração do NRF.

Resolução:

1. Verifique `nrf_uri` e conectividade.
2. Confirme se `mcc`/`mnc` correspondem à configuração PLMN do NRF.
3. Force um re-registro com `POST /api/oam/nrf/reregister` e revise os logs de inicialização.