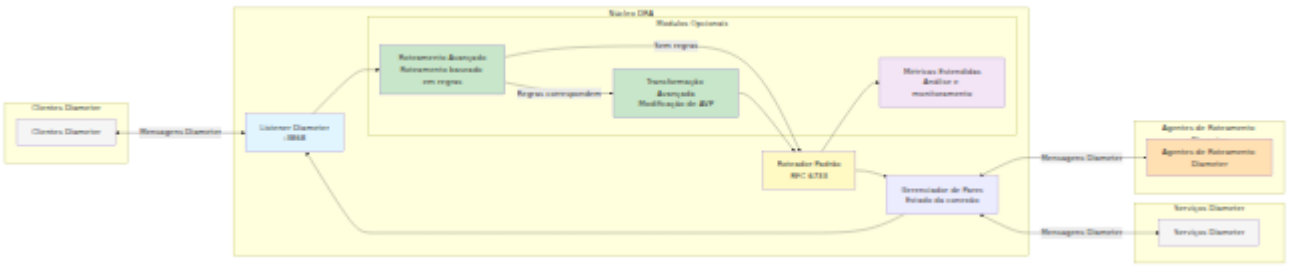
# Guia de Operações do DRA

## Índice



---

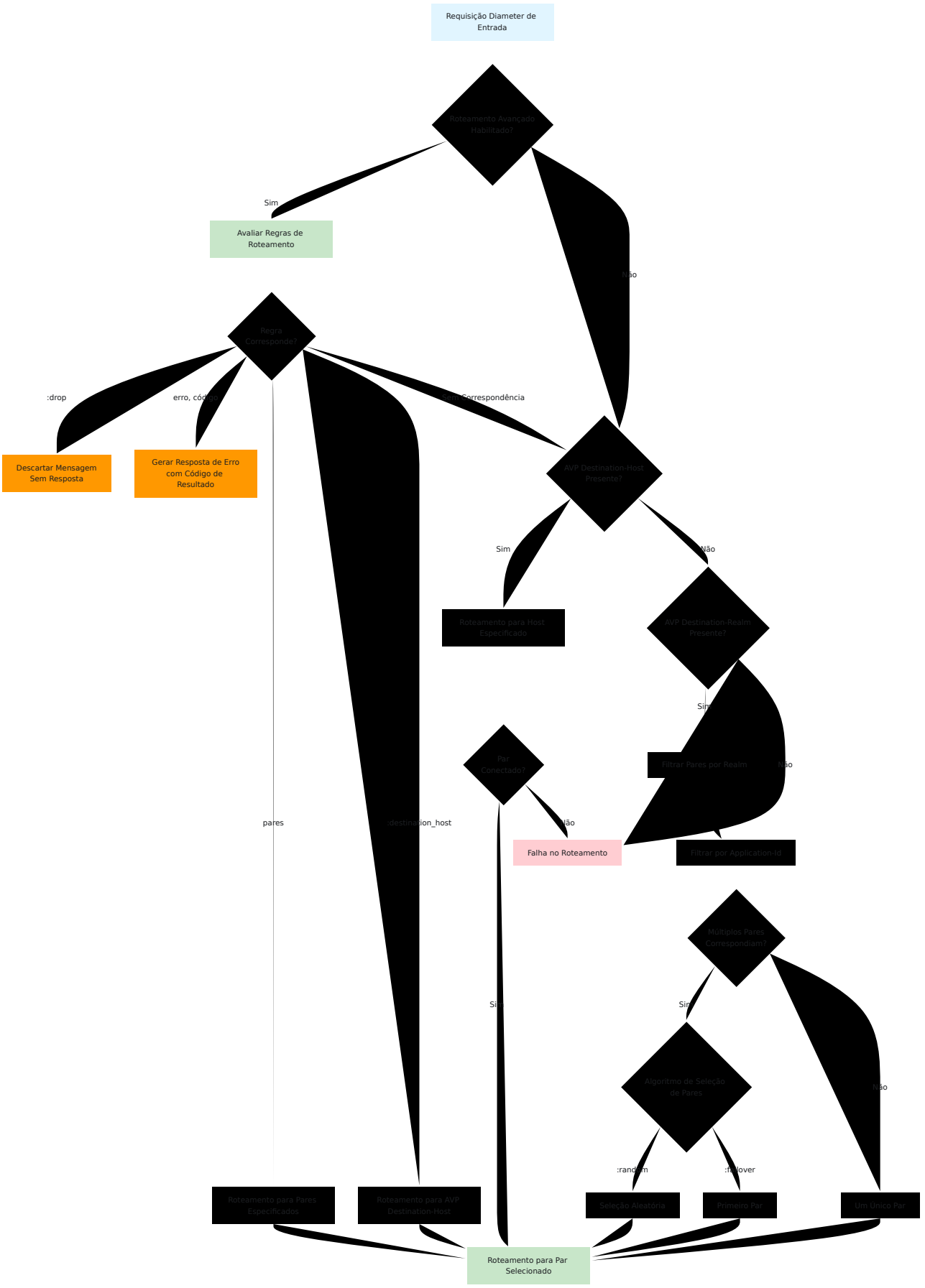
## Visão Geral da Arquitetura do DRA



---

# Roteamento Diameter Padrão

Sem os módulos Roteamento Avançado ou Transformação Avançada, o DRA realiza o roteamento Diameter padrão com base no Protocolo Base Diameter (RFC 6733):


Requisição Diameter de Entrada
Roteamento Avançado Habilitado?
Sim
Avaliar Regras de Roteamento
Não
Regra Corresponde?
:drop
erro, código
Sem Correspondência
Descartar Mensagem Sem Resposta
Gerar Resposta de Erro com Código de Resultado
AVP Destination-Host Presente?
Sim
Não
Roteamento para Host Especificado
AVP Destination-Realm Presente?
Sim
Não
Filtrar Pares por Realm
Filtrar por Application-Id
Par Conectado?
Não
Falha no Roteamento
pares
:destination_host
Sim
Múltiplos Pares Correspondiam?
Sim
Não
Algoritmo de Seleção de Pares
:random
:failover
Seleção Aleatória
Primeiro Par
Um Único Par
Roteamento para Pares Especificados
Roteamento para AVP Destination-Host
Roteamento para Par Selecionado

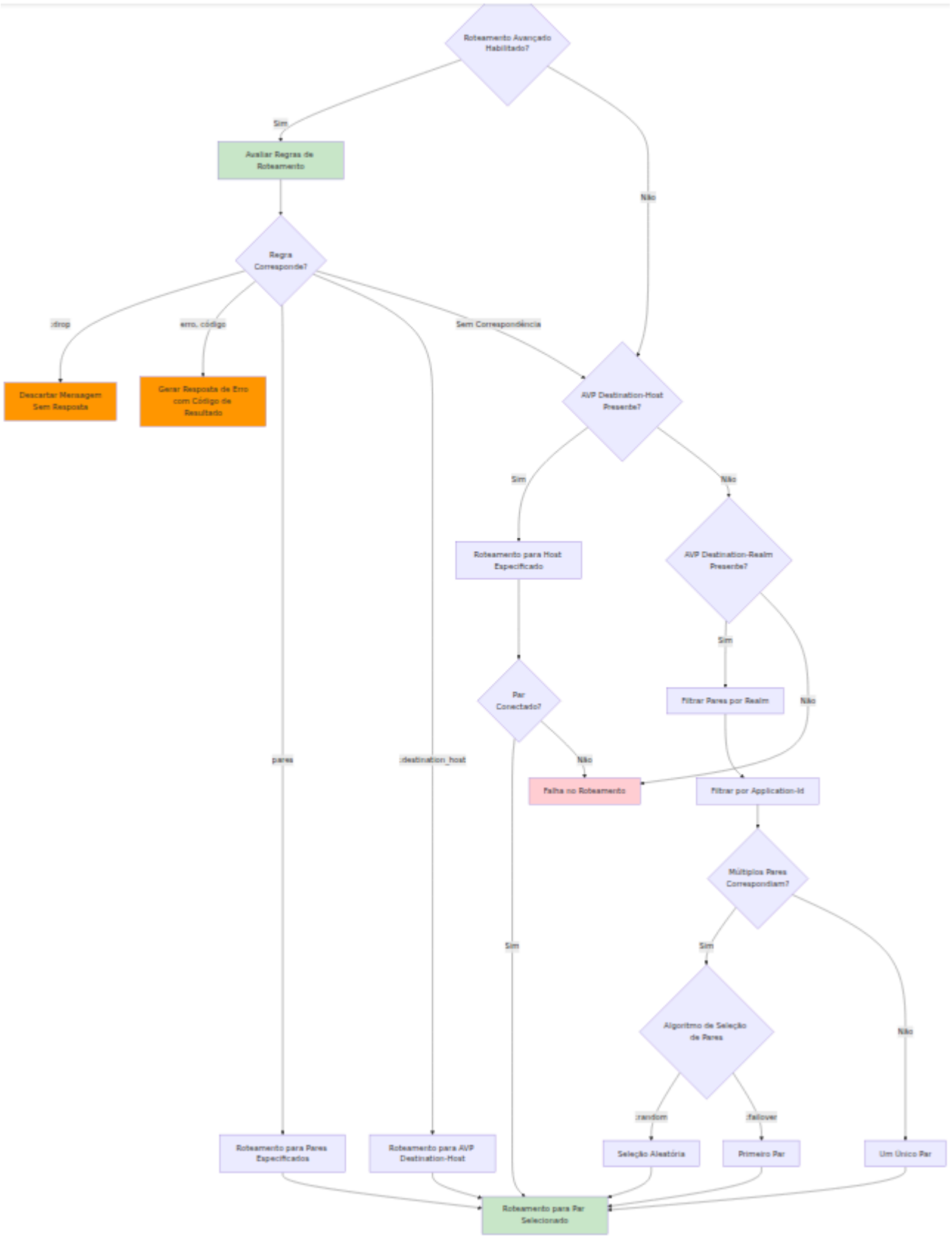
# Roteamento de Requisições

O DRA roteia mensagens de requisição usando um mecanismo baseado em prioridade definido na Seção 6.1 da RFC 6733:

1. **AVP Destination-Host (293)** - Se presente, o DRA roteia diretamente para o par especificado
    - Este é o mecanismo de roteamento de maior prioridade
    - Se o par não estiver conectado, o roteamento falha
    - Fornece controle de roteamento explícito em nível de host
2. **AVP Destination-Realm (283)** - Se o Destination-Host estiver ausente, roteia com base no realm
    - O DRA seleciona um par conectado que anuncia suporte para o realm de destino
    - O balanceamento de carga é aplicado quando múltiplos pares correspondem ao realm
    - O roteamento baseado em realm permite flexibilidade entre múltiplos hosts
3. **Application-Id** - Os pares são filtrados por aplicações Diameter suportadas
    - Apenas pares que anunciam suporte para o Application-Id da mensagem são considerados
    - Baseado na Troca de Capacidades (CER/CEA) durante o estabelecimento da conexão do par
    - Veja IDs de Aplicação 3GPP Comuns para referência

# Roteamento de Respostas

Pacotes de resposta usam um mecanismo de roteamento fundamentalmente diferente do das requisições:




- **Roteamento baseado em sessão**: Pacotes de resposta sempre seguem o caminho reverso da requisição

- **Preservação do ID de ponta a ponta**: O Identificador de Ponta a Ponta permanece inalterado em todos os saltos
- **Roteamento hop-by-hop**: O DRA usa o Identificador Hop-by-Hop para manter o estado de roteamento (muda a cada salto)
- **Sem avaliação de regras**: O DRA não avalia regras de roteamento ou conteúdos de AVP para respostas
- **Correlação com estado**: Tabelas de roteamento internas rastreiam qual par enviou cada requisição

**Por que as respostas não são roteadas por módulos avançados:**

- O roteamento de respostas é determinístico e deve retornar ao par de origem
- O protocolo Diameter requer que as respostas sigam o caminho de requisição estabelecido
- As decisões de roteamento para respostas são feitas com base no contexto da requisição original, não no conteúdo da resposta
- Isso garante o gerenciamento adequado da sessão e previne loops de roteamento

Veja Seção 6.2 da RFC 6733 para detalhes sobre o roteamento de mensagens de resposta.

## Seleção de Pares

Quando múltiplos pares correspondem aos critérios de roteamento, o `peer_selection_algorithm` configurado determina a seleção:

- `:random` - Seleciona aleatoriamente entre os pares disponíveis (padrão)
- `:failover` - Sempre seleciona o primeiro par da lista (baseado em prioridade)
- Os pares devem estar em **estado conectado** para serem selecionados
- Pares desconectados ou inativos são automaticamente excluídos

## Limitações do Roteamento Padrão

- Sem regras de roteamento personalizadas baseadas em valores de AVP (por exemplo, padrões de IMSI)
- Sem tradução de realm ou modificação de AVP
- Não é possível roteamento com base no par de origem
- Controle limitado sobre a distribuição de tráfego

Os módulos Roteamento Avançado e Transformação Avançada estendem esse comportamento padrão com capacidades de roteamento baseado em regras e manipulação de pacotes.

---

# Configuração Base do DRA

O DRA requer configuração base definindo sua identidade, configurações de rede e conexões de pares. Esta configuração estabelece a base para todas as operações de roteamento.

## Estrutura da Configuração

```
%{
  host: "dra01.example.com",
  realm: "example.com",
  listen_ip: "192.168.1.10",
  listen_port: 3868,
  service_name: :example_dra,
  product_name: "OmniDRA",
  vendor_id: 10415,
  request_timeout: 5000,
  peer_selection_algorithm: :random,
  allow_undefined_peers_to_connect: false,
  log_unauthorized_peer_connection_attempts: true,
  peers: [
    # Configurações de pares...
  ]
}
```

## Parâmetros de Identidade do DRA

| Parâmetro | Tipo | Descrição |
|---|---|---|
| `host` | String | A Identidade Diameter do DRA (nome de domínio totalmente qualificado) |
| `realm` | String | O realm Diameter do DRA |
| `product_name` | String | Nome do produto anunciado nas mensagens CER/CEA |
| `vendor_id` | Integer | Vendor-ID conforme definido na Seção 5.3.3 da RFC 6733 (10415 = 3GPP) |

## Configurações de Rede

| Parâmetro | Tipo | Descrição |
|---|---|---|
| `listen_ip` | String ou Lista | Endereço(s) IP que o DRA escuta. Para multihoming SCTP, use uma lista de strings IP (veja Multihoming SCTP) |
| `listen_port` | Integer | Porta TCP/SCTP para conexões Diameter (padrão: 3868) |
| `service_name` | Atom | Identificador de serviço interno Erlang |
| `request_timeout` | Integer | Timeout em milissegundos para pares de requisição/resposta (padrão: 5000) |

## Configurações de Seleção de Pares

| Parâmetro | Tipo | Descrição |
|---|---|---|
| `peer_selection_algorithm` | Atom | Algoritmo de balanceamento de carga: `:random` (seleção aleatória) ou `:failover` (prioridade do primeiro par) |
| `allow_undefined_peers_to_connect` | Boolean | Permitir conexões de pares não configurados (padrão: `false`) |
| `log_unauthorized_peer_connection_attempts` | Boolean | Registrar tentativas de conexão de pares não autorizados |

## Configuração de Pares

Cada par na lista `peers` define uma conexão Diameter:

```
%{
  host: "mme01.operator.com",
  realm: "operator.com",
  ip: "192.168.1.20",
  port: 3868,
  transport: :diameter_tcp,
  tls: false,
  initiate_connection: false
}
```

**Parâmetros de Par**

| Parâmetro | Tipo | Descrição |
|---|---|---|
| `host` | String | Identidade Diameter do par (FQDN) - deve corresponder exatamente para roteamento |
| `realm` | String | Realm Diameter do par |
| `ip` | String | Endereço IP principal do par para conexão (obrigatório) |
| `ips` | Lista | Lista de endereços IP para multihoming SCTP (opcional, veja Multihoming SCTP) |
| `port` | Integer | Porta Diameter do par (tipicamente 3868) |
| `transport` | Atom | Protocolo de transporte: `:diameter_tcp` ou `:diameter_sctp` |
| `tls` | Boolean | Habilitar criptografia TLS (se `true`, tipicamente use a porta 3869) |
| `initiate_connection` | Boolean | `true`: DRA conecta-se ao par, `false`: DRA aguarda o par conectar |

# Modos de Conexão

### Iniciar Conexão (`initiate_connection: true`)

- O DRA atua como cliente Diameter
- O DRA inicia a conexão TCP/SCTP com o par
- Usado para conectar-se ao HSS, PCRF ou outros sistemas de backend
- O DRA tentará reconectar se o par estiver inacessível

### Aceitar Conexão (`initiate_connection: false`)

- O DRA atua como servidor Diameter
- O DRA aguarda o par conectar
- Usado para conexões MME, SGSN, P-GW
- O par deve estar na configuração ou `allow_undefined_peers_to_connect: true`

## Exemplo de Configuração

```
%{
  host: "dra01.mvno.example.com",
  realm: "mvno.example.com",
  listen_ip: "10.100.1.10",
  listen_port: 3868,
  service_name: :mvno_dra,
  product_name: "OmniDRA",
  vendor_id: 10415,
  request_timeout: 5000,
  peer_selection_algorithm: :random,
  allow_undefined_peers_to_connect: false,
  log_unauthorized_peer_connection_attempts: true,
  peers: [
    # MME - aguarda o MME conectar
    %{
      host: "mme01.operator.example.com",
      realm: "operator.example.com",
      ip: "10.100.2.15",
      port: 3868,
      transport: :diameter_sctp,
      tls: false,
      initiate_connection: false
    },
    # HSS - DRA inicia conexão
    %{
      host: "hss01.mvno.example.com",
      realm: "mvno.example.com",
      ip: "10.100.3.141",
      port: 3868,
      transport: :diameter_tcp,
      tls: false,
      initiate_connection: true
    },
    # PCRF com TLS - DRA inicia conexão segura
    %{
      host: "pcrf01.mvno.example.com",
      realm: "mvno.example.com",
      ip: "10.100.3.22",
      port: 3869,
      transport: :diameter_tcp,
      tls: true,
```

```
      initiate_connection: true
    }
  ]
}
```

## Notas Importantes

- **Correspondência de Nome de Host**: Nomes de host de pares nas regras de Roteamento Avançado devem corresponder exatamente ao valor `host` configurado aqui (sensível a maiúsculas e minúsculas)
- **Troca de Capacidades**: Na conexão, os pares trocam aplicações suportadas via mensagens CER/CEA
- **Suporte a Aplicações**: O DRA anuncia todas as aplicações 3GPP suportadas (veja IDs de Aplicação 3GPP Comuns)
- **Vendor-ID 10415**: Valor padrão para aplicações 3GPP
- **Timeout de Requisição**: Afeta o TTL das Métricas Estendidas (timeout + 5 segundos)
- **Seleção de Pares**: Quando múltiplos pares correspondem aos critérios de roteamento, o `peer_selection_algorithm` determina qual é escolhido

## Considerações de Segurança

- Defina `allow_undefined_peers_to_connect: false` em produção
- Habilite `log_unauthorized_peer_connection_attempts: true` para monitoramento de segurança
- Certifique-se de que as regras de firewall correspondam às configurações de `listen_ip` e `listen_port`
- Valide certificados de pares ao usar TLS

---

# Multihoming SCTP

O multihoming SCTP fornece redundância de rede permitindo que os pontos finais se vinculem a múltiplos endereços IP. Se o caminho de rede primário

falhar, o SCTP automaticamente muda para um caminho alternativo sem interromper a sessão Diameter.

## Como Funciona


- Os heartbeats SCTP monitoram todos os caminhos de rede
- A troca automática ocorre se o caminho primário se tornar inacessível
- Sem interrupção da sessão Diameter durante a troca de caminho
- O kernel gerencia a seleção de caminho automaticamente

## Configuração

### Endereços de Escuta do DRA

Configure múltiplos endereços IP locais para o DRA se vincular:

```
%{
  # IP único (compatível com versões anteriores)
  listen_ip: "192.168.1.10",

  # Múltiplos IPs para multihoming SCTP
  listen_ip: ["192.168.1.10", "10.0.0.10"],

  listen_port: 3868,
  ...
}
```

**Notas:**

- O transporte TCP usa apenas o primeiro IP da lista
- O transporte SCTP se vincula a todos os IPs especificados
- O formato de string IP único permanece totalmente suportado

**Configuração de Par**

Configure múltiplos endereços IP remotos para conexões de pares:

```
peers: [
  %{
    host: "hss01.example.com",
    realm: "example.com",
    ip: "192.168.1.20",                          # IP Primário
(obrigatório)
    additional_ips: ["192.168.1.20", "10.0.0.20"],        # Todos
os IPs para multihoming
    port: 3868,
    transport: :diameter_sctp,
    tls: false,
    initiate_connection: true
  }
]
```

**Notas:**

- O campo `ip` é obrigatório para compatibilidade com versões anteriores
- O campo `ips` é opcional; se omitido, apenas `ip` é usado
- Para SCTP, inclua o IP primário na lista `ips`
- Para TCP, apenas `ip` é usado (TCP não suporta multihoming)

## Exemplo Completo

```
config :dra,
  diameter: %{
    service_name: :omnitouch_dra,
    listen_ip: ["192.168.1.10", "10.0.0.10"],  # DRA com
multihoming
    listen_port: 3868,
    host: "dra01",
    realm: "example.com",
    product_name: "OmniDRA",
    vendor_id: 10415,
    request_timeout: 5000,
    peer_selection_algorithm: :random,
    allow_undefined_peers_to_connect: false,
    peers: [
      # Conexão HSS com multihoming
      %{
        host: "hss01.example.com",
        realm: "example.com",
        ip: "192.168.1.20",
        additional_ips: ["192.168.1.20", "10.0.0.20"],
        port: 3868,
        transport: :diameter_sctp,
        tls: false,
        initiate_connection: true
      },
      # MME com IP único (compatível com versões anteriores)
      %{
        host: "mme01.example.com",
        realm: "example.com",
        ip: "192.168.1.30",
        port: 3868,
        transport: :diameter_sctp,
        tls: false,
        initiate_connection: false
      }
    ]
  }
```

## Requisitos

- O módulo do kernel SCTP deve estar carregado (pacote `lksctp-tools` no Linux)
- Todos os endereços IP devem ser roteáveis de/para o par
- As regras de firewall devem permitir tráfego SCTP em todos os IPs configurados
- Ambos os pontos finais devem ser configurados para multihoming para redundância total

## Limitações

- O transporte TCP não suporta multihoming (usa apenas o IP primário)
- TLS sobre multihoming SCTP pode ter limitações de compatibilidade
- O tempo de troca de caminho depende dos parâmetros SCTP do kernel

---

# Tabelas de Referência

## IDs de Aplicação 3GPP Comuns

| Application-Id | Interface | Descrição |
|---|---|---|
| 16777251 | S6a/S6d | MME/SGSN para HSS autenticação e dados de assinatura |
| 16777252 | S13/S13' | MME para verificação de identidade de equipamento EIR |
| 16777238 | Gx | PCEF para controle de políticas e cobrança PCRF |
| 16777267 | S9 | Política de roaming de PCRF doméstico para PCRF visitado |
| 16777272 | Sy | PCRF para vinculação de sessão OCS |
| 16777216 | Cx | I-CSCF/S-CSCF para registro IMS HSS |
| 16777217 | Sh | AS para dados de usuário IMS HSS |
| 16777236 | SLg | MME/SGSN para serviços de localização GMLC |
| 16777291 | SLh | GMLC para informações de assinante de localização HSS |
| 16777302 | S6m | MTC-IWF para HSS/HLR para dispositivos M2M |
| 16777308 | S6c | SMS-SC/IP-SM-GW para roteamento de SMS HSS |

| Application-Id | Interface | Descrição |
|---|---|---|
| 16777343 | S6t | SCEF para eventos de monitoramento HSS |
| 16777334 | Rx | AF para autorização de mídia PCRF |

## Códigos AVP Comuns

| Código | Nome AVP | Tipo | Uso |
|---|---|---|---|
| 1 | User-Name | UTF8String | Identificador do assinante (IMSI em 3GPP) |
| 264 | Origin-Host | DiameterIdentity | Nome do host do par de origem |
| 268 | Result-Code | Unsigned32 | Código de resultado padrão |
| 283 | Destination-Realm | DiameterIdentity | Realm de destino |
| 293 | Destination-Host | DiameterIdentity | Host de destino (opcional) |
| 296 | Origin-Realm | DiameterIdentity | Realm de origem |
| 297 | Experimental-Result | Grouped | Código de resultado específico do fornecedor |

## Códigos de Comando Comuns

Os códigos de comando fazem parte do cabeçalho da mensagem Diameter, não dos AVPs:

| Código | Nome do Comando | Descrição |
|---|---|---|
| 257 | CER/CEA | Capabilities-Exchange-Request/Answer |
| 258 | RAR/RAA | Re-Auth-Request/Answer |
| 274 | ASR/ASA | Abort-Session-Request/Answer |
| 275 | STR/STA | Session-Termination-Request/Answer |
| 280 | DWR/DWA | Device-Watchdog-Request/Answer |
| 282 | DPR/DPA | Disconnect-Peer-Request/Answer |
| 316 | ULR/ULA | Update-Location-Request/Answer (S6a) |
| 317 | CLR/CLA | Cancel-Location-Request/Answer (S6a) |
| 318 | AIR/AIA | Authentication-Information-Request/Answer (S6a) |
| 321 | PUR/PUA | Purge-UE-Request/Answer (S6a) |

---

# Módulo de Roteamento Avançado

O módulo de Roteamento Avançado fornece capacidades flexíveis de roteamento de mensagens baseadas em regras com suporte para condições de correspondência complexas.

**Importante**: Este módulo avalia **apenas pacotes de requisição Diameter de entrada** (não pacotes de resposta). Pacotes de resposta seguem o roteamento de sessão estabelecido de volta ao par de origem - veja Roteamento de Resposta para detalhes.

# Configuração

Habilite o módulo e defina regras de roteamento em sua configuração:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: <identificador_da_regra>
      match: <escopo_de_correspondência>
      filters: [<lista_de_filtros>]
      route:
        peers: [<lista_de_pares>]
```

# Parâmetros

| Parâmetro | Descrição |
|---|---|
| `enabled` | Defina como `True` para ativar o módulo |
| `rule_name` | Identificador único para a regra de roteamento |
| `match` | Como os filtros são combinados: `:all` (lógica AND - todos os filtros devem corresponder), `:any` (lógica OR - pelo menos um filtro deve corresponder), `:none` (lógica NOR - nenhum filtro pode corresponder) |
| `filters` | Lista de condições de filtro (veja Filtros Disponíveis) |
| `route` | Ação de roteamento (veja Ações de Roteamento abaixo) |

# Ações de Roteamento

O parâmetro `route` suporta múltiplas ações:

### Roteamento para Pares

```
route:
  peers: [peer01.example.com, peer02.example.com]
```

Roteia para nomes de host de pares especificados. Os pares devem ser:

- Definidos na configuração de pares Diameter do DRA
- O nome do host exato conforme configurado (sensível a maiúsculas e minúsculas)
- Atualmente conectados para que o roteamento seja bem-sucedido (pares desconectados são ignorados)

**Roteamento para AVP Destination-Host**

```
route: :destination_host
```

Roteia para o par especificado no AVP Destination-Host (293) da mensagem. Se o AVP Destination-Host estiver ausente, o roteamento volta ao comportamento normal.

**Descartar Tráfego**

```
route: :drop
```

Descarta silenciosamente a mensagem sem enviar qualquer resposta. Use para:

- Filtragem de tráfego e blackholing
- Bloqueio de requisições indesejadas
- Limitação de taxa descartando tráfego excessivo

**Comportamento**:

- A mensagem é descartada no DRA (não é encaminhada)
- Nenhuma mensagem de resposta é enviada ao par solicitante
- Implementa o comportamento `:discard` do Diameter Erlang

- Métrica: `diameter_advanced_routing_drop_count_total` (veja Métricas Prometheus)

#### Gerar Resposta de Erro

```
route: {:error, 3004}
```

Gera uma resposta de erro Diameter com o Código de Resultado especificado e a envia de volta ao par solicitante. Códigos de resultado comuns:

- `3002` - DIAMETER_UNABLE_TO_DELIVER (roteamento indisponível)
- `3003` - DIAMETER_REALM_NOT_SERVED (realm não suportado)
- `3004` - DIAMETER_TOO_BUSY (proteção contra sobrecarga, limitação de taxa)
- `5012` - DIAMETER_UNABLE_TO_COMPLY (rejeição geral)

**Comportamento**:

- O DRA gera uma resposta de erro com o Código de Resultado especificado
- A resposta inclui Origin-Host, Origin-Realm, Session-Id (auto-populado pelo Diameter)
- A mensagem NÃO é encaminhada para nenhum par
- Implementa `{:protocol_error, code}` do Diameter Erlang (equivalente a `{:answer_message, code}`)
- Métrica: `diameter_advanced_routing_error_count_total` (veja Métricas Prometheus)

## Filtros Disponíveis

#### Filtros Padrão

Disponíveis tanto no Roteamento Avançado quanto na Transformação Avançada:

- **`:application_id`** - Corresponder ao ID da aplicação Diameter (veja referência de ID de Aplicação)

- Valor único: `{:application_id, 16777251}` (S6a/S6d)
- Múltiplos valores: `{:application_id, [16777251, 16777252]}` (S6a ou S6b)

- **`:command_code`** - Corresponder ao código de comando Diameter
  - Valor único: `{:command_code, 318}` (requisição AIR)
  - Múltiplos valores: `{:command_code, [317, 318]}` (ULR ou AIR)

- **`:avp`** - Corresponder ao valor de AVP (veja referência de código AVP)
  - Correspondência exata: `{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}`
  - Correspondência regex: `{:avp, {1, ~r"999001.*"}}`
  - Múltiplos padrões: `{:avp, {1, ["505057001313606", ~r"999001.*", ~r"505057.*"]}}`
  - Qualquer valor (verificação de presença): `{:avp, {264, :any}}`

**Filtro Específico de Roteamento**

Disponível apenas no Roteamento Avançado:

- **`:via_peer`** - Corresponder ao par de onde a requisição foi recebida
  - Um único par: `{:via_peer, "omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org"}`
  - Múltiplos pares: `{:via_peer, ["omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org", "omnitouch-lab-dra02.epc.mnc001.mcc001.3gppnetwork.org"]}`
  - Qualquer par: `{:via_peer, :any}`

**Filtros Específicos de Transformação**

Disponíveis apenas na Transformação Avançada:

- **`:to_peer`** - Corresponder ao par de destino predeterminado (apenas pacotes de requisição)
  - Um único par: `{:to_peer, "dra01.omnitouch.com.au"}`

    - Múltiplos pares: `{:to_peer, ["dra01.omnitouch.com.au", "dra02.omnitouch.com.au"]}`

- **`:from_peer`** - Corresponder ao par que enviou a resposta (apenas pacotes de resposta)
    - Um único par: `{:from_peer, "hss-01.example.com"}`
    - Múltiplos pares: `{:from_peer, ["hss-01.example.com", "hss-02.example.com"]}`

- **`:packet_type`** - Corresponder à direção do pacote
    - Requisição: `{:packet_type, :request}`
    - Resposta: `{:packet_type, :answer}`

**Notas Importantes sobre Filtros**

- **Filtros AVP**: Recomendados apenas para AVPs simples (User-Name, Origin-Host, Destination-Realm, etc.)
    - AVPs agrupados **não são suportados** e não corresponderão
    - Valores binários complexos **não são suportados**
    - Use o formato: `{:avp, {code, value}}`

- **Operadores de Lista**: Suportados para todos os valores de filtro, exceto `:packet_type`
    - Quando uma lista é usada, aplica-se **lógica OR** dentro da lista
    - Exemplo: `{:command_code, [317, 318]}` corresponde ao código de comando 317 **OU** 318

- **Valores Especiais**:
    - `:any` - Corresponde a qualquer valor (verifica a presença do AVP)
    - Exemplo: `{:avp, {264, :any}}` corresponde se o AVP Origin-Host existir com qualquer valor

# Exemplos de Roteamento

**Exemplo 1: Roteamento via Par**

Roteie mensagens com base em qual DRA elas chegaram:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: temporary_until_cutover_s6a_via_to_local_hss
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
        - '{:via_peer, ["omnitouch-lab-
dra01.epc.mnc001.mcc001.3gppnetwork.org", "omnitouch-lab-
dra02.epc.mnc001.mcc001.3gppnetwork.org"]}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
      route:
        peers: [omnitouch-lab-
hss01.epc.mnc001.mcc001.3gppnetwork.org, omnitouch-lab-
hss02.epc.mnc001.mcc001.3gppnetwork.org]
```

**Como funciona:** Roteia tráfego S6a que chega via pares DRA específicos para nós HSS locais.

**Exemplo 2: Roaming Inbound com Correspondência de Padrão**

Roteie tráfego de roaming com base em padrões de IMSI:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: inbound_s6a_roaming_to_dcc
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
        - '{:avp, {1, ["505571234567", ~r"999001.*"]}}'
      route:
        peers: [dra01.omnitouch.com.au, dra02.omnitouch.com.au]
```

**Como funciona:** Roteia mensagens S6a de um Realm de Origem específico com padrões de IMSI correspondentes para pares DRA designados.

**Exemplo 3: Roteamento Dinâmico com :destination_host**

Roteie para o valor do AVP Destination-Host na mensagem:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: route_to_specified_destination_host
      match: ":all"
      filters:
        - '{:avp, {1, [~r"90199.*"]}}'  # Corresponder ao padrão
IMSI
      route: :destination_host
```

**Como funciona:**

- Quando os filtros correspondem, roteia para o par especificado no AVP Destination-Host (293)
- Se o AVP Destination-Host estiver ausente, a correspondência é considerada uma falha e volta ao roteamento normal
- Útil para honrar o roteamento quando o remetente especifica o destino exato

**Exemplo 4: Descartar Tráfego Indesejado**

Descarte tráfego de intervalos de IMSI específicos:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: drop_test_subscribers
      match: ":all"
      filters:
        - '{:application_id, 16777251}'  # S6a
        - '{:avp, {1, [~r"999999.*"]}}'  # Intervalo de IMSI de
teste
      route: :drop
```

**Como funciona:**

- Corresponde a mensagens S6a com IMSI começando com 999999
- Silenciosamente descarta a mensagem sem enviar qualquer resposta
- Útil para filtrar tráfego de teste ou bloquear intervalos de assinantes específicos
- Veja Métricas Prometheus para monitoramento de tráfego descartado

**Exemplo 5: Limitação de Taxa com Respostas de Erro**

Retorne DIAMETER_TOO_BUSY para padrões de tráfego específicos:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: rate_limit_high_volume_peer
      match: ":all"
      filters:
        - '{:via_peer, "mme-overloaded-01.example.com"}'
        - '{:application_id, 16777251}'
      route: {:error, 3004}
```

**Como funciona:**

- Corresponde ao tráfego S6a de um par sobrecarregado específico
- Retorna a resposta de erro DIAMETER_TOO_BUSY (3004)
- O par solicitante recebe um erro e deve recuar
- Útil para proteção contra sobrecarga e limitação de taxa
- Veja Métricas Prometheus para monitoramento de respostas de erro

**Exemplo 6: Respostas de Erro Condicionais por Comando**

Bloqueie tipos de comando específicos com códigos de erro apropriados:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: block_purge_requests
      match: ":all"
      filters:
        - '{:application_id, 16777251}'  # S6a
        - '{:command_code, 321}'         # PUR (Purge-UE-Request)
      route: {:error, 5012}
```

**Como funciona:**

- Corresponde a mensagens de requisição S6a Purge-UE-Request
- Retorna erro DIAMETER_UNABLE_TO_COMPLY (5012)
- Bloqueia operações específicas sem descartar o tráfego silenciosamente
- Útil para desabilitar seletivamente certos comandos Diameter

---

# Módulo de Transformação Avançada

O módulo de Transformação Avançada permite a modificação dinâmica de AVPs de mensagens Diameter com base em critérios de correspondência. Veja Processamento de Regras para detalhes sobre como as regras são avaliadas.

## Configuração

Habilite o módulo e defina regras de transformação:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: <identificador_da_regra>
      match: <escopo_de_correspondência>
      filters: [<lista_de_filtros>]
      transform:
        action: <ação_de_transformação>
        avps: [<modificações_de_avp>]
```

## Parâmetros

| Parâmetro | Descrição |
|---|---|
| `enabled` | Defina como `True` para ativar o módulo |
| `rule_name` | Identificador único para a regra de transformação |
| `match` | Como os filtros são combinados: `:all` (lógica AND), `:any` (lógica OR), `:none` (lógica NOR) - veja Lógica de Filtro |
| `filters` | Lista de condições de filtro (veja Filtros Disponíveis) |
| `transform.action` | Tipo de transformação (`:edit`, `:remove` ou `:overwrite`) |
| `transform.avps` | Lista de modificações de AVP a serem aplicadas (veja referência de código AVP) |

## Ações de Transformação

### Pacotes de Requisição (Requisições Diameter)

- **`:edit`** - Modificar valores de AVP existentes
  - Apenas modifica AVPs que existem na mensagem

    - Se o AVP não existir, nenhuma alteração é feita
- `:remove` - Remover AVPs da mensagem
- `:overwrite` - Substituir estruturas inteiras de AVP
    - Requer o parâmetro `dictionary` especificando o dicionário Diameter (por exemplo, `:diameter_gen_3gpp_s6a`)

**Pacotes de Resposta (Respostas Diameter)**

- `:remove` - Remover AVPs da mensagem
- `:overwrite` - Substituir estruturas inteiras de AVP
    - Requer o parâmetro `dictionary`

**Importante**: Se nenhuma regra corresponder, o pacote é passado através de forma transparente sem transformações.

# Sintaxe de Modificação de AVP

**Modificação padrão:**

- `{:avp, {<código>, <novo_valor>}}` - Definir AVP para novo valor

**Removendo AVPs:**

- `{:avp, {<código>, :any}}` - Remover AVP pelo ID (remove independentemente do valor atual)
- Nota: Remover com base no avp_id é suportado; remover com base no conteúdo do AVP não é suportado

**Sobrescrever com dicionário:**

```
transform: %{
  action: :overwrite,
  dictionary: :diameter_gen_3gpp_s6a,
  avps: [{:avp, {:"s6a_Supported-Features", {:"s6a_Supported-
Features", 10415, 1, 3221225470, []}}}]
}
```

# Exemplos de Transformação

### Exemplo 1: Reescrita de Realm de Destino Baseada em Par

Reescreva o Destination-Realm com base em onde a mensagem está sendo roteada:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: rewrite_s6a_destination_realm_for_Operator_X
      match: ":all"
      filters:
        - '{:to_peer, ["dra01.omnitouch.com.au",
"dra02.omnitouch.com.au"]}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
        - '{:avp, {1, [~r"9999999.*"]}}'
      transform:
        action: ":edit"
        avps:
          - '{:avp, {283, "epc.mnc999.mcc999.3gppnetwork.org"}}'
```

**Como funciona:** Quando requisições S6a são roteadas para pares DRA específicos e correspondem ao padrão IMSI, reescreve o Destination-Realm para a rede do Operador X.

### Exemplo 2: Roteamento de Múltiplos Carriers com Transformações

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name:
rewrite_s6a_destination_realm_for_roaming_partner_ausie
      match: ":all"
      filters:
        - '{:to_peer, ["dra01.omnitouch.com.au",
"dra02.omnitouch.com.au"]}'
        - '{:avp, {296, "epc.mnc057.mcc505.3gppnetwork.org"}}'
        - '{:avp, {1, [~r"50557.*"]}}'
      transform:
        action: ":edit"
        avps:
          - '{:avp, {283, "epc.mnc030.mcc310.3gppnetwork.org"}}'
```

**Como funciona:** Roteia diferentes intervalos de assinantes IMSI para os realms de rede apropriados com base em padrões de IMSI. A primeira regra correspondente vence (veja Ordem de Execução).

**Exemplo 3: Reescrita de Realm de MVNO**

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: rewrite_s6a_destination_realm_for_single_sub
      match: ":all"
      filters:
        - '{:to_peer, ["dra01.omnitouch.com.au",
"dra02.omnitouch.com.au"]}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
        - '{:avp, {1, ["505057000003606"]}}'  # Correspondência
exata de IMSI
      transform:
        action: ":edit"
        avps:
          - '{:avp, {283, "epc.mnc001.mcc001.3gppnetwork.org"}}'
```

**Como funciona:** Transforma o Destination-Realm para um assinante específico de MVNO para sua rede central hospedada.

### Exemplo 4: Transformação Apenas de Requisição com Filtro de Tipo de Pacote

Transforme apenas pacotes de requisição (não respostas):

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: Tutorial_Rule_AIR
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
        - '{:command_code, 318}'
        - '{:packet_type, :request}'
        - '{:avp, {1, "999999000000001"}}'
        - '{:avp, {264, :any}}'  # Origin-Host deve existir com
qualquer valor
      transform:
        action: ":edit"
        avps:
          - '{:avp, {1, "999999000000002"}}'
```

**Como funciona:**

- Corresponde apenas pacotes de requisição S6a **request** (não pacotes de resposta)
- Verifica se o User-Name (AVP 1) é igual a "999999000000001"
- Verifica se o Origin-Host (AVP 264) existe com qualquer valor
- Reescreve o User-Name para "999999000000002"
- Se o AVP não existir, nenhuma alteração é feita

### Exemplo 5: Remover AVP

Remova um AVP específico das mensagens:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: remove_user_name_avp
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
      transform:
        action: ":remove"
        avps:
          - '{:avp, {1, :any}}'  # Remove User-Name
independentemente do valor
```

**Como funciona:** Remove o AVP User-Name (código 1) de todas as mensagens S6a, independentemente de seu valor atual.

#### Exemplo 6: Sobrescrever AVP Agrupado em Pacotes de Resposta

Modifique AVPs agrupados complexos em pacotes de resposta usando a ação `:overwrite` com suporte a dicionário:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: add_sos_apn_to_ula
      match: ":all"
      filters:
        - '{:application_id, 16777251}'       # S6a/S6d
        - '{:command_code, 316}'              # ULA (Update
Location Answer)
        - '{:packet_type, :answer}'           # Apenas pacotes de
resposta
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'  #
Origin-Realm
      transform:
        action: ":overwrite"
        dictionary: ":diameter_gen_3gpp_s6a"
        avps:
          - '{:avp, {:"s6a_APN-Configuration-Profile",
              {:"s6a_APN-Configuration-Profile", 1, 0, [
                {:"s6a_APN-Configuration", 1, 0, "internet", [],
                  [{:"s6a_EPS-Subscribed-QoS-Profile", 9,
                    {:"s6a_Allocation-Retention-Priority", 1, [0],
[0], []}, []}],
                  [1], [], [], [1], ["0800"],
                  [{:s6a_AMBR, 4200000000, 4200000000, [], [],
[]}],
                  [], [], [], [], [], [], [], [], [], [], [], [],
[], [], []},
                {:"s6a_APN-Configuration", 2, 0, "ims", [],
                  [{:"s6a_EPS-Subscribed-QoS-Profile", 5,
                    {:"s6a_Allocation-Retention-Priority", 1, [0],
[1], []}, []}],
                  [0], [], [], [1], ["0800"],
                  [{:s6a_AMBR, 4200000000, 4200000000, [], [],
[]}],
                  [], [], [], [], [], [], [], [], [], [], [], [],
[], [], []},
                {:"s6a_APN-Configuration", 3, 0, "sos", [],
                  [{:"s6a_EPS-Subscribed-QoS-Profile", 5,
                    {:"s6a_Allocation-Retention-Priority", 1, [0],
[1], []}, []}],
                  [1], [], [], [1], ["0800"],
                  [{:s6a_AMBR, 4200000000, 4200000000, [], [],
```

```
[]}],
                  [], [], [], [], [], [], [], [], [], [], [], [],
[], [], []}
               ], []}
            }}'
```

**Como funciona:**

- Corresponde a pacotes de resposta S6a Update Location Answer (ULA) de um Realm de Origem específico
- Usa a ação `:overwrite` para substituir todo o AVP agrupado APN-Configuration-Profile
- **Requer o parâmetro `dictionary`** para codificar corretamente estruturas de AVP agrupadas complexas
- Adiciona três configurações de APN: "internet" (contexto 1), "ims" (contexto 2) e "sos" (contexto 3)
- Cada APN inclui perfis de QoS, limites de largura de banda (AMBR) e configurações de tipo PDN
- A transformação garante que o serviço de emergência (SOS) APN seja provisionado para todos os assinantes deste realm

**Quando usar `:overwrite` com dicionário:**

- Modificando AVPs agrupados com estruturas aninhadas (como APN-Configuration-Profile)
- Adicionando ou reestruturando dados de assinatura complexos 3GPP
- Quando a ação `:edit` não pode lidar com a complexidade do AVP
- O dicionário deve corresponder à aplicação Diameter (`:diameter_gen_3gpp_s6a` para S6a, etc.)

**Notas importantes:**

- `:overwrite` substitui todo o AVP, não apenas campos individuais
- A estrutura do AVP deve corresponder exatamente à definição do dicionário
- Estruturas incorretas causarão falhas de codificação e pacotes descartados
- Este é um recurso avançado - valide completamente em ambiente de teste primeiro

## Casos de Uso

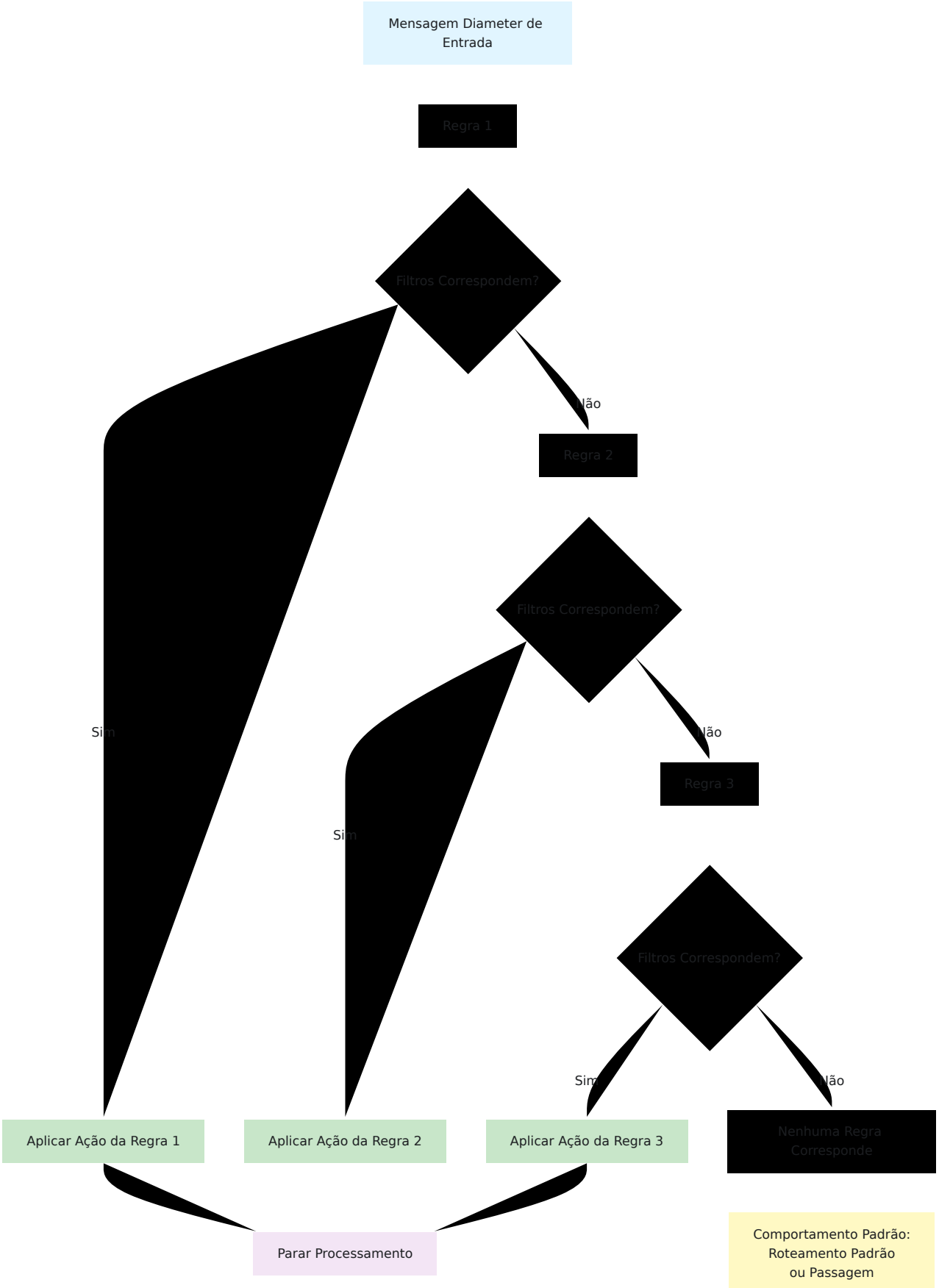
- **Suporte a MVNO**: Roteie tráfego de operadores virtuais para redes centrais hospedadas
- **Migração de Rede**: Redirecione gradualmente assinantes para nova infraestrutura
- **Tradução de Realm**: Converta entre diferentes esquemas de nomenclatura para parceiros de roaming
- **Multi-tenancy**: Isolar populações de assinantes por realm
- **Roteamento de Carrier**: Direcione tráfego para redes de carriers corretas com base em intervalos de IMSI

---

# Processamento de Regras

Aplica-se tanto aos módulos Roteamento Avançado quanto à Transformação Avançada.

# Ordem de Execução



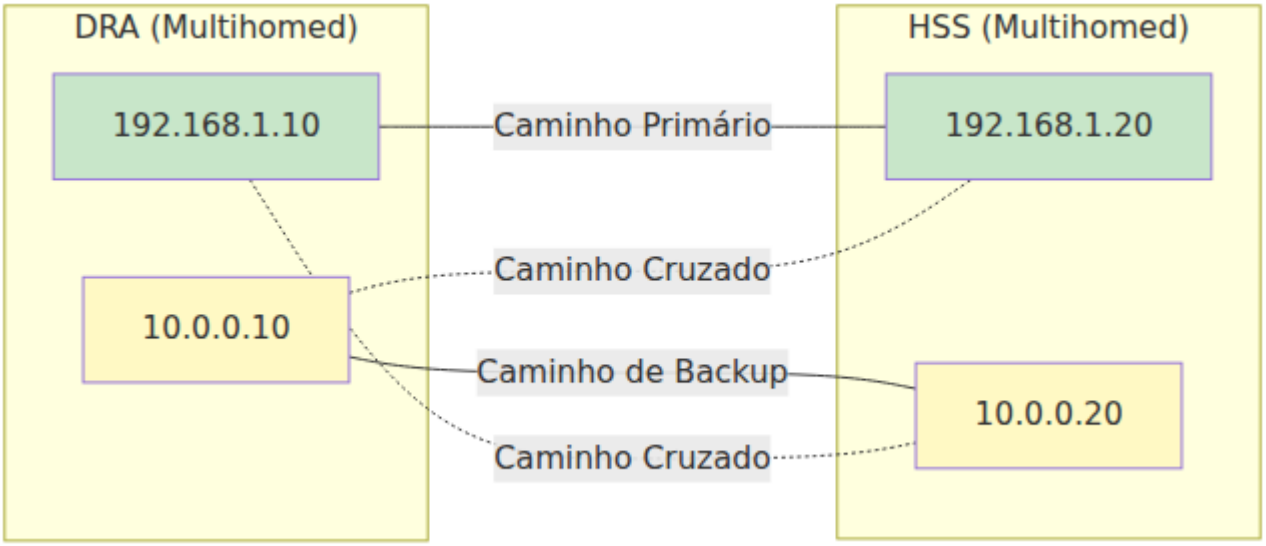
1. As regras são avaliadas **na ordem de cima para baixo** conforme definido na configuração
2. Os filtros dentro de uma regra são avaliados com base no parâmetro `match` (`:all`, `:any` ou `:none`)
3. **A primeira regra correspondente vence** - regras subsequentes não são avaliadas
4. Se nenhuma regra corresponder, o comportamento padrão de roteamento/passthrough é usado

# Lógica de Filtro

O parâmetro `match` determina como os filtros são combinados:

**match: :all (Lógica AND)**

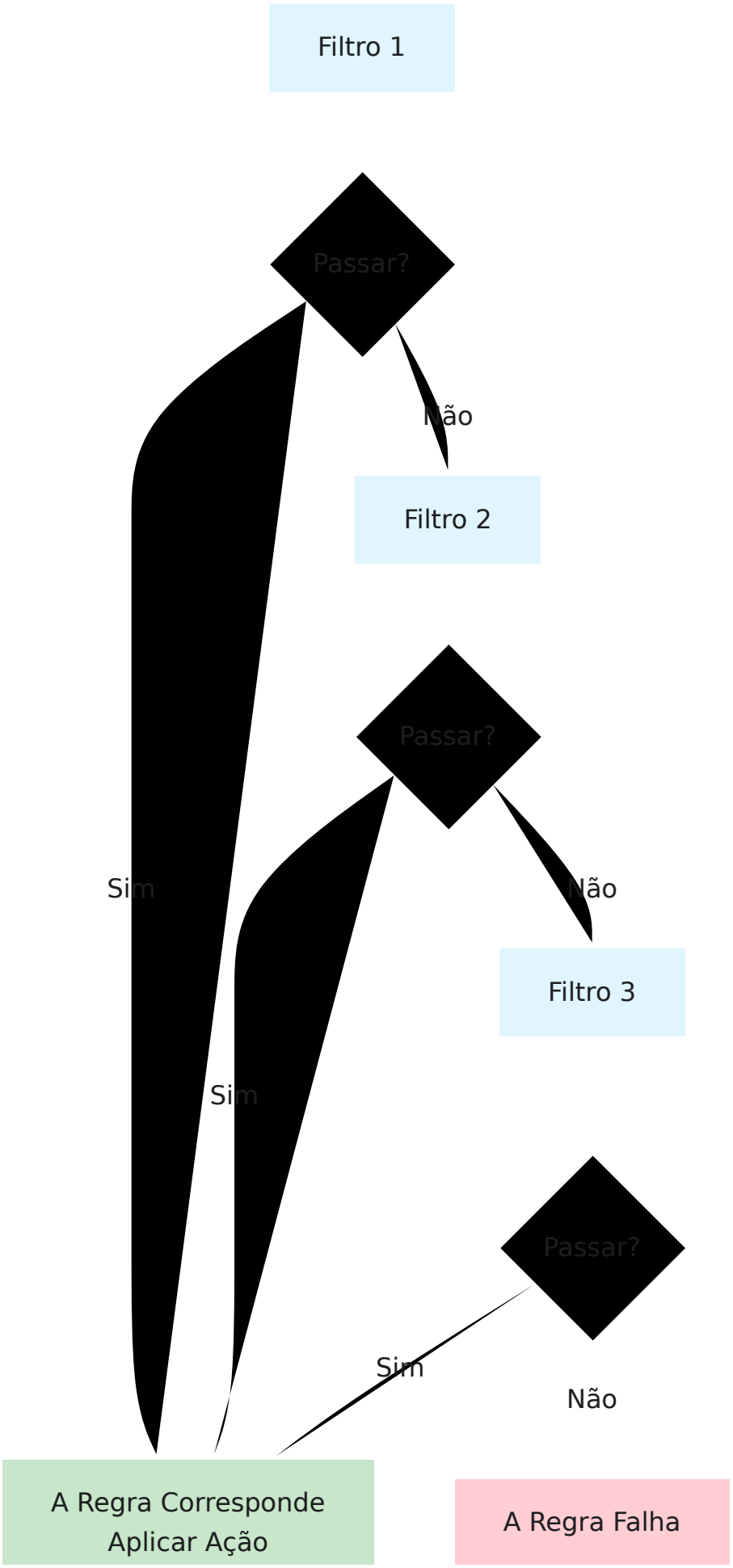
Todos os filtros devem corresponder para que a regra seja bem-sucedida.




Exemplo: Com 3 filtros, `filtro1 AND filtro2 AND filtro3` devem ser todos verdadeiros.

**match: :any (Lógica OR)**

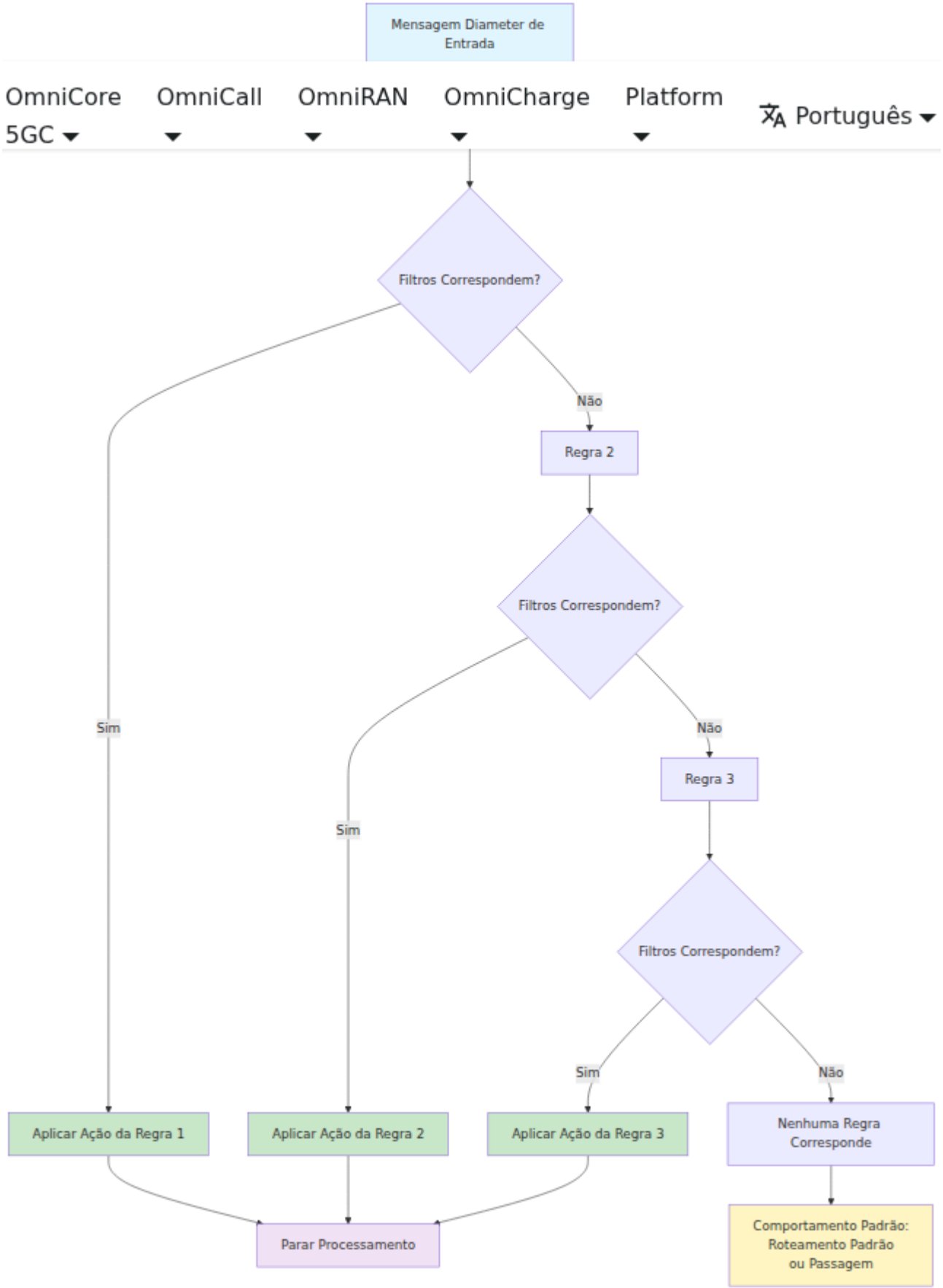
Pelo menos um filtro deve corresponder para que a regra seja bem-sucedida.


Filtro 1
Passar?
Não
Filtro 2
Passar?
Sim
Não
Filtro 3
Sim
Passar?
Sim
Não
A Regra Corresponde
Aplicar Ação
A Regra Falha

Exemplo: Com 3 filtros, `filtro1 OR filtro2 OR filtro3` (qualquer um passa).

**match: :none (Lógica NOR)**

Nenhum filtro pode corresponder para que a regra seja bem-sucedida (correspondência inversa).




Exemplo: Com 3 filtros, `NÃO filtro1 AND NÃO filtro2 AND NÃO filtro3` (todos devem falhar).

---

**Notas Adicionais:**

Ao usar operadores de lista dentro de um valor de filtro (por exemplo, `{:avp, {1, ["value1", "value2"]}}`), os valores usam lógica **OR** (qualquer pode corresponder).

## Padrões de Expressão Regular

Use a sintaxe `~r"padrão"` para correspondência regex:

- `~r"999001.*"` - Corresponde a IMSI começando com 999001
- `~r"^310[0-9]{3}.*"` - Corresponde a IMSI com padrões MNC específicos
- `~r".*teste$"` - Corresponde a valores terminando com "teste"

## Melhores Práticas

1. **Especificidade**: Ordene regras da mais específica para a mais geral
2. **Desempenho**: Coloque as correspondências mais comuns primeiro para reduzir a sobrecarga de processamento
3. **Teste**: Valide padrões regex antes da implantação
4. **Documentação**: Use valores descritivos para `rule_name` para clareza operacional
5. **Monitoramento**: Acompanhe taxas de correspondência de regras para verificar comportamento esperado

---

# Módulo de Métricas Estendidas

O módulo de Métricas Estendidas fornece telemetria e capacidades analíticas avançadas para analisar padrões de tráfego Diameter além das métricas padrão.

## Configuração

Habilite o módulo e configure tipos específicos de métricas:

```
module_extended_metrics:
  enabled: true
  attach_attempt_reporting_enabled: true
```

## Parâmetros

| Parâmetro | Descrição |
|---|---|
| `enabled` | Defina como `true` para ativar o módulo de métricas estendidas |
| `attach_attempt_reporting_enabled` | Habilitar rastreamento e relatório de tentativas de anexação LTE (S6a AIR/AIA) |

## Métricas Disponíveis

### Rastreamento de Tentativas de Anexação

Rastreia tentativas de anexação de assinantes LTE monitorando pares de mensagens Authentication Information Request (AIR) e Answer (AIA):

Parse error on line 36: ... style Metrics fill:#f3e5f5 style E ----------------------^ Expecting 'SOLID_OPEN_ARROW', 'DOTTED_OPEN_ARROW', 'SOLID_ARROW', 'BIDIRECTIONAL_SOLID_ARROW', 'DOTTED_ARROW', 'BIDIRECTIONAL_DOTTED_ARROW', 'SOLID_CROSS', 'DOTTED_CROSS', 'SOLID_POINT', 'DOTTED_POINT', got 'TXT'

Tentar novamente

**Medição**: `attach_attempt_count`

**Campos**:

- `imsi` - O IMSI do assinante (do AVP User-Name - veja códigos AVP)

**Tags**:

- `origin_host` - O par que originou a requisição de anexação
- `result_code` - O código de resultado Diameter da resposta HSS

**Como funciona**:

1. Quando uma AIR (código de comando 318, aplicação S6a 16777251 - veja IDs de Aplicação) é recebida, o módulo extrai:
   - End-to-End-ID para correlação de requisição/resposta
   - IMSI (AVP User-Name código 1)
   - Origin-Host (AVP código 264)
2. Os metadados da requisição são armazenados no ETS com TTL
3. Quando a AIA correspondente é recebida, o módulo:
   - Correlaciona usando End-to-End-ID
   - Extrai o código de resultado (AVP 268 ou AVP de resultado experimental 297)
   - Emite a métrica com IMSI, origin host e código de resultado

# Casos de Uso

- **Análise da Taxa de Sucesso de Anexação** - Rastreie tentativas de anexação bem-sucedidas vs falhadas por código de resultado
- **Solução de Problemas em Nível de IMSI** - Identifique assinantes que estão enfrentando falhas de anexação
- **Monitoramento de Desempenho da Rede** - Monitore padrões de tentativas de anexação por origem (MME/SGSN)
- **Análise de Roaming** - Analise taxas de sucesso de anexação de roaming inbound

# Integração

Métricas estendidas são exportadas via integração InfluxDB:

```
DRA.Metrics.InfluxDB.write(%{
  measurement: "attach_attempt_count",
  fields: %{imsi: "505057000000001"},
  tags: %{origin_host: "mme-01.example.com", result_code: 2001}
})
```

Os códigos de resultado são códigos padrão Diameter:

- `2001` - Sucesso (DIAMETER_SUCCESS)
- `5001` - Falha de autenticação (DIAMETER_AUTHENTICATION_REJECTED)
- `5004` - AVP Diameter não suportado
- Veja a RFC 6733 para a lista completa de códigos de resultado

## Notas Importantes

- Métricas de tentativas de anexação rastreiam apenas pares AIR/AIA S6a (Application-Id 16777251, Command-Code 318)
- Metadados de requisição expiram com base no timeout de requisição configurado + 5 segundos
- O processamento de métricas é assíncrono (processo gerado) para evitar bloquear o fluxo de mensagens
- O módulo opera de forma independente dos módulos de roteamento e transformação

---

# Módulos de Segurança e Direcionamento

O DRA é fornecido com três módulos autônomos, configuráveis de forma independente para segurança de interconexão e controle de roaming. Cada um possui seu próprio guia de operações dedicado:

- **Segurança Diameter** — Proteção alinhada ao GSMA FS.19/FS.21: o Firewall Diameter (verificações de formato de camada baixa mais filtragem

de Categoria 1/2/3), Ocultação de Topologia (remoção de Route-Record e reescrita de Origin), Limitação de Taxa por par e Sanitização de AVP.
- **Direcionamento de Roaming (SoR)** — Direciona assinantes de roaming para VPLMNs preferidos, rejeitando Update-Location-Requests que se originam de redes não preferidas.
- **Consulta de Assinante (SLF)** — Aprende dinamicamente associações de assinante a nó de serviço a partir do tráfego Diameter e roteia requisições SLg/SLh para o nó de serviço correto.

---

# Métricas Prometheus

O DRA expõe métricas abrangentes do Prometheus para monitorar tráfego Diameter, saúde de pares e operações de módulos. Todas as métricas estão disponíveis no endpoint `/metrics`.

## Métricas de Diameter Core

### Status do Par

**Métrica**: `diameter_peer_status` **Tipo**: Gauge **Descrição**: Se o par está conectado (1) ou não (0) **Tags**:

- `origin_host` - Identidade Diameter do par
- `ip` - Endereço IP do par

**Exemplo**:

```
# Verifique se um par específico está conectado
diameter_peer_status{origin_host="hss01.example.com"}

# Contar pares desconectados
count(diameter_peer_status == 0)
```

### Contagem de Mensagens

**Métrica**: `diameter_peer_message_count_total` **Tipo**: Counter **Descrição**: Total de mensagens Diameter trocadas com pares **Tags**:

- `origin_host` - Identidade Diameter do par
- `received_from` - Par de onde a mensagem foi recebida
- `application_id` - Diameter Application-Id (veja referência de ID de Aplicação)
- `cmd_code` - Diameter Command-Code (veja Códigos de Comando Comuns)
- `application_name` - Nome da aplicação legível por humanos (por exemplo, "3GPP_S6a")
- `cmd_name` - Nome do comando legível por humanos (por exemplo, "AIR")
- `direction` - "request" ou "response"

**Exemplo**:

```
# Taxa de requisições S6a AIR de um MME específico
rate(diameter_peer_message_count_total{
  cmd_code="318",
  direction="request",
  origin_host="mme01.example.com"
}[5m])

# Taxa total de mensagens por aplicação
sum by (application_name)
(rate(diameter_peer_message_count_total[5m]))
```

**Códigos de Resultado de Resposta**

**Métrica**: `diameter_peer_message_result_code_count_total` **Tipo**: Counter **Descrição**: Total de respostas Diameter por código de resultado **Tags**:

- `origin_host` - Solicitante original
- `routed_to` - Par que enviou a resposta
- `application_id` - Diameter Application-Id
- `cmd_code` - Diameter Command-Code
- `application_name` - Nome da aplicação
- `cmd_name` - Nome do comando

- `result_code` - Código de Resultado Diameter ou Código de Resultado Experimental

**Exemplo**:

```
# Taxa de sucesso para requisições S6a AIR
rate(diameter_peer_message_result_code_count_total{
  cmd_code="318",
  result_code="2001"
}[5m])

# Taxa de erro por código de resultado
sum by (result_code) (
  rate(diameter_peer_message_result_code_count_total{
    result_code!="2001"
  }[5m])
)
```

**Códigos de Resultado Comuns**:

- `2001` - DIAMETER_SUCCESS
- `3002` - DIAMETER_UNABLE_TO_DELIVER
- `3003` - DIAMETER_REALM_NOT_SERVED
- `3004` - DIAMETER_TOO_BUSY
- `5001` - DIAMETER_AUTHENTICATION_REJECTED
- `5004` - DIAMETER_INVALID_AVP_VALUE
- `5012` - DIAMETER_UNABLE_TO_COMPLY

**Atraso de Resposta**

**Métrica**: `diameter_peer_last_response_delay` **Tipo**: Gauge **Descrição**: Atraso mais recente de resposta em milissegundos (DRA → Par → DRA) **Tags**:

- `origin_host` - Solicitante original
- `routed_to` - Par que enviou a resposta
-

# Direcionamento de Roaming (SoR)

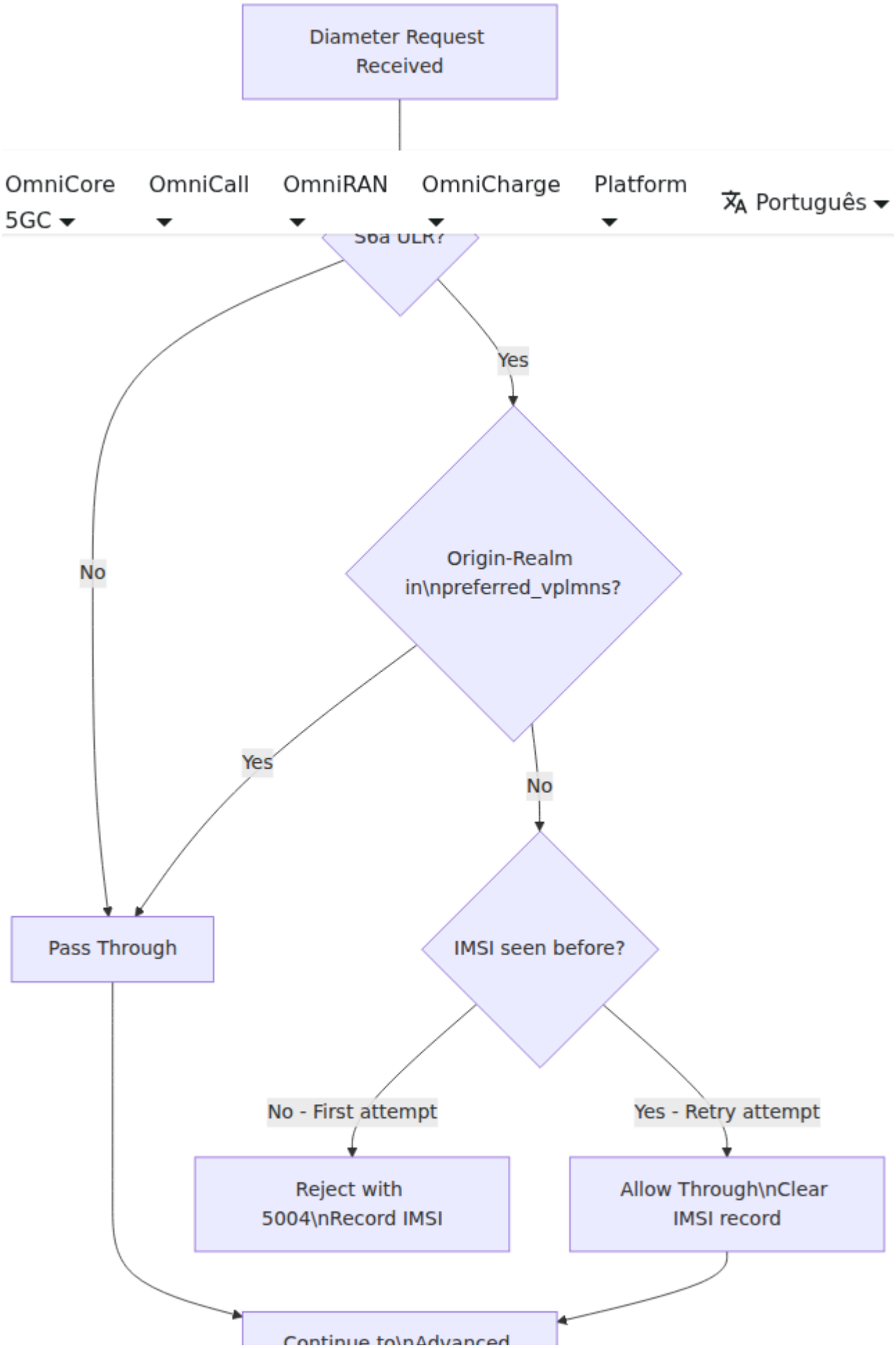
O Direcionamento de Roaming (SoR) permite que o HPLMN influencie a qual VPLMN um assinante em roaming se conecta. Quando um assinante tenta se registrar via um VPLMN não preferido, o DRA intercepta a S6a Update-Location-Request (ULR) e a rejeita com **DIAMETER_ERROR_ROAMING_NOT_ALLOWED (5004)**. Isso faz com que o UE se desconecte e se reconecte, idealmente selecionando um VPLMN preferido.

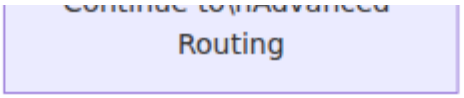
Se a cobertura preferida não estiver disponível e o assinante retornar ao mesmo VPLMN não preferido, o módulo permite o pedido na tentativa subsequente.

O SoR é definido na 3GPP TS 29.272 Seção 5.2.1.1 e na 3GPP TS 23.122.

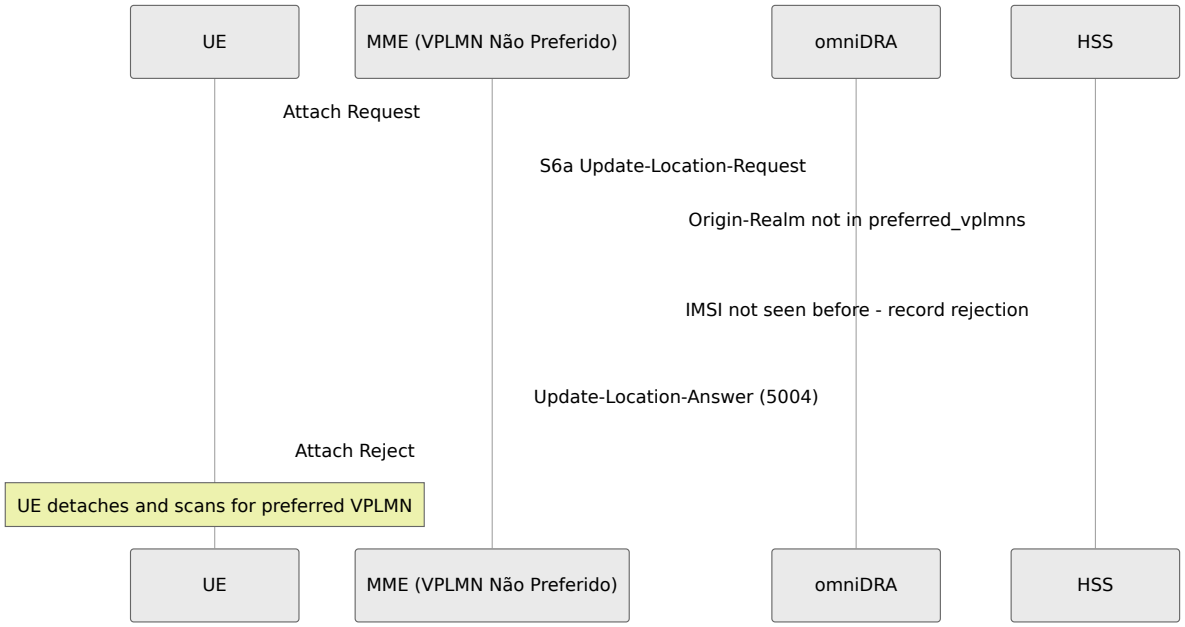
## Como Funciona

O módulo opera dentro do pipeline de processamento de pedidos do DRA, executando **antes** do Roteamento Avançado. Apenas S6a Update-Location-Requests (Application-Id `16777251`, Command-Code `316`) são avaliados. Todas as outras mensagens Diameter passam sem modificações.

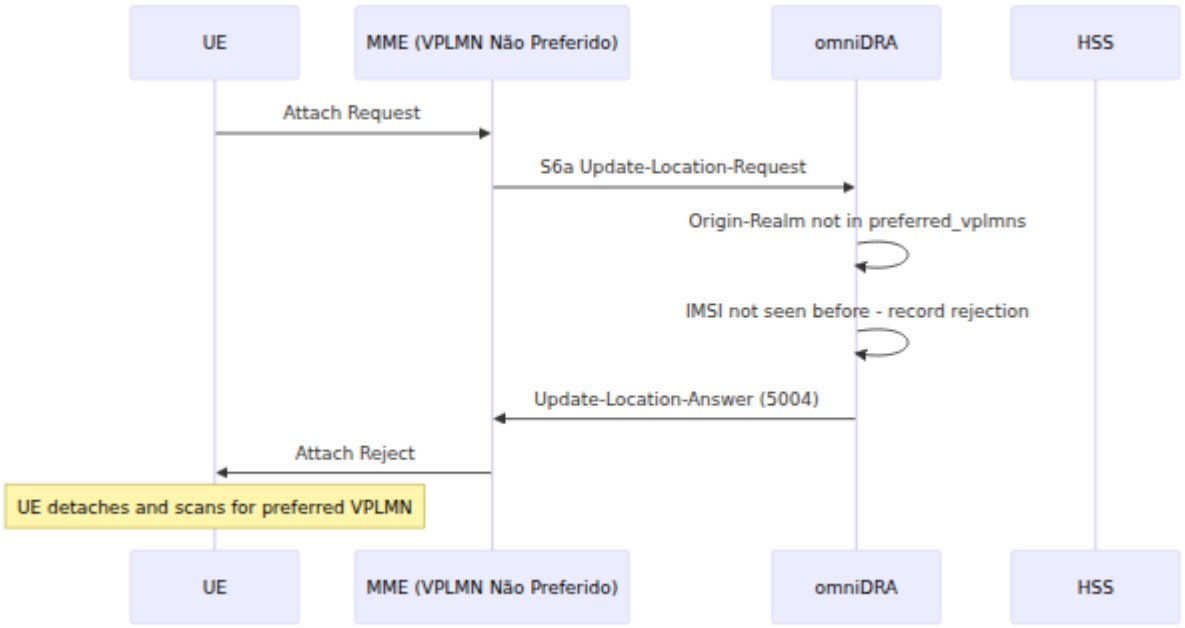

Diameter Request Received
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
S6a ULR?
Yes
No
Origin-Realm in\npreferred_vplmns?
Yes
No
Pass Through
IMSI seen before?
No - First attempt
Yes - Retry attempt
Reject with 5004\nRecord IMSI
Allow Through\nClear IMSI record
Continue to\nAdvanced




# Sequência: Primeira Tentativa (Rejeitada)




# Sequência: Segunda Tentativa (Permitida)



# Posição no Pipeline

O Direcionamento de Roaming é executado **antes** do Roteamento Avançado no pipeline de processamento de pedidos. Se o SoR rejeitar um pedido, o Roteamento Avançado nunca será alcançado para essa mensagem.


# Rastreamento de Assinantes

O módulo rastreia cada assinante rejeitado por IMSI. Cada registro inclui uma contagem de rejeições e um timestamp. Os registros são automaticamente limpos após a expiração do TTL configurado.

Quando um IMSI rastreado atinge o limite de `max_rejections` e envia outra ULR, o pedido é permitido e o registro de rastreamento é excluído.

Se nenhuma ULR adicional chegar dentro da janela TTL, o registro expira e é removido durante o próximo ciclo de limpeza. Uma ULR subsequente do mesmo IMSI será tratada como uma primeira tentativa e rejeitada novamente.

# Configuração

O módulo é configurado sob `module_roaming_steering` em `config/runtime.exs`.

```
module_roaming_steering: %{
  # Enable or disable the module
  enabled: false,

  # Number of rejections before allowing through
  max_rejections: 1,

  # Time in seconds before a tracked IMSI record expires
  rejection_ttl_seconds: 300,

  # Diameter Result-Code returned in the rejection answer
  rejection_result_code: 5004,

  # List of Origin-Realm values considered preferred VPLMNs
  # ULRs from these realms are always allowed through
  preferred_vplmns: [
    "epc.mnc001.mcc001.3gppnetwork.org",
    "epc.mnc002.mcc001.3gppnetwork.org"
  ]
}
```

## Parâmetros

| Parâmetro | Tipo | Obrigatório | Padrão | |
|---|---|---|---|---|
| `enabled` | Booleano | Sim | `false` | Habilitar ou Direcionam `false`, todo modificaçõe |
| `max_rejections` | Inteiro | Não | `1` | Número de antes de pe padrão de segunda é |
| `rejection_ttl_seconds` | Inteiro | Não | `300` | Tempo em s de IMSI rast assinante n dessa janel tratada com Também co |
| `rejection_result_code` | Inteiro | Não | `5004` | Código de R resposta de DIAMETER_ conforme 3 |
| `preferred_vplmns` | Lista | Não | `[]` | Lista de stri VPLMNs pre pares com u são sempre |

## Exemplos de Configuração

### Rejeitar Uma Vez, Depois Permitir

O comportamento padrão. Assinantes que atingem um VPLMN não preferido são rejeitados uma vez. Se retornarem, a ULR é encaminhada para o HSS.

```
module_roaming_steering: %{
  enabled: true,
  max_rejections: 1,
  rejection_ttl_seconds: 300,
  rejection_result_code: 5004,
  preferred_vplmns: [
    "epc.mnc001.mcc310.3gppnetwork.org",
    "epc.mnc002.mcc310.3gppnetwork.org"
  ]
}
```

**Como funciona:** Uma ULR chegando de um MME em `epc.mnc003.mcc310.3gppnetwork.org` (não na lista preferida) é rejeitada com o Código de Resultado 5004. O UE se desconecta e tenta encontrar uma rede preferida. Se retornar ao mesmo VPLMN não preferido dentro de 300 segundos, a ULR é encaminhada normalmente.

**Caso de uso:** Direcionamento de roaming padrão onde o operador tem acordos de roaming preferenciais com redes parceiras específicas.

#### Múltiplas Rejeições Antes de Permitir

Para cenários onde o operador deseja que o UE faça várias tentativas antes de voltar a um VPLMN não preferido.

```
module_roaming_steering: %{
  enabled: true,
  max_rejections: 3,
  rejection_ttl_seconds: 600,
  rejection_result_code: 5004,
  preferred_vplmns: [
    "epc.mnc001.mcc310.3gppnetwork.org"
  ]
}
```

**Como funciona:** O assinante é rejeitado três vezes antes de ser permitido na quarta tentativa. O TTL é estendido para 600 segundos para acomodar os ciclos adicionais de tentativa.

**Caso de uso:** Áreas urbanas densas onde a cobertura preferida existe, mas pode exigir várias tentativas de conexão para ser adquirida.

# Métricas

## Rejeições

**Métrica**: `diameter.roaming_steering.reject.count` **Tipo**: Contador
**Descrição**: Incrementado cada vez que uma ULR é rejeitada pelo módulo SoR.
**Rótulos**:

- `origin_realm` - Origin-Realm do VPLMN não preferido
- `result_code` - Código de Resultado Diameter retornado na rejeição
- `imsi` - IMSI do assinante rejeitado

## Permitido (VPLMN Preferido)

**Métrica**: `diameter.roaming_steering.allow.count` **Tipo**: Contador
**Descrição**: Incrementado quando uma ULR é permitida. **Rótulos**:

- `origin_realm` - Origin-Realm do VPLMN de origem
- `reason` - Por que o pedido foi permitido: `preferred_vplmn` ou `max_rejections_reached`
- `imsi` - IMSI do assinante (presente apenas quando a razão é `max_rejections_reached`)

## Respostas de Erro

**Métrica**: `diameter.roaming_steering.error.count` **Tipo**: Contador
**Descrição**: Incrementado quando o processador retorna uma mensagem de resposta de erro devido a uma rejeição do SoR. **Rótulos**:

- `result_code` - Código de Resultado Diameter
- `application_id` - Application-Id Diameter (numérico)
- `cmd_code` - Command-Code Diameter (numérico)
- `application_name` - Nome da aplicação legível por humanos

- `cmd_name` - Nome do comando legível por humanos

# Solução de Problemas

## Assinantes Sempre Rejeitados (Nunca Permitidos)

**Sintomas**: Assinantes em VPLMNs não preferidos são repetidamente rejeitados e nunca se conectam com sucesso.

**Causas possíveis**:

- `rejection_ttl_seconds` é muito curto, fazendo com que o registro de rastreamento expire antes que o assinante tente novamente
- `max_rejections` está definido para um valor maior do que o número de tentativas que o UE realiza antes de desistir

**Resolução**:

1. Aumentar `rejection_ttl_seconds` para acomodar o tempo de tentativa do UE
2. Verificar se `max_rejections` está definido para um valor que o UE pode realisticamente alcançar dentro de seu ciclo de tentativas

## Assinantes em VPLMNs Preferidos Sendo Rejeitados

**Sintomas**: ULRs de redes parceiras preferidas estão sendo rejeitadas com 5004.

**Causas possíveis**:

- O Origin-Realm do VPLMN preferido não está listado em `preferred_vplmns`
- A string Origin-Realm não corresponde exatamente (sensível a maiúsculas e minúsculas)

**Resolução**:

1. Verificar o Origin-Realm do par nos logs do DRA no momento da conexão
2. Garantir que a string exata do Origin-Realm esteja incluída na lista `preferred_vplmns`

## Módulo Não Tendo Efeito

**Sintomas**: ULRs de VPLMNs não preferidos estão sendo encaminhadas sem rejeição.

**Causas possíveis**:

- `enabled` está definido como `false`
- O processo do módulo não está em execução (verificar supervisor)

**Resolução**:

1. Verificar `enabled: true` na configuração
2. Confirmar que o DRA foi reiniciado após a alteração da configuração

# Referência

## Códigos Diameter

| Código | Nome | Descrição | Refer |
|---|---|---|---|
| 5004 | DIAMETER_ERROR_ROAMING_NOT_ALLOWED | Assinante não permitido a roaming neste VPLMN | 3GPP 29.27 Seçã |

## Comandos S6a

| Command-Code | Nome | Descrição | Referência |
|---|---|---|---|
| 316 | Update-Location-Request/Answer (ULR/ULA) | Enviado pelo MME para o HSS durante a conexão para atualizar o nó de serviço do assinante | 3GPP TS 29.272 Seção 7.2.3 |

## IDs de Aplicação

| ID | Interface | Descrição | Referência |
|---|---|---|---|
| 16777251 | S6a/S6d | Autenticação e gerenciamento de assinatura MME/SGSN para HSS | 3GPP TS 29.272 |

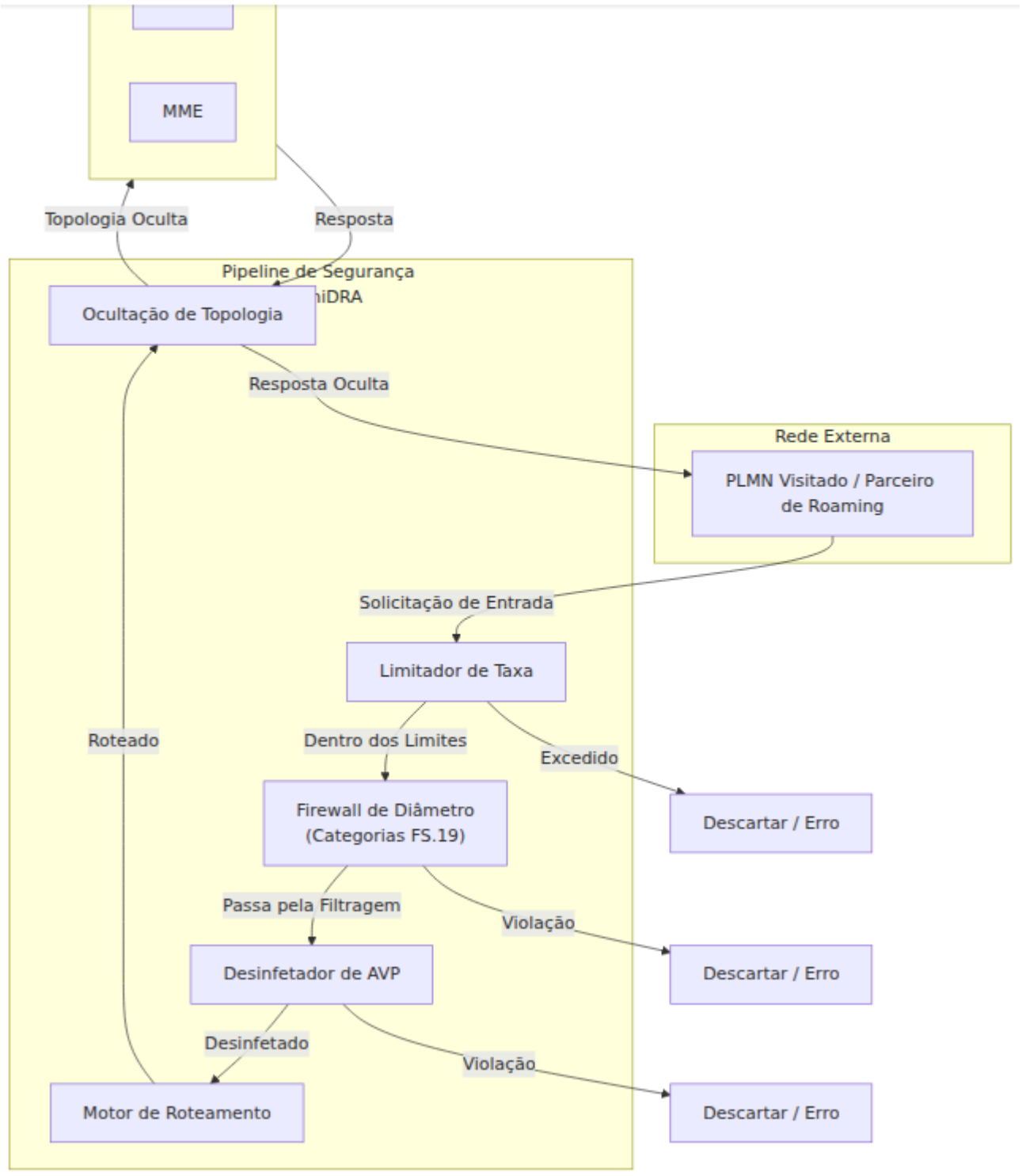
# Segurança do Diâmetro

OmniDRA fornece um conjunto abrangente de módulos de segurança do Diâmetro alinhados com **GSMA FS.19** (Segurança de Interconexão do Diâmetro, v10.0) e **GSMA FS.21** (Recomendações de Segurança de Sinalização de Interconexão, v12.0). Esses módulos protegem a rede contra ataques em nível de sinalização nas interfaces de interconexão do Diâmetro.

Cada módulo é configurável de forma independente, pode ser habilitado ou desabilitado sem afetar os outros e emite seus próprios eventos de telemetria para monitoramento e alerta.

# Visão Geral da Arquitetura





# Ordem de Processamento

As solicitações de Diâmetro de entrada passam pelos módulos de segurança na seguinte ordem. Uma mensagem rejeitada em qualquer estágio nunca é encaminhada para estágios subsequentes.

| Ordem | Módulo | Direção | Propósito |
|---|---|---|---|
| 1 | Limitador de Taxa | Entrada | Proteção contra inundação volumétrica |
| 2 | Firewall de Diâmetro | Entrada | Filtragem de protocolo e conteúdo FS.19 |
| 3 | Desinfetador de AVP | Entrada | Validação e remoção de AVP |
| 4 | Motor de Roteamento | - | Roteamento padrão de Diâmetro |
| 5 | Ocultação de Topologia | Saída | Ocultação da topologia da rede |

# Ações

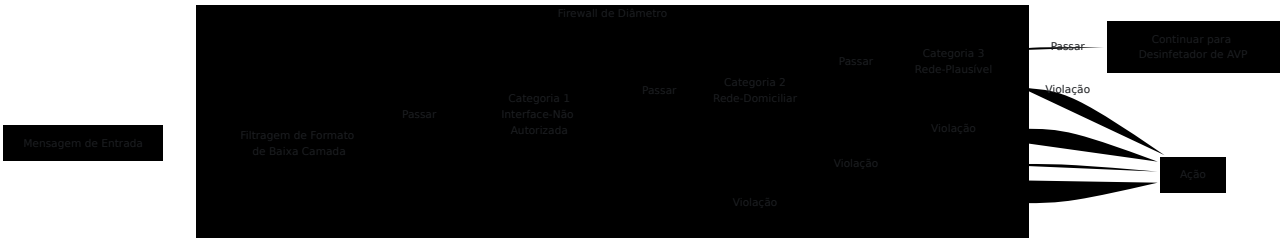
Todos os módulos de segurança suportam ações configuráveis quando uma violação é detectada. As ações são configuradas por módulo (e por categoria para o Firewall de Diâmetro).

| Aç��o | Comportamento | Resultado do Diâmetro |
|---|---|---|
| `{:error, 3002}` | Responder com uma resposta de erro do Diâmetro | DIAMETER_UNABLE_TO_DELIVER |
| `{:error, 3004}` | Responder com uma resposta de erro do Diâmetro | DIAMETER_TOO_BUSY |
| `{:error, 3007}` | Responder com uma resposta de erro do Diâmetro | DIAMETER_APPLICATION_UNSUPPORTED |
| `:drop` | Descartar a mensagem silenciosamente | Nenhuma resposta enviada |
| `:log_only` | Registrar a violação, mas permitir a passagem da mensagem | Mensagem continua |

---

# Firewall de Diâmetro

O Firewall de Diâmetro implementa as quatro camadas de filtragem definidas na **Seção 3.3 do GSMA FS.19** e a categorização de pacotes independente de protocolo da **Seção 7 do FS.21**. As mensagens são avaliadas através de cada camada na ordem; uma violação em qualquer camada interrompe o processamento adicional.




# Filtragem de Formato de Baixa Camada

**Referência GSMA:** Seção 3.3.4 do FS.19, Seção 7.3.1 do FS.21

A filtragem de baixa camada detecta violações em nível de protocolo e tentativas básicas de anti-spoofing sem precisar entender a semântica da aplicação de camadas superiores. Esta camada captura mensagens malformadas antes que elas cheguem a uma inspeção mais profunda.

**Verificações realizadas:**

| Verificação | Referência FS.19 | Descrição |
| --- | --- | --- |
| Duplicação de AVP | Seção 3.3.4, 4.8.1 | Detecta instâncias duplicadas de AVPs que devem aparecer no máximo uma vez (por exemplo, Origin-Host, Origin-Realm). Previne ataques de evasão de duplicação de AVP. |
| Ordenação de Session-Id | Seção 3.3.4 | Valida que o Session-Id (AVP 263) é o primeiro AVP na mensagem, conforme RFC 6733 Seção 8.8. |
| Destination-Host em Respostas | Seção 3.3.4 | Rejeita mensagens de resposta que contêm um AVP Destination-Host, que é uma violação de protocolo que pode indicar evasão de filtro. |
| Validação de AVP Obrigatórios | Seção 3.3.5 | Valida que os AVPs obrigatórios estão presentes para cada código de comando (por exemplo, ULR deve conter User-Name e Visited-PLMN-Id conforme 3GPP TS 29.272 Seção 7.2.3). |

```
low_layer: %{
  enabled: true,
  action: {:error, 3002},
  # Códigos de AVP que não devem aparecer mais de uma vez (Seção
3.3.4 do FS.19)
  # 264 = Origin-Host, 296 = Origin-Realm, 283 = Destination-
Realm, 293 = Destination-Host
  single_instance_avps: [264, 296, 283, 293],
  # Session-Id deve ser o primeiro AVP (Seção 8.8 da RFC 6733)
  enforce_session_id_first: true,
  # Mensagens de resposta não devem conter Destination-Host (Seção
3.3.4 do FS.19)
  reject_destination_host_in_answers: true,
  # AVPs obrigatórios por código de comando (3GPP TS 29.272)
  # 1 = User-Name (IMSI), 1407 = Visited-PLMN-Id, 264 = Origin-
Host, 296 = Origin-Realm
  mandatory_avps: %{
    316 => [1, 1407, 264, 296],
    318 => [1, 1407, 264, 296]
  }
}
```

## Categoria 1 — Filtragem de Pacotes Interface-Não Autorizada

**Referência GSMA:** Seção 3.3.5 do FS.19, Seção 7.2.1 do FS.21

A filtragem da Categoria 1 garante que apenas IDs de Aplicação de Diâmetro e Códigos de Comando autorizados sejam aceitos em cada interface de interconexão. Isso previne o acesso externo a interfaces apenas internas (por exemplo, bloqueando comandos Sh sobre uma interface S6a) e impõe que parceiros de roaming enviem apenas tipos de mensagens cobertos por seus acordos de roaming.

A abordagem de lista branca segue a recomendação do FS.19: **bloquear todas as mensagens de Diâmetro, exceto aquelas explicitamente necessárias para uma determinada interface**.

As listas brancas podem ser configuradas globalmente (aplicando-se a todos os pares) ou por par, permitindo que diferentes parceiros de roaming tenham

conjuntos de mensagens permitidas diferentes.

```
category_1: %{
  enabled: true,
  action: {:error, 3007},
  whitelists: %{
    # Lista branca por par — restringir este parceiro a ULR e AIR
apenas (Seção 3.3.5 do FS.19)
    "restricted-partner.roaming.com" => %{
      16_777_251 => [316, 318]
    },
    # Lista branca padrão — comandos S6a padrão permitidos para
todos os outros pares
    all: %{
      # S6a/S6d (Seção 3.3.5 do FS.19, Tabela 2)
      16_777_251 => [316, 317, 318, 319, 320, 321, 323],
      # S13 — Verificação de Identidade do ME (Seção 3.3.5 do
FS.19)
      16_777_252 => [324],
      # S6c — SMS via HSS (Seção 3.3.5.1 do FS.19)
      16_777_312 => [8388647, 8388648],
      # SGd — SMS via MME (Seção 3.3.5.2 do FS.19)
      16_777_313 => [8388645, 8388646]
    }
  }
}
```

**Lista Branca Comum S6a/S6d:**

| Código de Comando | Nome | Direção | Referência |
|---|---|---|---|
| 316 | Update-Location-Request/Answer | MME → HSS | 3GPP TS 29.272 §7.2.3 |
| 317 | Cancel-Location-Request/Answer | HSS → MME | 3GPP TS 29.272 §7.2.7 |
| 318 | Authentication-Information-Request/Answer | MME → HSS | 3GPP TS 29.272 §7.2.5 |
| 319 | Insert-Subscriber-Data-Request/Answer | HSS → MME | 3GPP TS 29.272 §7.2.9 |
| 320 | Delete-Subscriber-Data-Request/Answer | HSS → MME | 3GPP TS 29.272 §7.2.11 |
| 321 | Purge-UE-Request/Answer | MME → HSS | 3GPP TS 29.272 §7.2.13 |
| 323 | Notify-Request/Answer | MME → HSS | 3GPP TS 29.272 §7.2.15 |

**IDs de Aplicação Comuns para interconexão de roaming** (Seção 3.3.5 do FS.19):

| ID de Aplicação | Interface | Referência |
|---|---|---|
| 16777251 | S6a/S6d | 3GPP TS 29.272 |
| 16777252 | S13 | 3GPP TS 29.272 |
| 16777312 | S6c | 3GPP TS 29.338 |
| 16777313 | SGd | 3GPP TS 29.338 |
| 16777255 | SLg | 3GPP TS 29.172 |
| 16777267 | S9 | 3GPP TS 29.215 |

## Categoria 2 — Filtragem de Pacotes da Rede Domiciliar

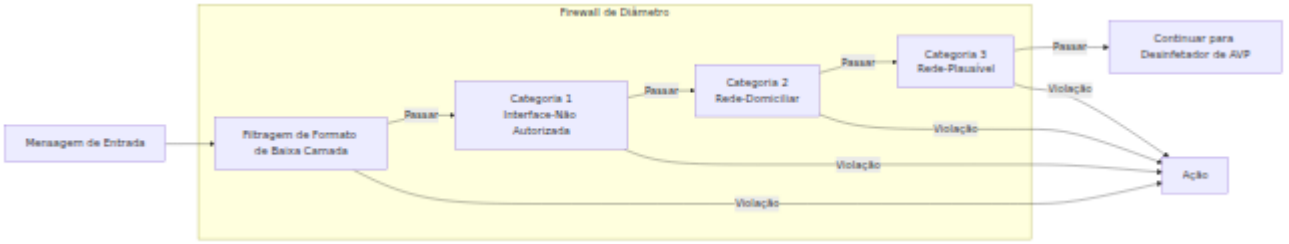
**Referência GSMA:** Seção 3.3.6 do FS.19, Seção 7.2.2 do FS.21

A filtragem da Categoria 2 protege os assinantes da rede domiciliar de serem alvos de mensagens que chegam da interconexão. Mensagens em códigos de comando protegidos são inspecionadas quanto à identidade do assinante (IMSI no AVP User-Name, MSISDN no AVP 701). Se a identidade corresponder a um prefixo de assinante domiciliar, a mensagem é rejeitada — o tráfego legítimo para assinantes domiciliares deve se originar de dentro da rede domiciliar, não de pares externos.

Isso previne ataques onde uma entidade externa envia atualizações de localização, solicitações de dados de assinantes ou consultas de autenticação visando os próprios assinantes do operador.

```
# Nível superior: definir prefixos de assinantes domiciliares
home_imsi_prefixes: ["31338", "31339"],
home_msisdn_prefixes: ["+1313"],

category_2: %{
  enabled: true,
  action: {:error, 3002},
  # Códigos de comando S6a que transportam identidade do assinante
(Seção 3.3.6 do FS.19, Tabela 12)
  protected_command_codes: [316, 317, 318, 319, 320, 321, 323]
}
```




# Categoria 3 — Filtragem de Pacotes da Rede Plausível

**Referência GSMA:** Seção 3.3.7 do FS.19, Seção 7.2.3 do FS.21

A filtragem da Categoria 3 detecta mudanças de localização implausíveis rastreando a última rede vista para cada assinante (IMSI). Quando uma Solicitação de Atualização de Localização chega de uma rede visitada diferente da anterior, o tempo decorrido é comparado com um limite configurável. Uma mudança de domínio ocorrendo mais rápido do que fisicamente possível indica um ataque potencial (por exemplo, atualizações de localização falsificadas).

Este módulo mantém rastreamento por IMSI usando uma tabela ETS com limpeza automática de entradas obsoletas (entradas com mais de 24 horas são removidas periodicamente).

**Verificações realizadas:**

| Verificação | Referência FS.19 | Descrição |
| --- | --- | --- |
| Verificaç��o de Localização Anterior | Seção 3.3.7.1 | Compara o Origin-Realm da ULR atual com o último Origin-Realm visto para esse IMSI |
| Verificação de Velocidade/Tempo | Seção 3.3.7.2 | Marca mudanças de domínio que ocorrem dentro de uma janela de tempo mínima configurável |

```
category_3: %{
  enabled: true,
  # Começar com log_only para observar padrões antes de impor
(Seção 3.3.7 do FS.19)
  action: :log_only,
  # Velocidade máxima plausível em km/h (Seção 3.3.7.2 do FS.19)
  max_velocity_kmh: 1200,
  # Tempo mínimo em segundos entre ULRs de diferentes domínios
para o mesmo IMSI
  min_time_between_updates_seconds: 2
}
```

## Configuração

O Firewall de Diâmetro é habilitado no nível superior, com cada camada de filtragem configurada de forma independente. Os trechos de configuração mostrados acima em cada seção de categoria são combinados sob uma única chave `module_diameter_firewall`:

```
config :dra,
  module_diameter_firewall: %{
    enabled: true,
    home_imsi_prefixes: ["31338", "31339"],
    home_msisdn_prefixes: ["+1313"],
    low_layer: %{ ... },      # Veja Filtragem de Formato de Baixa
Camada acima
    category_1: %{ ... },     # Veja Categoria 1 acima
    category_2: %{ ... },     # Veja Categoria 2 acima
    category_3: %{ ... }      # Veja Categoria 3 acima
  }
```

## Parâmetros de Nível Superior

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `enabled` | Booleano | Sim | `false` | Habilitar ou desabilitar todo o módulo Firewall de Diâmetro. Quando `false`, todas as mensagens passam sem inspeção. |
| `home_imsi_prefixes` | Lista | Não | `[]` | Lista de strings de prefixo IMSI identificando assinantes domiciliares. Usado pela filtragem da Categoria 2. Exemplo: `["31338", "31339"]`. |
| `home_msisdn_prefixes` | Lista | Não | `[]` | Lista de strings de prefixo MSISDN identificando assinantes domiciliares. |

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
| --- | --- | --- | --- | --- |
| | | | | Usado pela filtragem da Categoria 2. Exemplo: `["+1313"]`. |

## Parâmetros de Baixa Camada

| Parâmetro | Tipo | Obrigatório | Padrã |
|---|---|---|---|
| `enabled` | Booleano | Não | `true` |
| `action` | Ação | Não | `{:erro 3002}` |
| `single_instance_avps` | Lista | Não | `[264, 296, 283, 293]` |

| Parâmetro | Tipo | Obrigatório | Padrã |
| --- | --- | --- | --- |
| | | | |
| `enforce_session_id_first` | Booleano | Não | `true` |
| `reject_destination_host_in_answers` | Booleano | Não | `true` |
| `mandatory_avps` | Mapa | Não | `%{}` |

| Parâmetro | Tipo | Obrigatório | Padrã |
|---|---|---|---|
| | | | |

## Parâmetros da Categoria 1

| Parâmetro | Tipo | Obrigatório | Padrão | Descri |
|---|---|---|---|---|
| `enabled` | Booleano | Não | `true` | Habilitar ou desabilitar Categoria 1. |
| `action` | Ação | Não | `{:error, 3007}` | Ação a ser tomada qua violação da Categoria padrão responde com DIAMETER_APPLICATIC |
| `whitelists` | Mapa | Sim | - | Mapa de nome de host para combinações per Aplicação / Código de chave `:all` fornece ur padrão. Entradas espe prioridade. |

## Parâmetros da Categoria 2

| Parâmetro | Tipo | Obrigatório | Padrão | Descr |
|---|---|---|---|---|
| `enabled` | Booleano | Não | `true` | Habilita desabili filtrage Categor |
| `action` | Ação | Não | `{:error, 3002}` | Ação a tomada quando assinan domicili alvo da intercor |
| `protected_command_codes` | Lista | Não | `[]` | Códigos comand que aci verifica de identida do assi domicili Tipicam os códig de com S6a que transpo a identi do assi |

## Parâmetros da Categoria 3

| Parâmetro | Tipo | Obrigatório | Padrão |
|---|---|---|---|
| `enabled` | Booleano | Não | `false` |
| `action` | Ação | Não | `:log_only` |
| `max_velocity_kmh` | Inteiro | Não | `1200` |

| Parâmetro | Tipo | Obrigatório | Padrão |
| --- | --- | --- | --- |
| | | | |
| `min_time_between_updates_seconds` | Inteiro | Não | `2` |

## Eventos de Telemetria

| Evento | Descrição |
|---|---|
| `[:diameter, :firewall, :low_layer, :block]` | Violação de formato de baixa camada detectada e bloqueada |
| `[:diameter, :firewall, :category_1, :block]` | Violação da Categoria 1 detectada e bloqueada |
| `[:diameter, :firewall, :category_2, :block]` | Violação da Categoria 2 detectada e bloqueada |
| `[:diameter, :firewall, :category_3, :block]` | Violação da Categoria 3 detectada e bloqueada |
| `[:diameter, :firewall, :pass, :count]` | Mensagem passou por todas as verificações do firewall |

Todos os eventos incluem metadados: `origin_host`, `application_id`, `command_code`, `application_name`, `command_name` e `reason`.

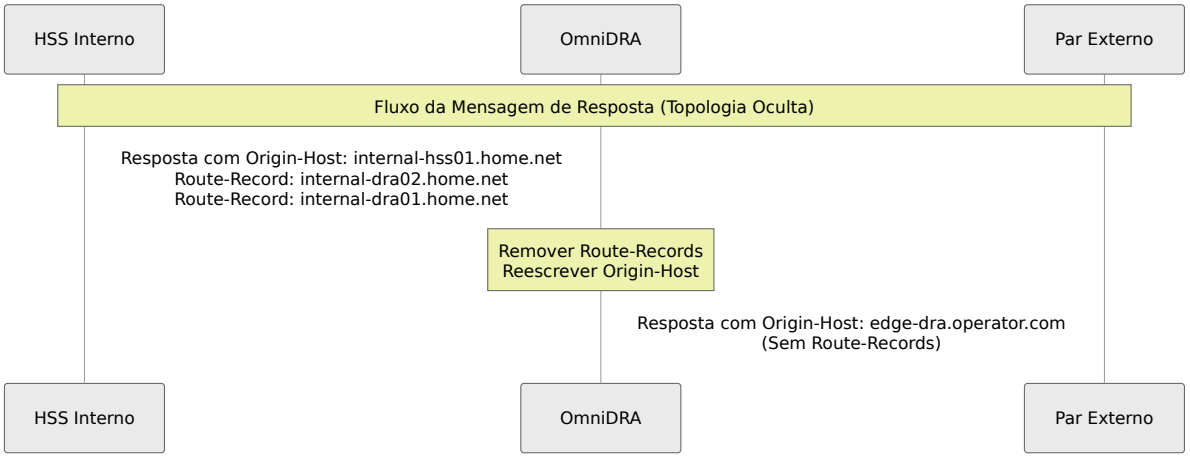
---

# Ocultação de Topologia

**Referência GSMA:** Seção 2.4 do FS.19, Seção 3.4; Seção 3.6 do FS.21

O módulo de Ocultação de Topologia impede que a topologia interna da rede seja exposta a pares externos. Quando mensagens de Diâmetro atravessam vários nós internos, elas acumulam AVPs Route-Record e transportam valores de Origin-Host/Origin-Realm que revelam nomes de host internos, estrutura da rede e contagem de nós. Essas informações podem ser usadas por atacantes para mapear a rede interna para ataques direcionados.

A ocultação de topologia opera no **caminho de saída** — após as decisões de roteamento terem sido tomadas, antes que as mensagens sejam encaminhadas para o par de destino.




# Recursos

| Recurso | Referência FS.19 | Descrição |
|---|---|---|
| Remoção de Route-Record | Seção 2.4 | Remove todos os AVPs Route-Record (282) que revelam caminhos de roteamento internos e nomes de host de nós |
| Reescrita de Origin-Host | Seção 3.4 | Substitui o AVP Origin-Host pela própria identidade do DRA (ou um valor personalizado) em mensagens de resposta |
| Reescrita de Origin-Realm | Seção 3.4 | Opcionalmente substitui o AVP Origin-Realm para ocultar a estrutura do domínio interno |
| Controle por Par | - | Aplicar ocultação de topologia de forma seletiva — apenas a pares externos, ou a todos os pares |

```
module_topology_hiding: %{
  enabled: true,
  # Remover AVPs Route-Record que revelam caminhos internos (Seção
2.4 do FS.19)
  strip_route_records: true,
  # Reescrever Origin-Host em respostas para ocultar nomes de nós
internos (Seção 3.4 do FS.19)
  rewrite_origin_host: %{enabled: true, replacement: :self},
  # Opcionalmente ocultar a estrutura do domínio interno
  rewrite_origin_realm: %{enabled: false, replacement: :self},
  # Aplicar a todos os pares externos, ou listar nomes de host
específicos
  external_peers: :all
}
```

## Parâmetros

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `enabled` | Booleano | Sim | `false` | Habilitar ou desabilitar o módulo de Ocultação de Topologia. |
| `strip_route_records` | Booleano | Não | `true` | Remover todos os AVPs Route-Record (código 282) das mensagens antes de encaminhá-las para pares externos. |
| `rewrite_origin_host` | Mapa | Não | Veja abaixo | Controla a reescrita de Origin-Host em mensagens de resposta. |
| `rewrite_origin_realm` | Mapa | Não | Veja abaixo | Controla a reescrita de Origin-Realm em mensagens de resposta. |

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
| --- | --- | --- | --- | --- |
| `external_peers` | `:all` ou Lista | Não | `:all` | Pares considerados externos. A ocultação de topologia é aplicada apenas às mensagens destinadas a esses pares. Use `:all` para implantações de interconexão Use uma lista de nomes de host para aplicação seletiva. |

**Parâmetros de Reescrita** (aplicam-se tanto a `rewrite_origin_host` quanto a `rewrite_origin_realm`):

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `enabled` | Booleano | Não | varia | Habilitar reescrita para este AVP. O padrão é `true` para Origin-Host, `false` para Origin-Realm. |
| `replacement` | `:self` ou String | Não | `:self` | Valor de substituição. `:self` usa a própria identidade do DRA da configuração `diameter` (`host.realm`). Um valor string é usado como está. |

## Eventos de Telemetria

| Evento | Descrição |
|---|---|
| `[:diameter, :topology_hiding, :route_record, :stripped]` | AVPs Route-Record removidos. A medição inclui `count` de AVPs removidos. |
| `[:diameter, :topology_hiding, :origin_host, :rewritten]` | AVP Origin-Host reescrito |
| `[:diameter, :topology_hiding, :origin_realm, :rewritten]` | AVP Origin-Realm reescrito |

---

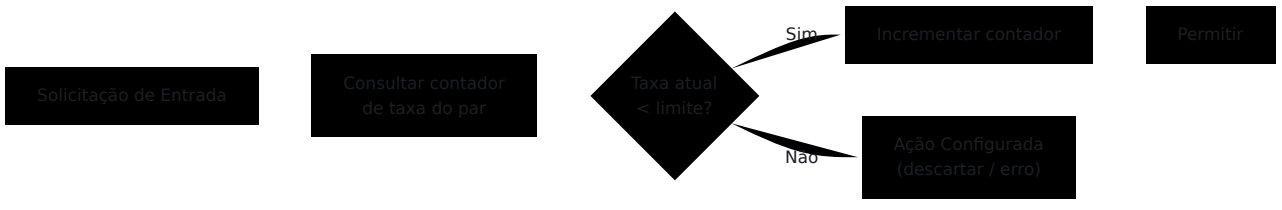
# Limitador de Taxa

**Referência GSMA:** Seção 3.4 do FS.19 (Disponibilidade / Proteção contra DoS)

O Limitador de Taxa aplica limites de taxa de mensagens por par para proteger contra ataques volumétricos e inundação de mensagens. Ele opera como a primeira verificação de segurança no pipeline — antes de qualquer análise de mensagem ou inspeção de conteúdo — para descarregar a carga excessiva o mais cedo possível.

Os limites de taxa são rastreados por par usando um contador de janela deslizante. Cada par tem um contador independente que é redefinido a cada segundo.

```
module_rate_limiter: %{
  enabled: true,
  # Limite padrão para todos os pares (Seção 3.4 do FS.19)
  default_max_requests_per_second: 1000,
  default_action: {:error, 3004},
  # Sobrescritas por par
  peer_limits: %{
    "high-volume-partner.roaming.com" => %
{max_requests_per_second: 5000, action: {:error, 3004}},
    "restricted-peer.roaming.com" => %{max_requests_per_second:
100, action: :drop}
  }
}
```



## Parâmetros

| Parâmetro | Tipo | Obrigatório | Padrão |
|---|---|---|---|
| `enabled` | Booleano | Sim | `false` |
| `default_max_requests_per_second` | Inteiro | Não | `1000` |
| `default_action` | Ação | Não | `{:error, 3004}` |
| `peer_limits` | Mapa | Não | `%{}` |

**Parâmetros de Sobrescrita por Par:**

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `max_requests_per_second` | Inteiro | Não | Herda padrão | Máximo de solicitações por segundo para este par específico. |
| `action` | Ação | Não | Herda padrão | Ação quando este par excede seu limite de taxa. |

## Eventos de Telemetria

| Evento | Descrição |
|---|---|
| `[:diameter, :rate_limiter, :throttled]` | Uma mensagem foi limitada por taxa. As medições incluem `current_rate` e `limit`. |
| `[:diameter, :rate_limiter, :allowed]` | Uma mensagem estava dentro dos limites de taxa. |

---

# Desinfetador de AVP

**Referência GSMA:** Seção 3.3.4, 4.8.1, 4.8.2 do FS.19; Seção 3.6 do FS.21

O Desinfetador de AVP valida e desinfeta AVPs de Diâmetro na fronteira de interconexão. Ele aborda as recomendações da Seção 3.6 do FS.21 para lidar

com ataques de manipulação em nível de protocolo que operam abaixo do nível das categorias de filtragem do FS.19.

## Recursos

| Recurso | Referência GSMA | Descrição |
|---|---|---|
| Remoção de AVP de Fornecedor Desconhecido | Seção 3.6 do FS.21 | Remove AVPs de fornecedores não listados. Previne a injeção de AVPs proprietários que podem acionar comportamentos indesejados em nós de backend. |
| Profundidade de Aninhamento de AVP Agrupados | Seção 3.6 do FS.21 | Impõe uma profundidade máxima de aninhamento para AVPs agrupados. Previne ataques de estouro de pilha usando estruturas de AVP profundamente aninhadas. |

```
module_avp_sanitizer: %{
  enabled: true,
  action: {:error, 3002},
  # Permitir apenas AVPs de fornecedor padrão na interconexão
(Seção 3.6 do FS.21)
  # 0 = IETF, 10415 = 3GPP, 13019 = ETSI, 5535 = 3GPP2
  allowed_vendor_ids: [0, 10415, 13019, 5535],
  strip_unknown_vendor_avps: true,
  # Prevenir estouro de pilha via AVPs agrupados profundamente
aninhados (Seção 3.6 do FS.21)
  max_avp_nesting_depth: 10
}
```

## Parâmetros

| Parâmetro | Tipo | Obrigatório | Padrão | |
|---|---|---|---|---|
| `enabled` | Booleano | Sim | `false` | Habil desab Desin |
| `action` | Ação | Não | `{:error, 3002}` | Ação quan de pr aninh detec remo forne silenc remo ofens bloqu mens |
| `allowed_vendor_ids` | Lista | Não | `[0, 10415, 13019, 5535]` | Lista forne perm interc de ou forne remo |
| `strip_unknown_vendor_avps` | Booleano | Não | `true` | Habil AVPs não n `allow` |
| `max_avp_nesting_depth` | Inteiro | Não | `10` | Profu máxi aninh |

| Parâmetro | Tipo | Obrigatório | Padrão | |
|---|---|---|---|---|
| | | | | agrup<br>Mens<br>exced<br>sujeit<br>confi |

**IDs de Fornecedor Comuns:**

| ID de Fornecedor | Organização | Notas |
|---|---|---|
| 0 | IETF | AVPs padrão de Diâmetro definidos em RFCs |
| 10415 | 3GPP | Todos os AVPs definidos pela 3GPP (S6a, Gx, Rx, etc.) |
| 13019 | ETSI | AVPs definidos pela ETSI |
| 5535 | 3GPP2 | AVPs definidos pela 3GPP2 (interoperabilidade CDMA) |

## Eventos de Telemetria

| Evento | Descrição |
|---|---|
| `[:diameter, :avp_sanitizer, :unknown_vendor, :stripped]` | Um ou mais AVPs de fornecedores desconhecidos foram removidos |
| `[:diameter, :avp_sanitizer, :nesting_depth, :violation]` | Uma mensagem excedeu a profundidade máxima de aninhamento de AVP |
| `[:diameter, :avp_sanitizer, :pass, :count]` | Mensagem passou por todas as verificações de desinfecção |

---

# Recomendações de Implantação

## Ordem Recomendada de Habilitação

Ao implantar os módulos de segurança pela primeira vez, habilite-os de forma incremental para evitar interromper o tráfego ao vivo:

1. **Limitador de Taxa** — Comece com limites generosos e monitore padrões de tráfego. Aperte os limites uma vez que as taxas de referência sejam compreendidas.
2. **Desinfetador de AVP** — Baixo risco de falsos positivos. Habilite a remoção e verificações de aninhamento.
3. **Firewall de Diâmetro (Categoria 1)** — Defina listas brancas com base em acordos de roaming. Comece com combinações de ID de Aplicação / Código de Comando conhecidas e boas.
4. **Firewall de Diâmetro (Baixa Camada)** — Habilite verificações de conformidade de protocolo.
5. **Firewall de Diâmetro (Categoria 2)** — Configure prefixos IMSI/MSISDN domiciliares.

6. **Ocultação de Topologia** — Habilite primeiro a remoção de Route-Record, depois a reescrita de Origin-Host.
7. **Firewall de Diâmetro (Categoria 3)** — Habilite com a ação `:log_only` para observar padrões de localização antes de impor.

## Defesa em Profundidade

Esses módulos implementam o princípio de **defesa em profundidade** descrito na Seção 3.4 do FS.19. Cada camada aborda uma classe diferente de ataque:

| Classe de Ataque | Defesa Primária | Defesa Secundária |
|---|---|---|
| DoS Volumétrico | Limitador de Taxa | Firewall de Diâmetro (todas as categorias) |
| Abuso de Interface | Categoria 1 | Desinfetador de AVP |
| Alvo de Assinante Domiciliar | Categoria 2 | Categoria 1 (restrição de interface) |
| Falsificação de Localização | Categoria 3 | Categoria 2 (verificação de assinante domiciliar) |
| Descoberta de Topologia | Ocultação de Topologia | - |
| Injeção de AVP | Desinfetador de AVP | Filtragem de Baixa Camada |
| Evasão de Protocolo (Duplicação de AVP) | Filtragem de Baixa Camada | Desinfetador de AVP |

# Referência Cruzada de Documentos GSMA

| Módulo / Recurso | FS.19 v10.0 | FS.21 v12.0 |
|---|---|---|
| Filtragem de Formato de Baixa Camada | Seção 3.3.4 | Seção 7.3.1 |
| Filtragem da Categoria 1 | Seção 3.3.5, Anexo B.3.3 | Seção 7.2.1 |
| Filtragem da Categoria 2 | Seção 3.3.6, Anexo B.3.4 | Seção 7.2.2, Seção 16 |
| Filtragem da Categoria 3 | Seção 3.3.7, Anexo B.3.5 | Seção 7.2.3 |
| Ocultação de Topologia | Seção 2.4, Seção 3.4 | Seção 3.6 |
| Limitação de Taxa | Seção 3.4 | - |
| Desinfecção de AVP | Seção 4.8.1, 4.8.2 | Seção 3.6 |
| Defesa em Profundidade | Seção 3.4 | Seção 3.15 |
| Categorias de Filtragem (Independente de Protocolo) | Anexo A | Seção 7 |

# Função de Consulta de Assinante (SLF)

A Função de Consulta de Assinante (SLF) aprende dinamicamente qual elemento de rede está atendendo cada assinante ao observar o tráfego de sinalização Diameter que passa pelo DRA. Ele usa essas ligações aprendidas para rotear solicitações subsequentes — como consultas de serviço de localização — diretamente para o nó de atendimento correto, sem exigir regras de roteamento estáticas.

Isso é particularmente útil para solicitações de serviço de localização SLg/SLh, onde o GMLC precisa alcançar o MME de atendimento para um determinado assinante, mas não tem conhecimento prévio de qual MME é esse.

O conceito de SLF é descrito na 3GPP TS 29.172 e na 3GPP TS 29.173 para serviços de localização, com o registro de assinantes definido na 3GPP TS 29.272.

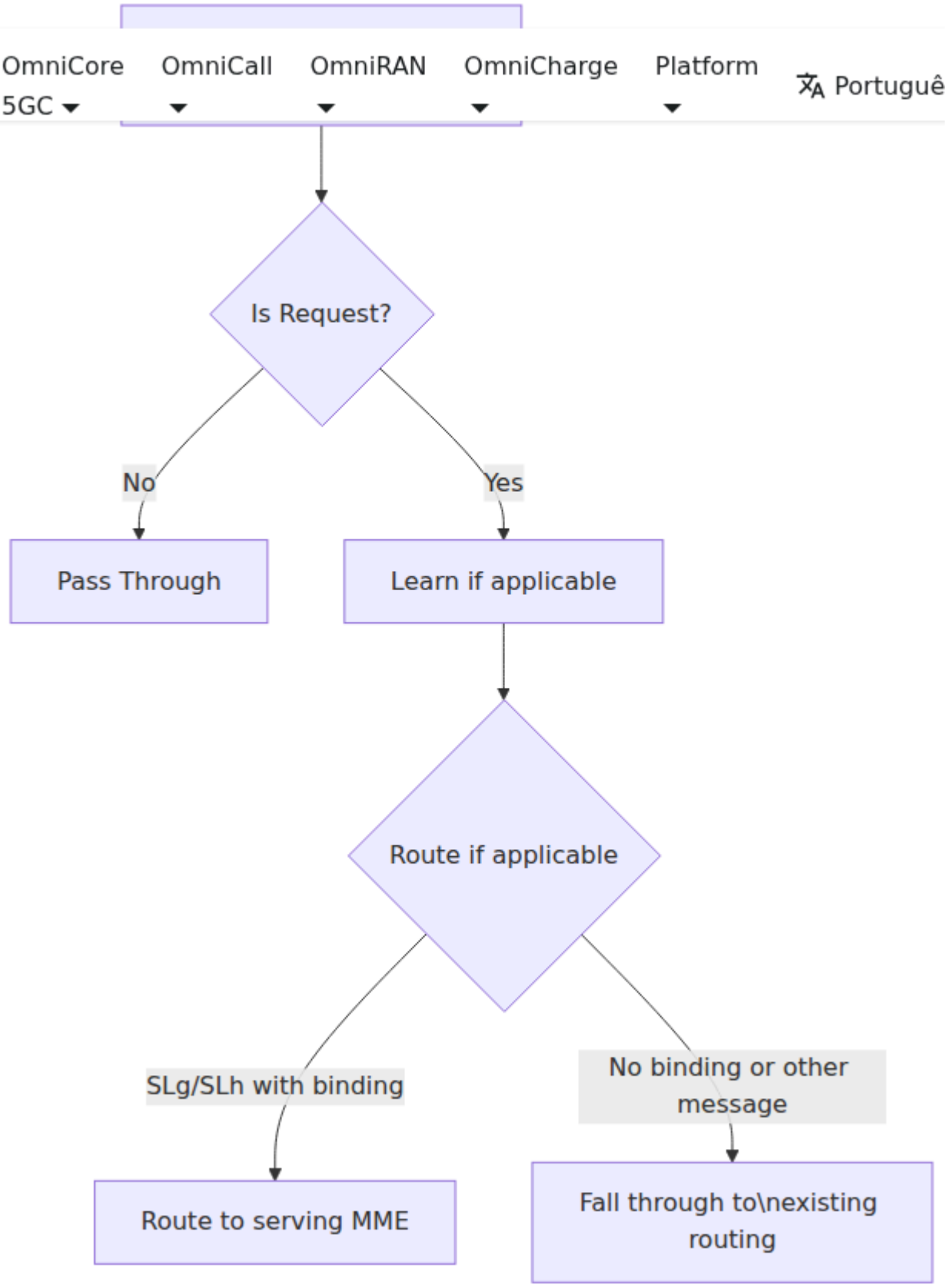
## Como Funciona

O módulo opera em duas fases: **aprendizado** e **roteamento**.

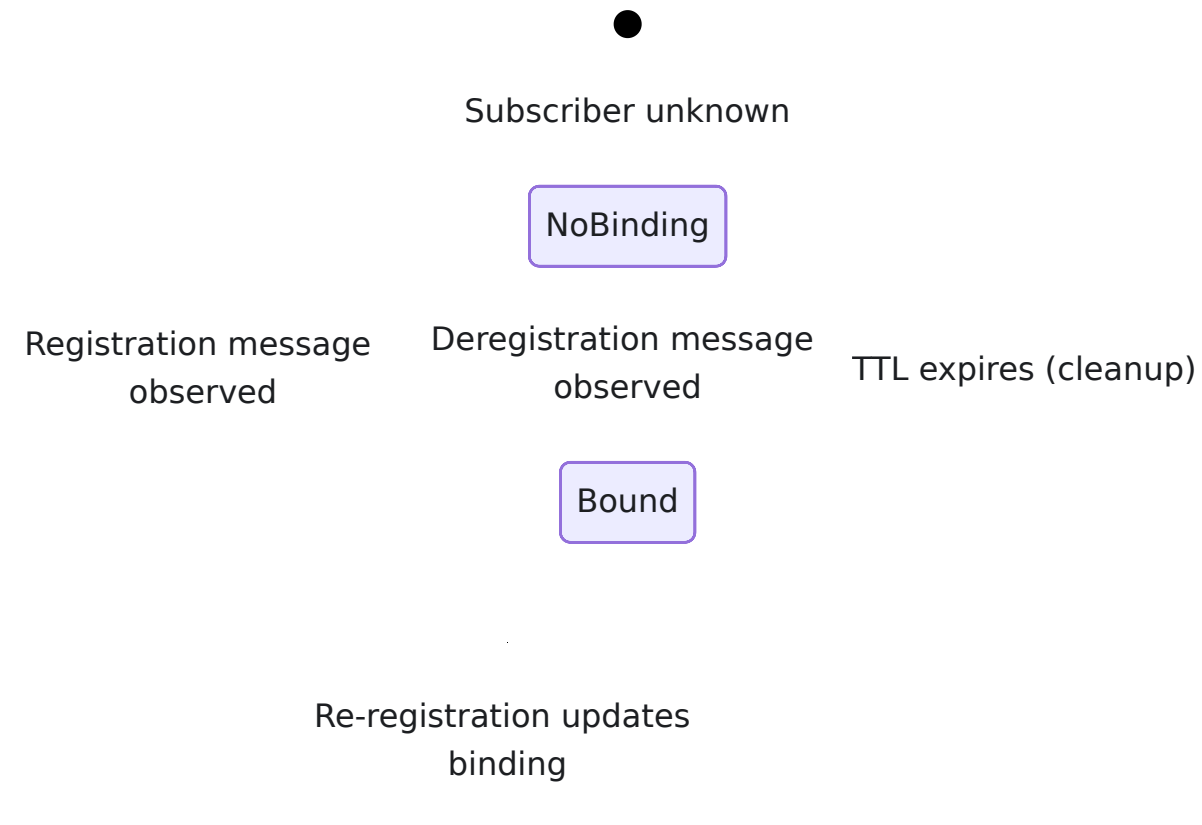
Durante o **aprendizado**, o módulo observa passivamente mensagens de registro e desregistro que fluem pelo DRA. Quando um assinante se registra (por exemplo, via um S6a Update-Location-Request), o módulo registra qual elemento de rede está agora atendendo esse assinante.

Durante o **roteamento**, quando uma solicitação chega que precisa alcançar o nó de atendimento de um assinante (por exemplo, um SLg Provide-Location-Request), o módulo consulta a ligação e roteia diretamente para o par correto.

Se nenhuma ligação existir para um assinante, a solicitação cai na lógica de roteamento existente (incluindo Roteamento Avançado).

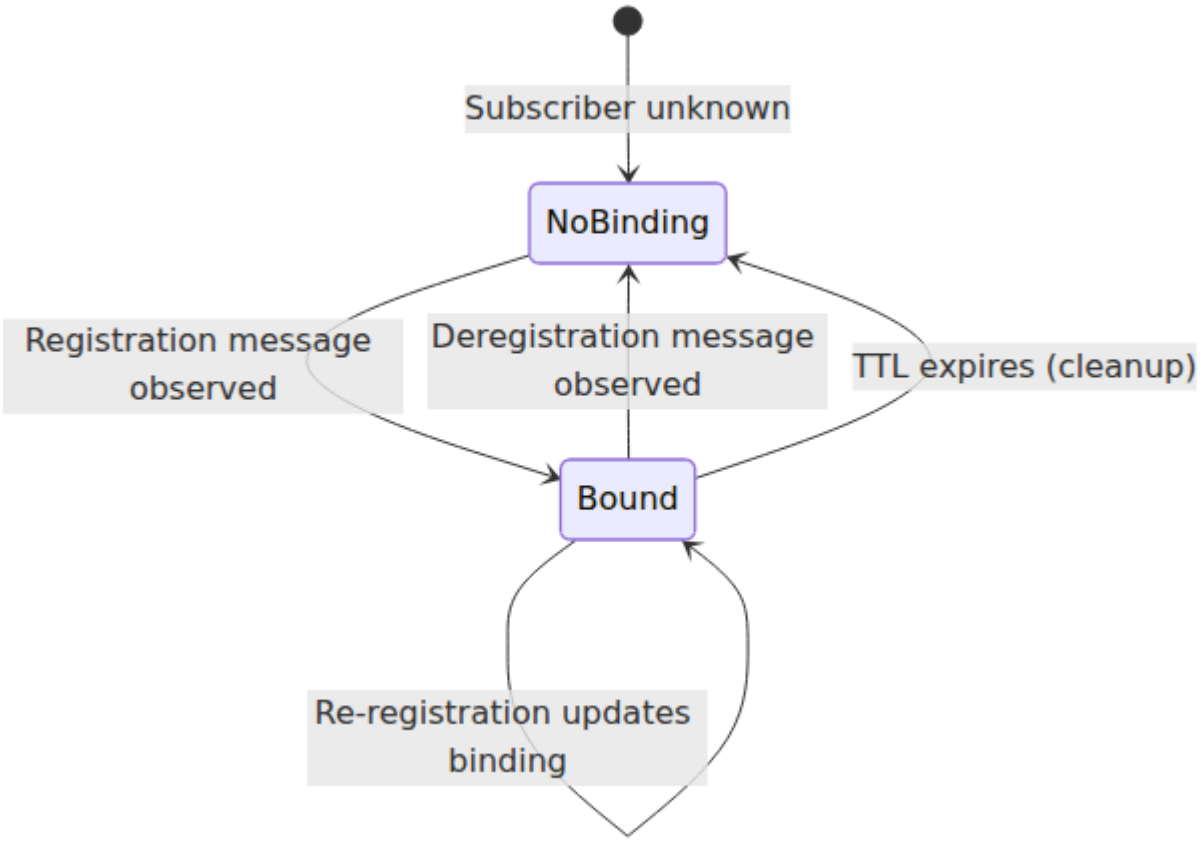

Is Request?
No
Yes
Pass Through
Learn if applicable
Route if applicable
SLg/SLh with binding
No binding or other message
Route to serving MME
Fall through to\nexisting routing

# Ciclo de Vida da Ligação




## Aprendizado: Mensagens que Criam Ligações

O módulo aprende ligações de assinante para nó de atendimento a partir das seguintes mensagens Diameter:




| Interface | Mensagem | Código-Comando | Ligação Criada | Fonte IMSI |
|---|---|---|---|---|
| S6a | Update-Location-Request (ULR) | 316 | `serving_mme` | User-Name AVP (1) |
| Gx | Credit-Control-Request Initial (CCR-I) | 272 (CC-Request-Type=1) | `serving_pgw` | Subscription-Id AVP (443) |
| Cx | Server-Assignment-Request (SAR) | 301 (Server-Assignment-Type=1) | `serving_cscf` | Public-Identity AVP (601) |

## Aprendizado: Mensagens que Removem Ligações

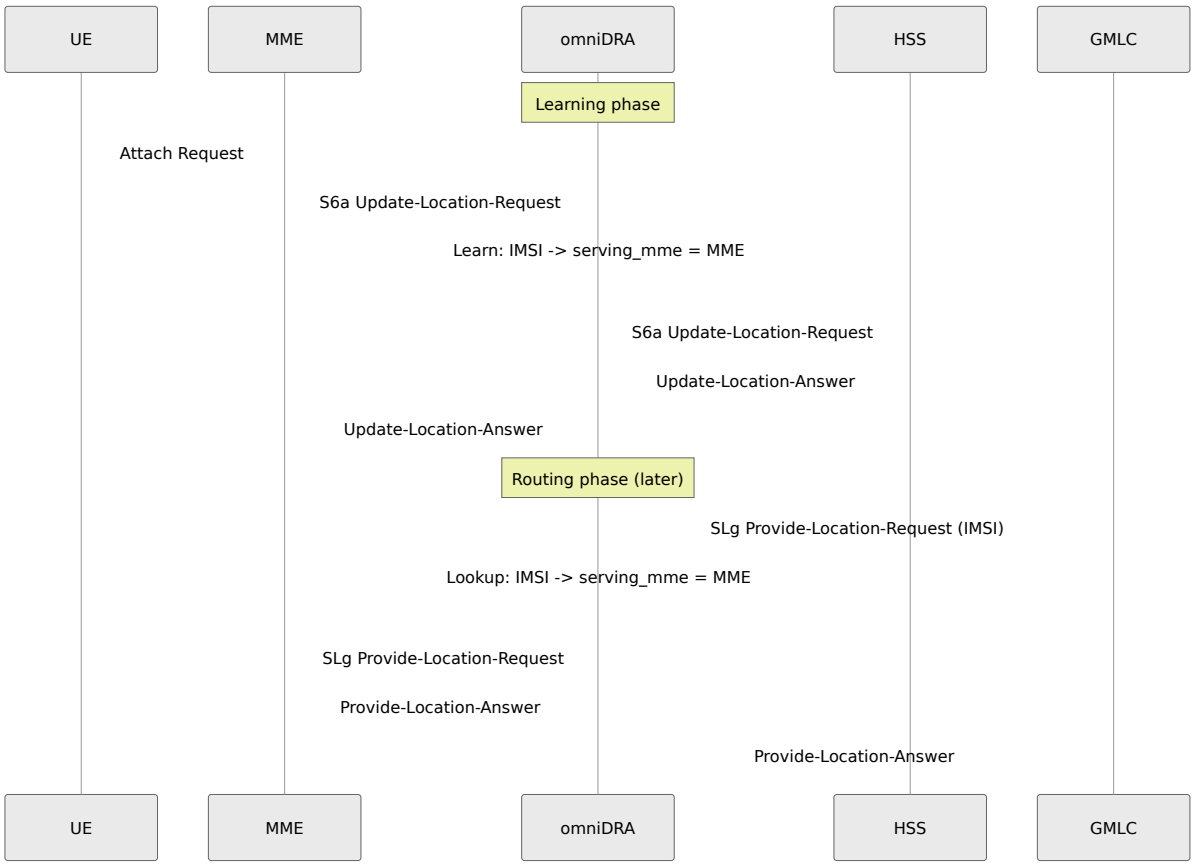
| Interface | Mensagem | Código-Comando | Ligação Removida |
|---|---|---|---|
| S6a | Cancel-Location-Request (CLR) | 317 | `serving_mme` |
| S6a | Purge-UE-Request (PUR) | 321 | `serving_mme` |
| Gx | Credit-Control-Request Termination (CCR-T) | 272 (CC-Request-Type=3) | `serving_pgw` |
| Cx | Server-Assignment-Request (SAR) | 301 (Server-Assignment-Type=4) | `serving_cscf` |

Quando a última ligação para um assinante é removida, todo o registro do assinante é excluído da tabela.

## Roteamento: Mensagens que Usam Ligações

| Interface | Mensagem | Código-Comando | Ligação Usada |
|---|---|---|---|
| SLg | Provide-Location-Request (PLR) | 8388620 | `serving_mme` |
| SLh | LCS-Routing-Info-Request (RIR) | 8388622 | `serving_mme` |

# Fluxo de Exemplo: Solicitação de Serviço de Localização




# Posição no Pipeline

A Consulta de Assinante é executada **antes** do Roteamento Avançado no pipeline de processamento de solicitações. Se o SLF encontrar uma ligação e substituir a rota, o Roteamento Avançado ainda será executado, mas a rota do SLF terá precedência.


# Configuração

O módulo é configurado sob `module_subscriber_lookup` em `config/runtime.exs`.

```
module_subscriber_lookup: %{
  # Enable or disable the module
  enabled: false,

  # How long to keep bindings before expiring (seconds)
  binding_ttl_seconds: 86400
}
```

## Parâmetros

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `enabled` | Booleano | Sim | `false` | Habilitar ou desabilitar o módulo de Consulta de Assinante. Quando `false`, todas as solicitações passam sem modificações e nenhuma ligação é aprendida. |
| `binding_ttl_seconds` | Inteiro | Não | `86400` | Tempo em segundos antes que uma ligação de assinante expire e seja removida durante a limpeza. O padrão é 24 horas. A limpeza é executada na metade do intervalo TTL, com um mínimo de |

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| | | | | 60 segundos. |

# Exemplos de Configuração

### Implantação Padrão

Adequado para a maioria das redes onde os assinantes se registram novamente dentro de 24 horas.

```
module_subscriber_lookup: %{
  enabled: true,
  binding_ttl_seconds: 86400
}
```

**Como funciona:** O módulo aprende as ligações MME/PGW/CSCF a partir do tráfego que passa e as mantém por 24 horas. Solicitações SLg e SLh são automaticamente roteadas para o MME de atendimento correto. As ligações são atualizadas sempre que uma nova mensagem de registro é observada para o mesmo assinante.

**Caso de uso:** Redes com requisitos de serviço de localização (LCS) onde o GMLC precisa alcançar o MME de atendimento sem mapeamentos estáticos de assinante para MME.

### Ambiente de Alta Rotatividade

Para redes com mobilidade frequente de assinantes onde ligações obsoletas devem expirar rapidamente.

```
module_subscriber_lookup: %{
  enabled: true,
  binding_ttl_seconds: 3600
}
```

**Como funciona:** As ligações expiram após 1 hora. Isso é apropriado quando os assinantes se movem frequentemente entre MMEs e ligações obsoletas causariam solicitações de localização mal direcionadas. O intervalo de limpeza é executado a cada 30 minutos.

**Caso de uso:** Redes urbanas densas ou eventos com alta mobilidade de assinantes.

# Métricas

## Atualizações de Ligações

**Métrica**: `diameter.subscriber_lookup.binding.update` **Tipo**: Contador **Descrição**: Incrementado cada vez que uma ligação de assinante é criada ou atualizada. **Rótulos**:

- `imsi` - IMSI do assinante
- `binding_type` - Tipo de ligação: `serving_mme`, `serving_pgw` ou `serving_cscf`
- `serving_host` - Origin-Host do elemento de rede de atendimento

## Exclusões de Ligações

**Métrica**: `diameter.subscriber_lookup.binding.delete` **Tipo**: Contador **Descrição**: Incrementado cada vez que uma ligação de assinante é explicitamente removida via uma mensagem de desregistro. **Rótulos**:

- `imsi` - IMSI do assinante
- `binding_type` - Tipo de ligação removida

## Solicitações Roteadas

**Métrica**: `diameter.subscriber_lookup.route.count` **Tipo**: Contador **Descrição**: Incrementado cada vez que uma solicitação SLg/SLh é roteada com sucesso usando uma ligação aprendida. **Rótulos**:

- `imsi` - IMSI do assinante
- `binding_type` - Tipo de ligação usado para roteamento (atualmente sempre `serving_mme`)
- `serving_host` - Origin-Host do par para o qual a solicitação foi roteada

# Solução de Problemas

## Solicitações de Localização Não Sendo Roteadas para o MME de Atendimento

**Sintomas**: Solicitações SLg Provide-Location ou SLh LCS-Routing-Info não estão sendo roteadas para o MME esperado, caindo no roteamento padrão em vez disso.

**Possíveis causas**:

- O assinante não se registrou através deste DRA, portanto, nenhuma ligação existe
- A ligação expirou (TTL excedido)
- O par MME de atendimento não está conectado a este DRA
- `enabled` está definido como `false`

**Resolução**:

1. Verifique se o módulo está habilitado na configuração
2. Verifique se o tráfego S6a ULR para o assinante flui através deste DRA (as ligações são aprendidas apenas a partir do tráfego observado)
3. Verifique se `binding_ttl_seconds` é longo o suficiente para cobrir a lacuna entre o registro e a solicitação de localização
4. Confirme se o MME de atendimento está conectado como um par

## Ligações Não Sendo Aprendidas

**Sintomas**: A tabela de ligações permanece vazia apesar do tráfego S6a/Gx/Cx passando pelo DRA.

**Possíveis causas**:

- O módulo não está habilitado
- O processo do módulo não está em execução (verifique o supervisor)
- As mensagens não contêm um IMSI válido no AVP esperado

**Resolução**:

1. Verifique `enabled: true` na configuração
2. Confirme se o DRA foi reiniciado após a alteração da configuração
3. Verifique os logs de depuração do DRA para entradas `SubscriberLookup: Learned` para confirmar a atividade de ligação

# Ligações Obsoletas Causando Solicitações Mal Roteadas

**Sintomas**: Solicitaç��es de localização estão sendo roteadas para um MME que não está mais atendendo o assinante.

**Possíveis causas**:

- O Cancel-Location-Request (CLR) ou Purge-UE-Request (PUR) não passaram por este DRA
- `binding_ttl_seconds` está definido muito alto para o padrão de mobilidade da rede

**Resolução**:

1. Reduza `binding_ttl_seconds` para corresponder ao intervalo esperado de re-registro do assinante
2. Certifique-se de que todo o tráfego de desregistro S6a flua através deste DRA

# Referência

## IDs de Aplicação

| ID | Interface | Descrição | Referência |
|---|---|---|---|
| 16777251 | S6a/S6d | Autenticação e gerenciamento de assinatura de MME/SGSN para HSS | 3GPP TS 29.272 |
| 16777238 | Gx | Controle de políticas e cobrança de PCEF para PCRF | 3GPP TS 29.212 |
| 16777216 | Cx | Registro IMS de I-CSCF/S-CSCF para HSS | 3GPP TS 29.229 |
| 16777255 | SLg | Serviços de localização de GMLC para MME | 3GPP TS 29.172 |
| 16777291 | SLh | Informações de roteamento LCS de GMLC para HSS/DRA | 3GPP TS 29.173 |

## Códigos de Comando

| Código | Nome | Interface | Descrição |
|---|---|---|---|
| 272 | Credit-Control-Request/Answer (CCR/CCA) | Gx | Controle de política e cobrança em nível de sessão |
| 301 | Server-Assignment-Request/Answer (SAR/SAA) | Cx | Registro e desregistro IMS |
| 316 | Update-Location-Request/Answer (ULR/ULA) | S6a | Atualização de localização do assinante no MME |
| 317 | Cancel-Location-Request/Answer (CLR/CLA) | S6a | Cancelamento de localização iniciado pelo HSS |
| 321 | Purge-UE-Request/Answer (PUR/PUA) | S6a | Purga de UE iniciada pelo MME |
| 8388620 | Provide-Location-Request/Answer (PLR/PLA) | SLg | Solicitação de serviço de localização para o MME de atendimento |
| 8388622 | LCS-Routing-Info-Request/Answer (RIR/RIA) | SLh | Consulta de informações de roteamento LCS |

## Tipos de Ligação

| Tipo de Ligação | Aprendido De | Usado Por | Descrição |
|---|---|---|---|
| `serving_mme` | S6a ULR | SLg PLR, SLh RIR | O MME que atualmente atende o assinante |
| `serving_pgw` | Gx CCR-I | — | O PGW que gerencia a sessão do assinante (reservado para roteamento futuro) |
| `serving_cscf` | Cx SAR (Registro) | — | O S-CSCF que atende o registro IMS do assinante (reservado para roteamento futuro) |