Roteamento Avançado

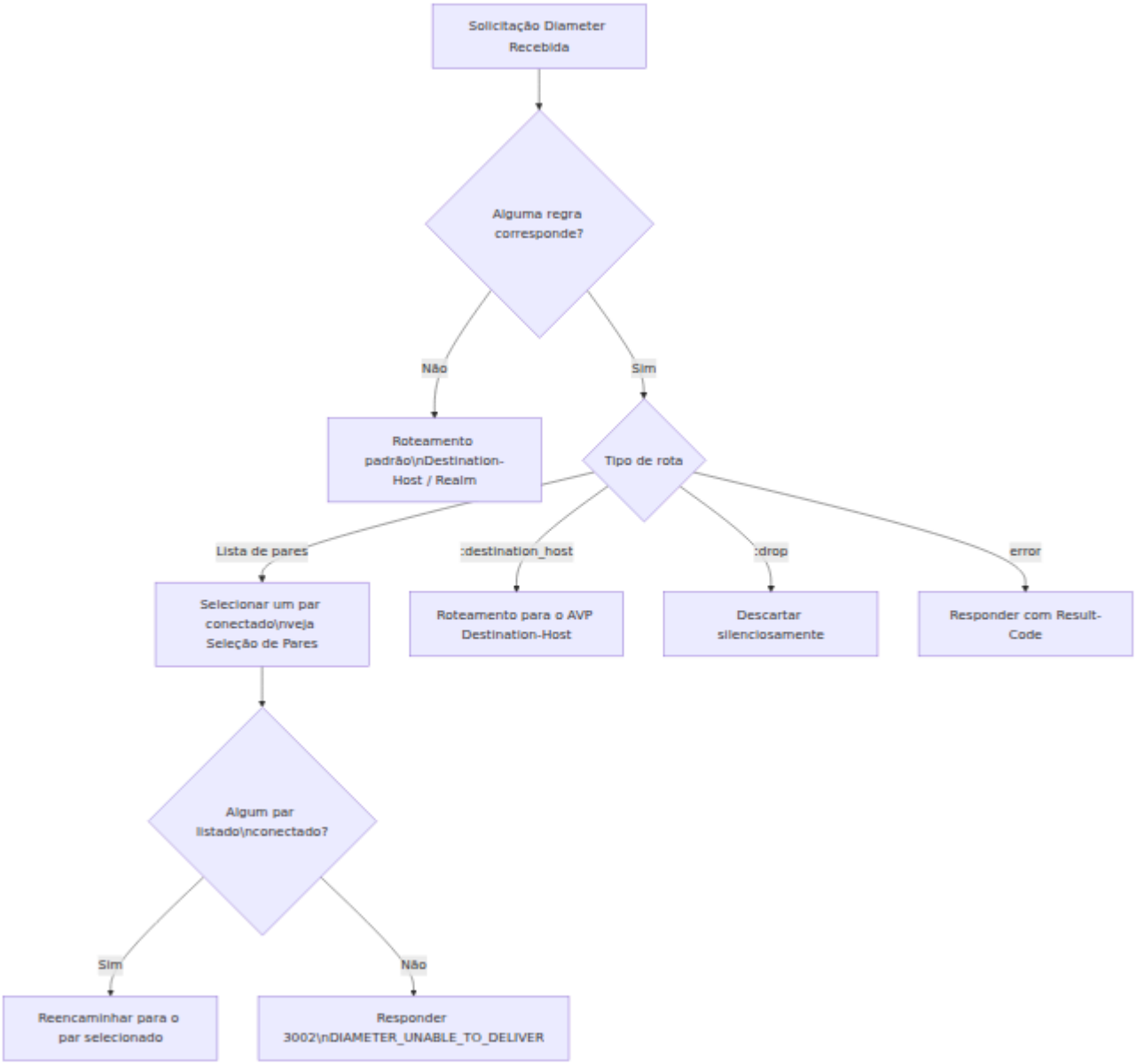
O Roteamento Avançado permite que o DRA tome decisões de roteamento com base no conteúdo de cada solicitação Diameter. Ele não usa apenas **Destination-Realm**. **Regras** correspondem a propriedades da mensagem, como par de origem, Application-Id, Command-Code e valores AVP. Quando uma regra corresponde, ela direciona a solicitação para um conjunto escolhido de pares ou um destino especial.

Uma regra também pode distribuir o tráfego entre vários pares para **failover** e **balanceamento de carga**. Ela utiliza a `priority` e `weight` por par. Para aplicações com estado, como Gx (PCRF) e Gy (OCS), o Roteamento Avançado preserva a **afinidade de sessão**. Mensagens durante a sessão sempre retornam ao par que possui a sessão.

O Roteamento Avançado avalia **apenas pacotes de solicitação Diameter de entrada**. Pacotes de resposta seguem o roteamento de sessão estabelecido de volta ao par de origem. O DRA nunca os reavalia em relação a essas regras. Para reescrever o conteúdo da mensagem, veja Transformação Avançada. Para roteamento baseado em realm/host que é executado quando nenhuma regra corresponde, veja Roteamento Padrão.

Como Funciona

O DRA avalia cada solicitação de entrada em relação às regras configuradas **em ordem, de cima para baixo**. A **primeira** regra cujos filtros correspondem vence e decide a rota. O DRA não avalia as regras posteriores. Se nenhuma regra corresponder, a solicitação continua com o roteamento padrão **Destination-Host** / **Destination-Realm**.



Uma regra correspondente pode resolver apenas para pares que **não estão atualmente conectados**. Nesse caso, o DRA responde com **DIAMETER_UNABLE_TO_DELIVER (3002)**. O DRA nunca devolve a solicitação para o roteamento padrão. O roteamento padrão a entregaria ao nó errado e retornaria um resultado enganoso.

Posição do Pipeline

O Roteamento Avançado é executado após o Direcionamento de Roaming e a Pesquisa de Assinantes. Sua decisão substitui o par que o roteamento padrão selecionaria.

Processamento de Regras

O DRA avalia as regras de cima para baixo. A **primeira regra correspondente vence**. Cada regra combina seus filtros de acordo com seu modo de `match`. Ela aplica sua ação de `route` assim que corresponde. Se nenhuma regra corresponder, o DRA recua para o roteamento padrão.

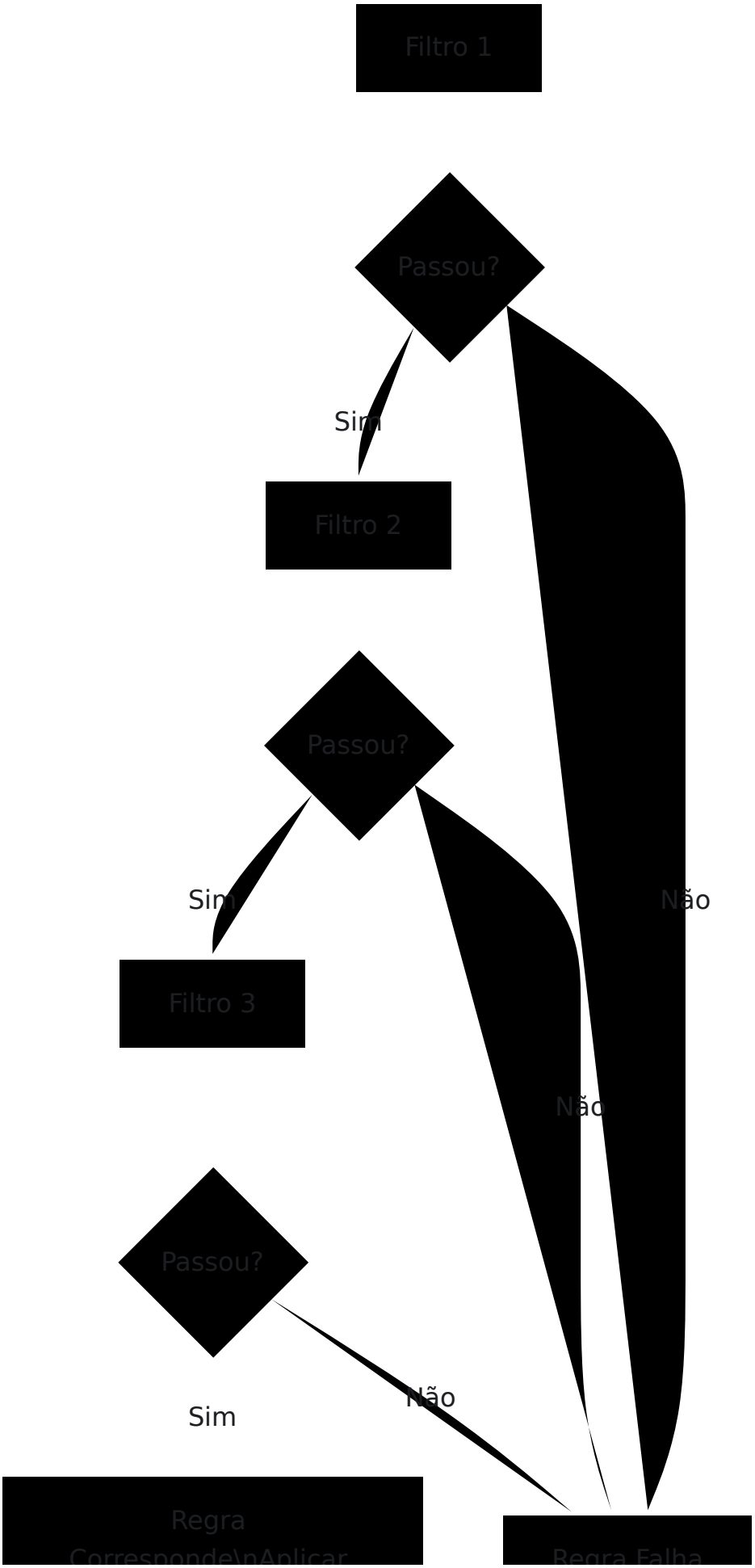
1. O DRA avalia as regras **em ordem de cima para baixo**, conforme definido na configuração.
2. O DRA combina os filtros dentro de uma regra de acordo com o modo de `match` da regra (`:all`, `:any` ou `:none`).
3. A **primeira regra correspondente vence**. O DRA não avalia as regras posteriores.
4. Se nenhuma regra corresponder, o DRA utiliza o roteamento padrão.

Modos de Correspondência

O parâmetro `match` determina como o DRA combina os filtros de uma regra.

`:all` (lógica AND)

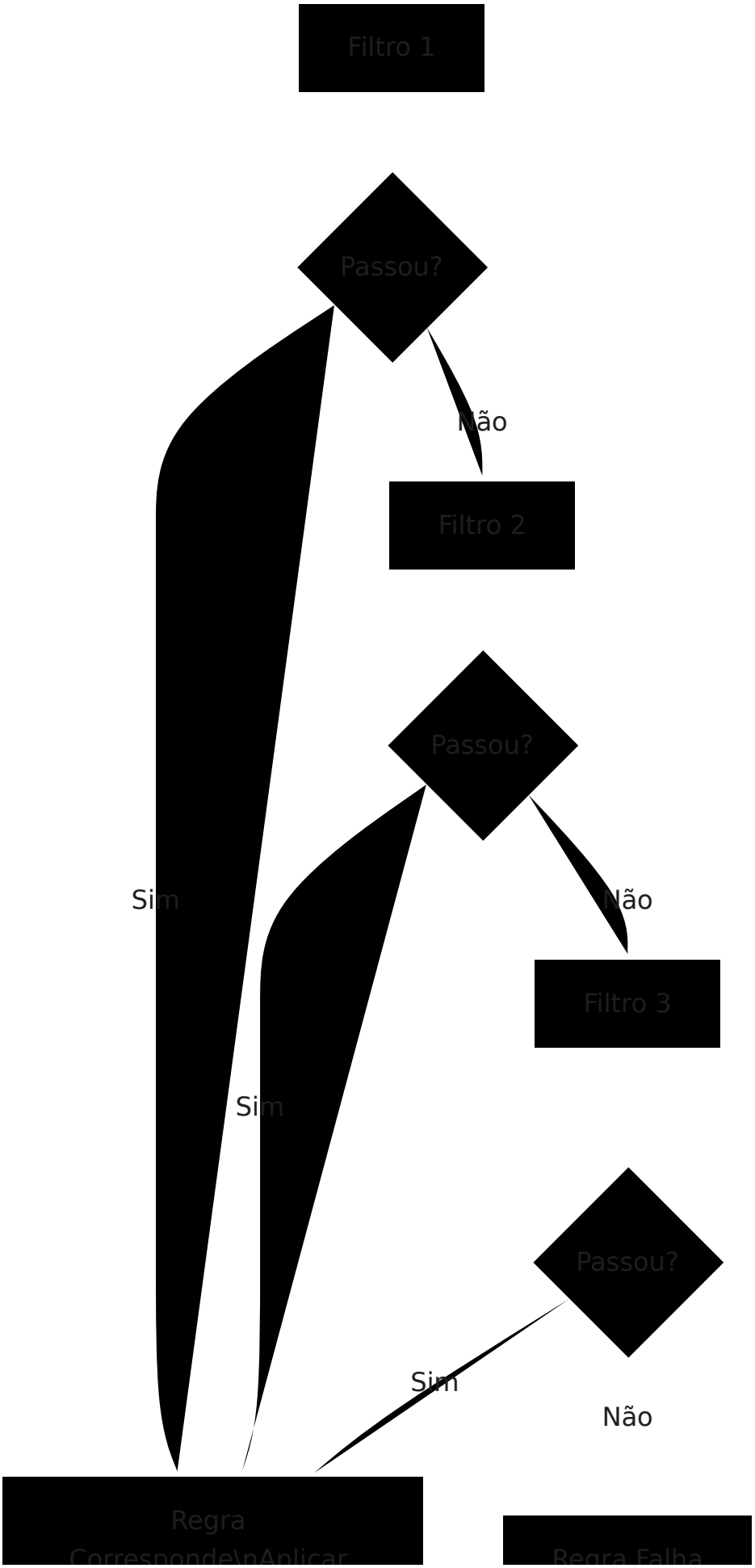
Cada filtro deve corresponder para que a regra tenha sucesso. Com três filtros, `filter1 AND filter2 AND filter3` deve ser verdadeiro.


Filtro 1
Passou?
Sim
Filtro 2
Passou?
Sim
Não
Filtro 3
Não
Passou?
Não
Sim
Regra Corresponde\nAplicar
Regra Falha

[redacted]	[redacted]
Ação	

`:any` (lógica OR)

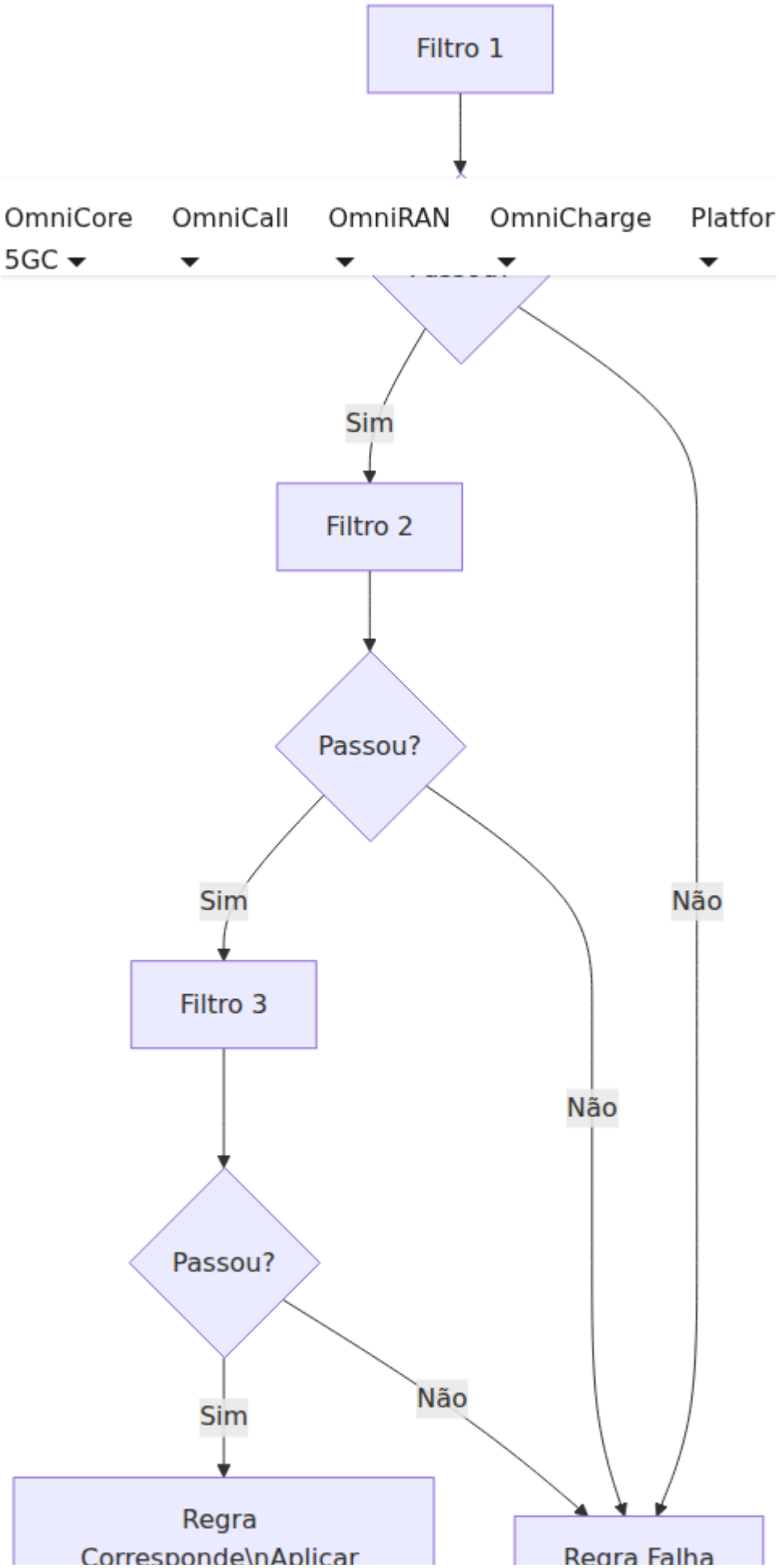
Pelo menos um filtro deve corresponder. Com três filtros, `filter1 OR filter2 OR filter3` tem sucesso se algum passar.


Filtro 1
Passou?
Não
Filtro 2
Passou?
Sim
Não
Filtro 3
Sim
Passou?
Sim
Não
Regra Corresponde\nAplicar
Regra Falha

[redacted]	[redacted]
Ação	

`:none` (lógica NOR)

Nenhum filtro pode corresponder. Esta é uma correspondência inversa. Com três filtros, `NOT filter1 AND NOT filter2 AND NOT filter3` deve ser verdadeiro para que a regra tenha sucesso. Cada filtro deve falhar.


Filtro 1
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platfor
Sim
Filtro 2
Passou?
Sim
Não
Filtro 3
Não
Passou?
Sim
Não
Regra Corresponde\nAplicar
Regra Falha

Seleção de Pares

Quando uma regra roteia para uma lista de pares, o DRA seleciona apenas entre pares que estão **atualmente conectados**. A forma como você escreve os pares determina como o DRA escolhe entre eles.

Nomes de host simples

Uma lista de nomes de host simples segue o `peer_selection_algorithm` global (`:failover` ou `:random`) configurado sob `diameter`. Este é o comportamento original e não muda. Você deve definir cada par na configuração Diameter do DRA. Nomeie cada par pelo seu **exato** nome de host, que é sensível a maiúsculas e minúsculas. Cada par deve estar atualmente conectado. O DRA ignora pares desconectados.

```
route: %{peers: ["hss01.example.com", "hss02.example.com"]}
```

- `:failover` - sempre o primeiro par conectado na lista. Isso fornece um primário determinístico, depois um secundário.
- `:random` - um par conectado aleatório. Isso distribui a carga de maneira uniforme.

Prioridade e peso

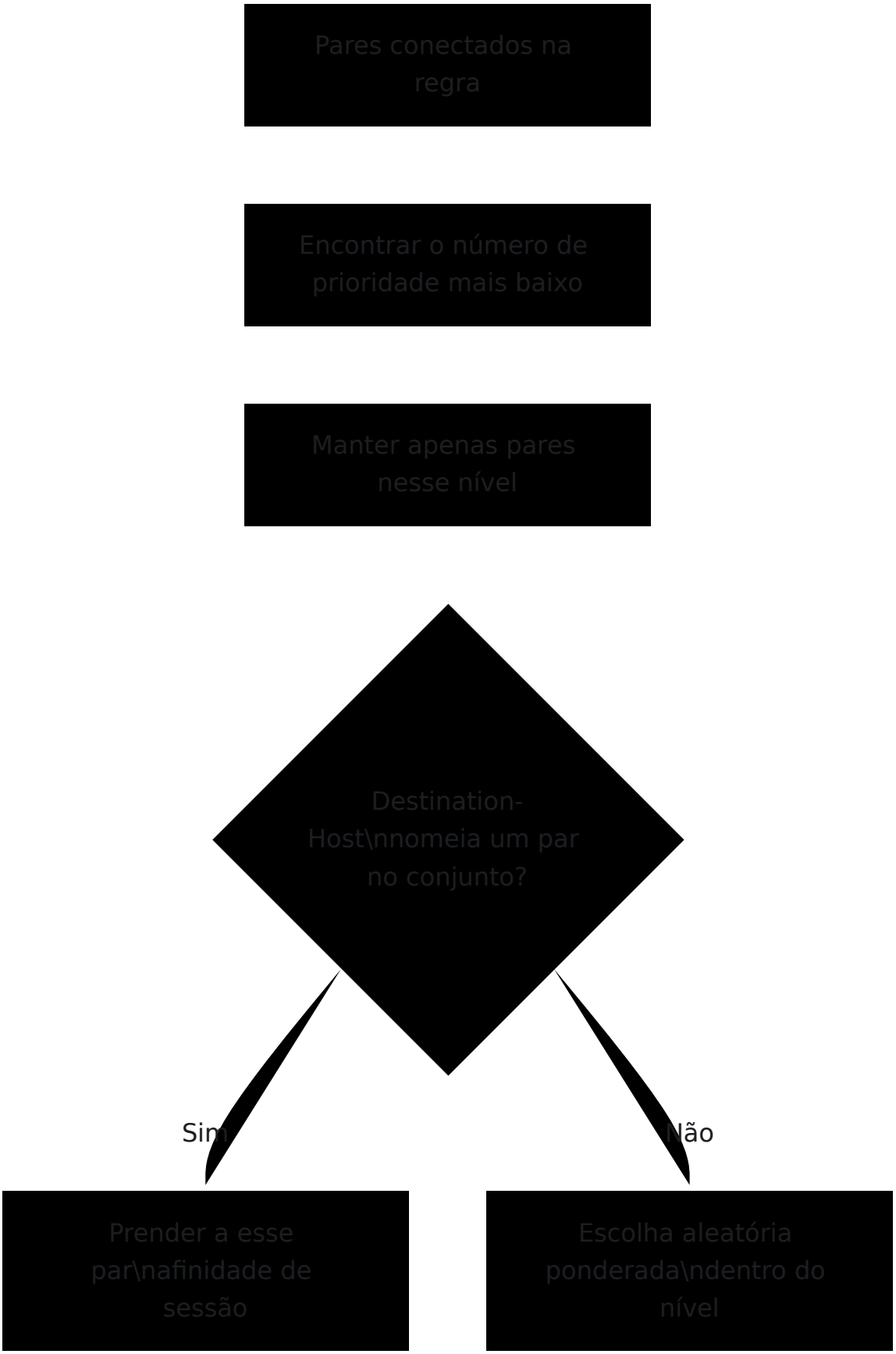
Para failover e balanceamento de carga por regra, escreva os pares como mapas com `priority` e `weight`. Isso é independente do algoritmo global.

```
route: %{
  peers: [
    %{host: "pcrf01.example.com", priority: 10, weight: 50},
    %{host: "pcrf02.example.com", priority: 10, weight: 50},
    %{host: "pcrf-dr.example.com", priority: 20, weight: 100}
  ]
}
```

A seleção é **"prioridade escolhe o nível, peso divide a carga"**:

1. O DRA considera apenas pares conectados.
2. O **número de prioridade mais baixo** presente vence o nível. Um número menor é preferido, assim como nos registros DNS SRV. Níveis numerados mais altos atuam como backups. O DRA utiliza um nível de backup apenas após todos os pares conectados em um nível inferior estarem indisponíveis. Isso fornece **failover**.
3. Dentro do nível vencedor, o DRA escolhe um par aleatoriamente, **ponderado por `weight`**. Pesos iguais distribuem a carga uniformemente. Pesos desiguais tendem a enviesar a divisão. Isso fornece **balanceamento de carga**.

Se você omitir `priority` ou `weight`, cada um terá o valor padrão de `100`.



Este modelo único expressa todas as topologias comuns:

Objetivo	Como expressá-lo
Failover puro	Prioridades distintas em ordem decrescente (por exemplo, 10, 20, 30); peso irrelevante
Balanceamento de carga puro	Uma prioridade para todos; pesos iguais
Balanceamento de carga ponderado	Uma prioridade para todos; pesos na proporção desejada (por exemplo, 70 / 30)
Balanceamento de carga com backup	Dois pares na prioridade 10, um terceiro na prioridade 20

Afinidade de Sessão (Gx / Gy)

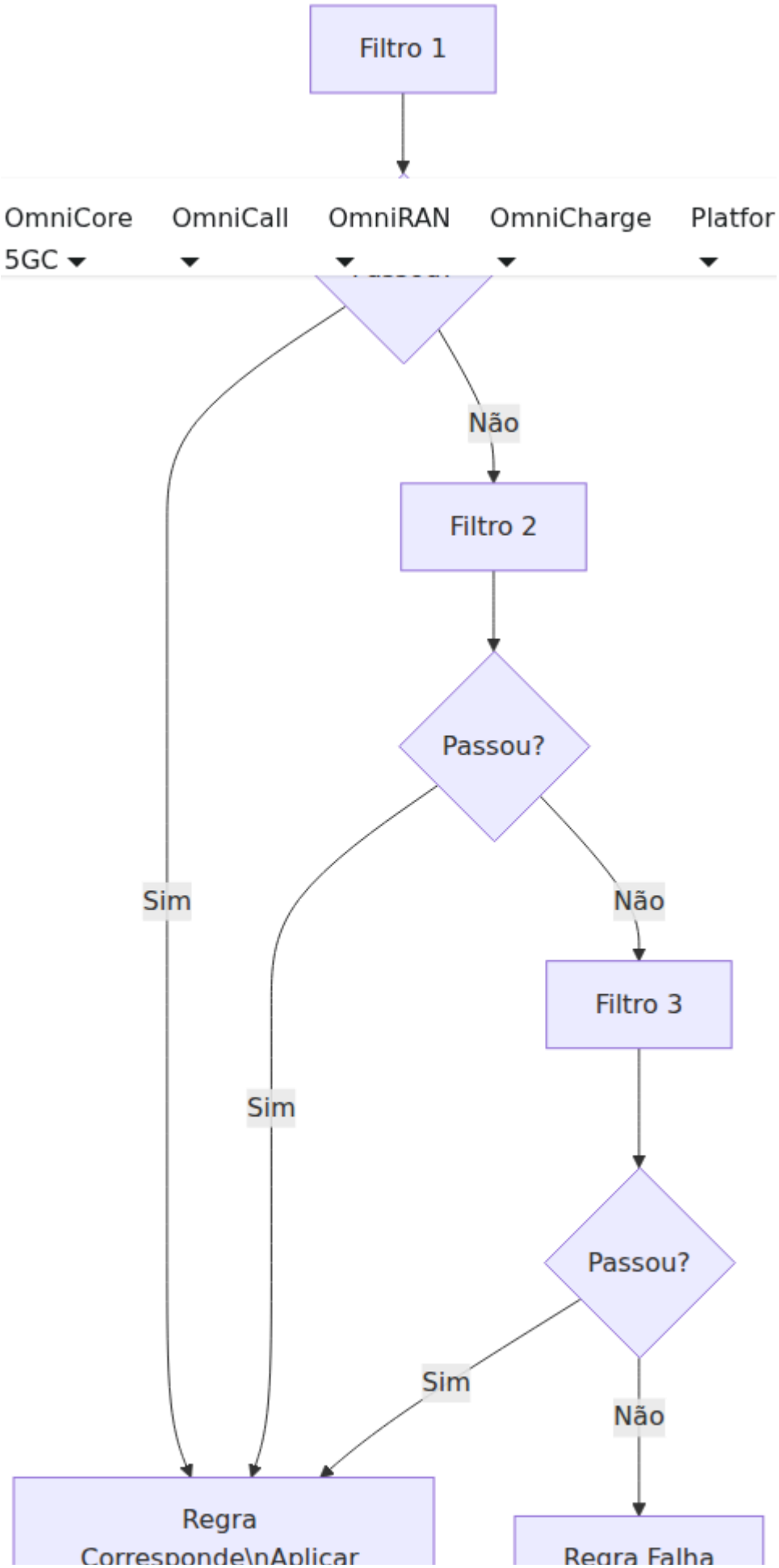
Aplicações com estado carregam um **Session-Id** durante a vida de uma sessão. Mensagens durante a sessão devem alcançar o **mesmo** servidor que lidou com a solicitação que iniciou a sessão. Se o DRA reequilibrar uma mensagem durante a sessão para um par diferente, esse par a rejeita com **DIAMETER_UNKNOWN_SESSION_ID (5002)**.

O Roteamento Avançado preserva a afinidade com o mecanismo Diameter padrão, o **AVP Destination-Host (293)**. Quando uma regra usa pares de prioridade/peso:

- Se a solicitação carrega um Destination-Host que nomeia um dos pares conectados da regra, o DRA **prende** a solicitação a esse par. Isso substitui prioridade e peso.
- Se não houver um Destination-Host desse tipo, o DRA balanceia a carga da solicitação normalmente.

A solicitação que inicia a sessão (por exemplo, Gx CCR-Initial) não carrega Destination-Host. O DRA a balanceia entre o nível. A resposta retorna o Origin-Host do servidor escolhido. O cliente então coloca esse valor em Destination-Host em todas as solicitações posteriores (CCR-Update, CCR-Terminate). Isso

prende essas solicitações automaticamente a esse servidor. Em Gx/Gy, o **AVP CC-Request-Type (416)** distingue o tipo de solicitação.


Filtro 1
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platfor
Não
Filtro 2
Passou?
Sim
Não
Filtro 3
Sim
Passou?
Sim
Não
Regra Corresponde\nAplicar
Regra Falha

Essa afinidade se aplica a listas de pares de prioridade/peso. Listas de nomes de host simples seguem o algoritmo global. Esse comportamento não os afeta.

Configuração

Configure o Roteamento Avançado sob `module_advanced_routing` em `config/runtime.exs`.

```
module_advanced_routing: %{
  # Habilitar ou desabilitar o módulo
  enabled: false,

  # As regras são avaliadas em ordem; a primeira correspondência
vence
  rules: [
    %{
      rule_name: "route_s6a_air_to_hss",
      match: :all,
      filters: [
        {:application_id, 16_777_251},
        {:command_code, 318}
      ],
      route: %{
        peers: ["hss01.example.com"]
      }
    }
  ]
}
```

Parâmetros do Módulo

Parâmetro	Tipo	Requerido	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Interruptor mestre para o módulo. Quando `false`, o DRA não avalia regras e todo o tráfego usa roteamento padrão.
`rules`	Lista	Sim	`[]`	Lista ordenada de regras de roteamento. O DRA as avalia de cima para baixo. A primeira correspondência vence. Veja Parâmetros da Regra.

Parâmetros da Regra

Parâmetro	Tipo	Requerido	Padrão	Descrição
`rule_name`	String	Sim	-	Um nome descritivo e único para a regra. O DRA o mostra em logs e na interface de gerenciamento.
`match`	Átomo	Sim	-	Como os filtros se combinam: `:all` (AND lógico), `:any` (OR lógico) ou `:none` (NOR lógico). Veja Modos de Correspondência.
`filters`	Lista	Sim	-	As condições que uma mensagem deve atender. Veja Filtros.
`route`	Mapa ou Átomo	Sim	-	Para onde o DRA envia tráfego correspondente. Isso é uma lista de pares ou um destino especial. Veja Destinos de Roteamento.

Filtros

Um filtro é uma tupla de `{filter_type, value}`. O Roteamento Avançado suporta esses filtros:

Filtro	Exemplo	Descrição
`:via_peer`	`{:via_peer, "mme01.example.com"}`	O Origin-Host do par que enviou a solicitação (o salto mais recente).
`:application_id`	`{:application_id, 16_777_251}`	O Application-Id Diameter. Veja IDs de Aplicação.
`:command_code`	`{:command_code, 318}`	O Command-Code Diameter.
`:avp`	`{:avp, {1, "999999000000001"}}`	Um valor AVP, escrito como `{avp_code, value}`.

Valores de Filtro

- **Valores de lista (OR).** Qualquer valor de filtro pode ser uma lista. O DRA trata a lista como um OR lógico sobre seus membros. Por exemplo, `{:command_code, [317, 318]}` corresponde ao código de comando 317 ou 318. `{:via_peer, ["dra01.example.com", "dra02.example.com"]}` corresponde a qualquer par.
- **Valores Regex.** Valores de filtro AVP e membros de lista suportam expressões regulares através de um sigilo regex, por exemplo `{:avp, {1, ~r"999001.*"}}`. Você pode misturar membros, por exemplo `{:avp, {1, ["505571234567", ~r"999001.*", ~r"505057.*"]}}`.
- **Verificação de presença (`:any`).** Um valor de `:any` corresponde quando o atributo está presente com qualquer valor. Por exemplo, `{:avp, {264, :any}}` corresponde sempre que o AVP Origin-Host existir. `{:via_peer, :any}` corresponde a qualquer par de origem.

Limitações de Filtros AVP

Use filtros AVP para AVPs **simples** apenas, como User-Name, Origin-Host e Destination-Realm. **AVPs agrupados não são suportados** e não

correspondem. Valores binários complexos não são suportados. Sempre use a forma `{:avp, {code, value}}`.

Destinos de Roteamento

O parâmetro `route` aceita um mapa de lista de pares ou um dos átomos/tuplas de destino especiais abaixo.

Destino	Exemplo	Descrição
Lista de pares	`%{peers: [...]}`	Roteie para um dos pares listados. As entradas podem ser nomes de host simples ou mapas de prioridade/peso. Veja Seleção de Pares.
Destination-Host	`:destination_host`	Roteie para o par nomeado no **AVP Destination-Host (293)** da mensagem. Se o AVP estiver ausente, a regra não se aplica e o DRA usa roteamento padrão.
Descartar	`:drop`	Descartar a mensagem sem resposta ao par solicitante. Use isso para bloquear tráfego indesejado, como assinantes de teste ou intervalos indesejados.
Erro	`{:error, 3004}`	Responder com uma resposta de erro Diameter que carrega o Result-Code dado. O DRA preenche automaticamente Origin-Host, Origin-Realm e Session-Id. O DRA não encaminha a mensagem para nenhum par.

Para uma aplicação **de reencaminhamento**, apenas códigos de resultado 3xxx (protocolo) são estritamente válidos nas respostas geradas (`3002`, `3003`, `3004`, `3007`). Algumas implantações também usam `5012` (DIAMETER_UNABLE_TO_COMPLY) para rejeitar comandos específicos. Veja as tabelas de referência para os significados.

Parâmetros de Pares (forma prioridade/peso)

Parâmetro	Tipo	Requerido	Padrão	Descrição
`host`	String	Sim	-	O Origin-Host do par. Deve corresponder exatamente à identidade Diameter configurada do par e é sensível a maiúsculas e minúsculas.
`priority`	Inteiro	Não	`100`	O nível. O DRA prefere números mais baixos. Ele usa níveis numerados mais altos apenas quando todos os pares conectados em um nível inferior estão indisponíveis.
`weight`	Inteiro	Não	`100`	A parte do tráfego dentro de um nível. O DRA divide o tráfego entre os pares de um nível em proporção aos seus pesos. Um peso de `0` exclui um par, a menos que todos os pares no nível sejam `0`.

Exemplos de Configuração

Failover para um Par de Backup

Roteie Solicitações de Informação de Autenticação S6a para um HSS primário. Recaia para um HSS de recuperação de desastres apenas quando o primário estiver fora.

```
%{
  rule_name: "s6a_air_failover",
  match: :all,
  filters: [
    {:application_id, 16_777_251},
    {:command_code, 318}
  ],
  route: %{
    peers: [
      %{host: "hss01.example.com", priority: 10},
      %{host: "hss-dr.example.com", priority: 20}
    ]
  }
}
```

Como funciona: Enquanto `hss01` estiver conectado, cada solicitação vai para ele. Se `hss01` desconectar, as solicitações falham para `hss-dr`. Se ambos estiverem fora, o DRA responde às solicitações com 3002.

Balanceamento de Carga Uniforme

Espalhe o tráfego Gx uniformemente entre dois PCRFs.

```
%{
  rule_name: "gx_load_balance",
  match: :all,
  filters: [
    {:application_id, 16_777_238}
  ],
  route: %{
    peers: [
      %{host: "pcrf01.example.com", priority: 10, weight: 50},
      %{host: "pcrf02.example.com", priority: 10, weight: 50}
    ]
  }
}
```

Como funciona: O DRA divide novas sessões (CCR-Initial) aproximadamente 50/50 entre os dois PCRFs. O DRA prende mensagens durante a sessão ao PCRF que lidou com a sessão. Ele usa o Destination-Host que as mensagens carregam.

Balanceamento de Carga Ponderado com Backup

Envie mais tráfego para um PCRF maior e menos para um menor. Mantenha um backup para quando ambos estiverem indisponíveis.

```
%{
  rule_name: "gx_weighted_with_backup",
  match: :all,
  filters: [
    {:application_id, 16_777_238}
  ],
  route: %{
    peers: [
      %{host: "pcrf01.example.com", priority: 10, weight: 70},
      %{host: "pcrf02.example.com", priority: 10, weight: 30},
      %{host: "pcrf-dr.example.com", priority: 20, weight: 100}
    ]
  }
}
```

Como funciona: Enquanto qualquer PCRF de prioridade 10 estiver conectado, o DRA divide novas sessões 70/30 entre eles. O tráfego se move para `pcrf-dr` apenas quando **ambos** os PCRFs de prioridade 10 estiverem fora. Sessões estabelecidas permanecem presas ao seu servidor durante sua vida útil.

Roteamento por Par de Origem

Roteie tráfego S6a que chega de DRAs upstream específicos para um par local de HSS. Isso é útil durante uma migração ou transição.

```
%{
  rule_name: "s6a_via_peer_to_local_hss",
  match: :all,
  filters: [
    {:application_id, 16_777_251},
    {:via_peer, ["dra01.example.com", "dra02.example.com"]},
    {:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}
  ],
  route: %{
    peers: ["hss01.example.com", "hss02.example.com"]
  }
}
```

Como funciona: O DRA roteia para os nós HSS locais apenas solicitações S6a que chegaram de `dra01` ou `dra02` e carregam o Origin-Realm correspondente (AVP 296). Tudo o mais cai em regras posteriores ou no roteamento padrão.

Roaming de Entrada com Correspondência de Padrão por IMSI

Roteie tráfego de roaming de entrada com base em padrões de IMSI (User-Name, AVP 1).

```
%{
  rule_name: "inbound_s6a_roaming",
  match: :all,
  filters: [
    {:application_id, 16_777_251},
    {:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}},
    {:avp, {1, ["505571234567", ~r"999001.*"]}}
  ],
  route: %{
    peers: ["dra01.example.com", "dra02.example.com"]
  }
}
```

Como funciona: A regra corresponde a mensagens S6a do Origin-Realm dado cujo User-Name é exatamente `505571234567` ou começa com `999001`. O DRA roteia essas mensagens para os pares designados. O valor da lista aplica lógica OR entre a string exata e a regex.

Honrar o AVP Destination-Host

Roteie para qualquer par que o remetente nomeou, apenas para tráfego correspondente.

```
%{
  rule_name: "honour_destination_host",
  match: :all,
  filters: [
    {:avp, {1, [~r"90199.*"]}}
  ],
  route: :destination_host
}
```

Como funciona: Quando o IMSI corresponde, o DRA envia a solicitação para o par nomeado em seu AVP Destination-Host (293). Se o AVP Destination-Host estiver ausente, o DRA trata a correspondência como uma falha e aplica o roteamento padrão.

Descartar Tráfego Indesejado

Descartar silenciosamente tráfego de faixa de teste.

```
%{
  rule_name: "drop_test_subscribers",
  match: :all,
  filters: [
    {:application_id, 16_777_251},
    {:avp, {1, [~r"999999.*"]}}
  ],
  route: :drop
}
```

Como funciona: O DRA descarta mensagens S6a cujo IMSI começa com `999999`. Ele não envia resposta. Use isso para filtrar tráfego de teste ou bloquear intervalos de assinantes específicos.

Proteção contra Sobrecarga com uma Resposta de Erro

Retornar DIAMETER_TOO_BUSY para um par específico sobrecarregado.

```
%{
  rule_name: "rate_limit_high_volume_peer",
  match: :all,
  filters: [
    {:via_peer, "mme-overloaded-01.example.com"},
    {:application_id, 16_777_251}
  ],
  route: {:error, 3004}
}
```

Como funciona: O DRA responde ao tráfego S6a do par nomeado com **DIAMETER_TOO_BUSY (3004)**. Isso sinaliza ao par para recuar. O DRA não encaminha nada para cima.

Bloquear um Comando Específico

Rejeitar um tipo de comando particular com um erro explicativo em vez de um descarte silencioso.

```
%{
  rule_name: "block_purge_requests",
  match: :all,
  filters: [
    {:application_id, 16_777_251},
    {:command_code, 321}
  ],
  route: {:error, 5012}
}
```

Como funciona: O DRA responde a Solicitações de Purga de UE S6a (comando 321) com **DIAMETER_UNABLE_TO_COMPLY (5012)**. Isso desativa essa operação e deixa outros comandos intocados.

Lendo a Configuração Ativa

Você pode ler as regras ativas pela API de gerenciamento em `GET /api/rules`. A resposta inclui as regras de roteamento, as regras de transformação e o `routing_algorithm` global.

Para cada regra que roteia para uma lista de pares, a API relata como a seleção se comporta:

- `route.selection` é `"priority_weight"` quando a regra usa mapas de prioridade/peso. Caso contrário, é o algoritmo global (`"failover"` ou `"random"`).
- No modo `priority_weight`, a API retorna cada par com sua `priority` e `weight` efetivos. Ela preenche os padrões para que a distribuição seja explícita.

```
{
  "status": "success",
  "response": {
    "routing_algorithm": "random",
    "routing_rules": [
      {
        "rule_name": "gx_weighted_with_backup",
        "match": "all",
        "route": {
          "selection": "priority_weight",
          "peers": [
            { "host": "pcrf01.example.com", "priority": 10,
"weight": 70 },
            { "host": "pcrf02.example.com", "priority": 10,
"weight": 30 },
            { "host": "pcrf-dr.example.com", "priority": 20,
"weight": 100 }
          ]
        }
      }
    ],
    "transform_rules": []
  }
}
```

Melhores Práticas

1. **Especificidade primeiro.** Ordene as regras da mais específica para a mais geral. A primeira correspondência vence.
2. **Correspondências comuns cedo.** Coloque as correspondências de maior volume perto do topo. Isso reduz a sobrecarga de avaliação.
3. **Valide regex.** Teste padrões de expressões regulares contra mensagens capturadas antes da implantação.
4. **Nomeie claramente.** Use valores descritivos para `rule_name`. Logs e a interface de gerenciamento então se explicam.
5. **Prefira prioridade/peso para aplicativos com estado.** Use mapas de pares de prioridade/peso para Gx/Gy. O DRA então aplica a afinidade de sessão automaticamente.

Resolução de Problemas

Sessões Rejeitadas com UNKNOWN_SESSION_ID

Sintomas: Sessões Gx ou Gy estabelecem corretamente, mas o servidor rejeita atualizações durante a sessão com Result-Code 5002.

Possíveis causas:

- Os clientes (PCEF/OCF) não definem Destination-Host em solicitações durante a sessão, então o DRA não pode prendê-las.
- Uma regra roteia esse tráfego com nomes de host simples e o algoritmo global `:random`. Esse algoritmo não aplica afinidade de sessão.

Resolução:

1. Confirme que as solicitações durante a sessão carregam um Destination-Host que nomeia o servidor que respondeu à solicitação inicial.
2. Converta a lista de pares da regra em mapas de prioridade/peso. O DRA então aplica a afinidade de sessão automaticamente.

Tráfego Correspondente Respondido com 3002

Sintomas: Uma regra deve roteirizar certas solicitações, mas o DRA responde a elas com DIAMETER_UNABLE_TO_DELIVER.

Possíveis causas:

- Nenhum dos pares listados na regra correspondente está atualmente conectado.
- Os nomes de host na regra não correspondem exatamente aos valores Origin-Host dos pares.

Resolução:

1. Verifique o estado de conexão dos pares na interface de gerenciamento ou nos logs.

2. Confirme que os valores `host` na regra correspondem exatamente aos Origin-Host dos pares. A correspondência é sensível a maiúsculas e minúsculas.

Uma Regra Não Está Tendo Efeito

Sintomas: O tráfego que deveria corresponder a uma regra passa pelo roteamento padrão em vez disso.

Possíveis causas:

- `enabled` está definido como `false`.
- Uma regra anterior na lista corresponde primeiro e vence a rota.
- Um valor de filtro não corresponde. Isso pode ser uma discrepância de maiúsculas e minúsculas em um valor AVP ou um AVP agrupado não suportado.

Resolução:

1. Confirme que `enabled` está `true`. Reinicie o DRA após a alteração.
2. Revise a ordem das regras. A primeira regra correspondente vence.
3. Confirme os valores dos filtros contra uma mensagem capturada. Lembre-se de que o DRA não pode corresponder a AVPs agrupados.

Referência

Códigos de Resultado Diameter

Código	Nome	Descrição	Referênci
3002	DIAMETER_UNABLE_TO_DELIVER	Nenhum par conectado está disponível para entregar a solicitação	RFC 6733 §7.1.3
3003	DIAMETER_REALM_NOT_SERVED	Nenhuma rota configurada atende ao realm da solicitação	RFC 6733 §7.1.3
3004	DIAMETER_TOO_BUSY	O nó está sobrecarregado e não pode processar a solicitação	RFC 6733 §7.1.3
5002	DIAMETER_UNKNOWN_SESSION_ID	O servidor não reconhece o Session-Id. Por exemplo, uma mensagem durante a sessão chegou ao servidor errado	RFC 6733 §7.1.5
5012	DIAMETER_UNABLE_TO_COMPLY	O DRA não pode honrar a	RFC 6733 §7.1.5

Código	Nome	Descrição	Referênci
		solicitação. Use isso para rejeitar comandos específicos	

AVPs Relevantes

AVP	Código	Descrição	Referência
Destination-Host	293	Nomeia o par específico que deve receber uma solicitação. Esta é a base para afinidade de sessão	RFC 6733 §6.5
Session-Id	263	Identifica uma sessão durante sua vida útil	RFC 6733 §8.8
CC-Request-Type	416	Distingue solicitações INITIAL, UPDATE e TERMINATION em Gx/Gy	RFC 4006 §8.3

IDs de Aplicação

ID	Interface	Descrição	Referência
16777238	Gx	Controle de política entre PCEF e PCRF	3GPP TS 29.212
16777251	S6a/S6d	Autenticação MME/SGSN e gerenciamento de assinatura para HSS	3GPP TS 29.272
4	Gy	Cobrança online entre OCF e OCS	RFC 4006

Documentação Relacionada

- Roteamento Padrão - Roteamento Destination-Host / Destination-Realm usado quando nenhuma regra corresponde.
- Transformação Avançada - Reescrita de conteúdo para solicitações e respostas, compartilhando o mesmo mecanismo de filtro.
- Referência de Configuração - Referência completa dos parâmetros `config/runtime.exs`, incluindo o `peer_selection_algorithm` global.

Transformação Avançada

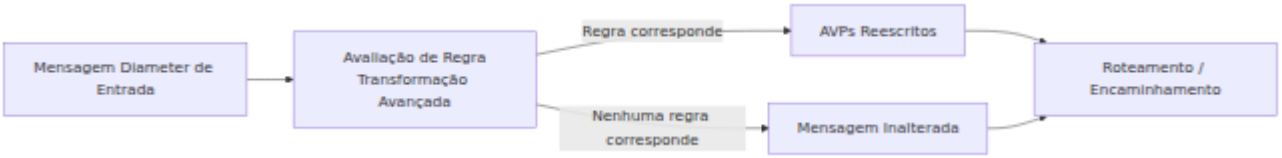
O módulo **Transformação Avançada** reescreve Pares de Atributo-Valor (**AVPs**) dentro de mensagens Diameter que correspondem a critérios definidos pelo operador. Use-o para corrigir, traduzir ou normalizar conteúdos de mensagens enquanto passam pelo OmniDRA. Você não precisa alterar os elementos de rede que enviam ou recebem.

Visão Geral

A Transformação Avançada avalia uma lista de **regras** contra cada mensagem Diameter. Uma regra combina um conjunto de **filtros** (os critérios de correspondência) com uma **transformação** (a mudança a ser aplicada). Quando uma regra corresponde, o DRA aplica sua transformação à mensagem e para o processamento de regras.

A Transformação Avançada fornece três ações de transformação. `:edit` altera o valor de um AVP existente. `:remove` exclui um ou mais AVPs por código. `:overwrite` reescreve um ou mais AVPs por nome, incluindo AVPs agrupados. Veja Ações de Transformação.

A Transformaç►►o Avançada compartilha seu modelo de processamento de regras com o mecanismo de roteamento. Veja Roteamento Avançado para entender como o DRA toma decisões de roteamento.

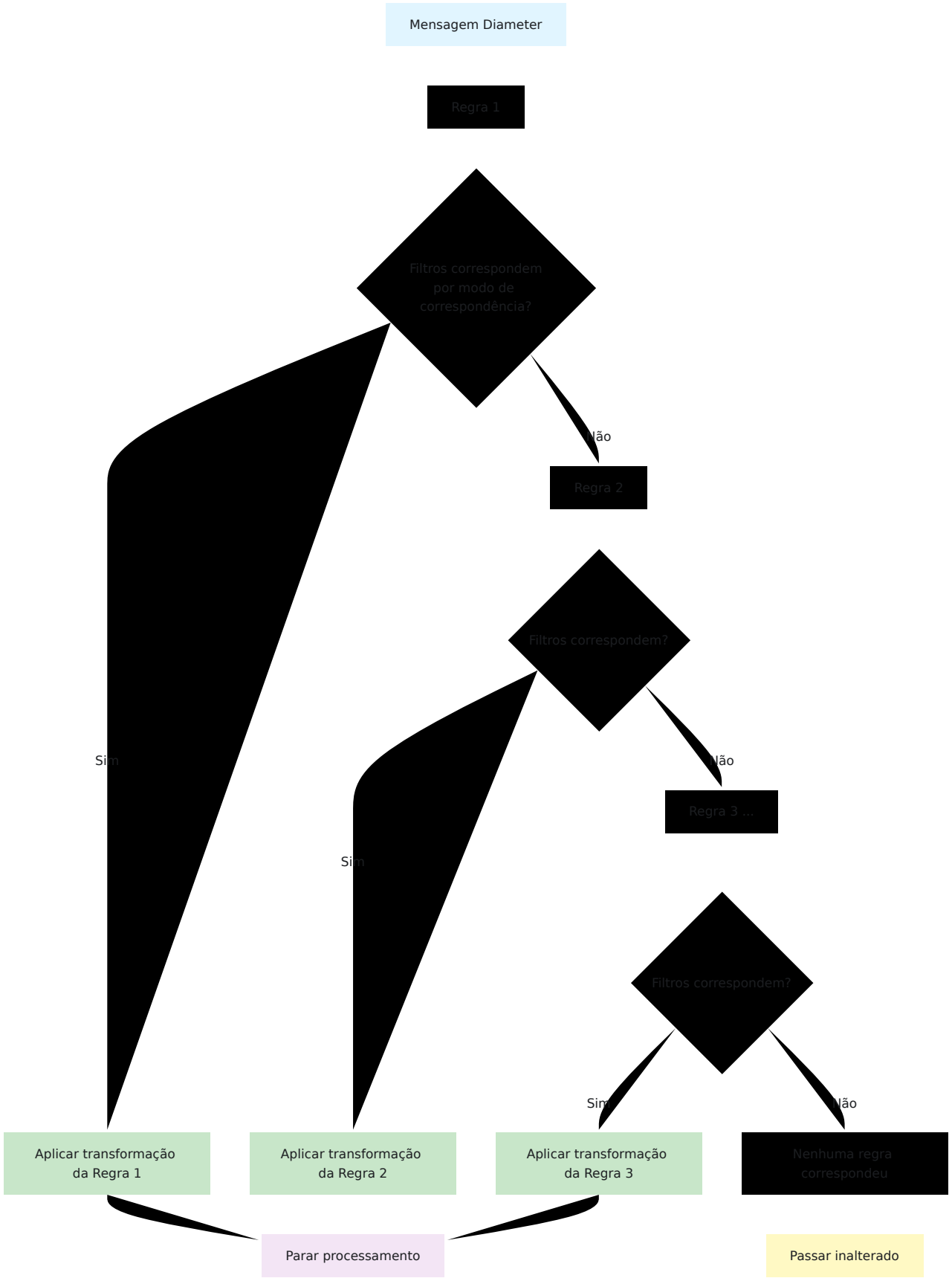


Como as Regras São Processadas

O DRA avalia as regras **em ordem, de cima para baixo**, conforme aparecem na configuração. A **primeira regra correspondente vence**. Assim que os filtros de uma regra correspondem, o DRA aplica sua transformação e não considera mais regras. Se nenhuma regra corresponder, a mensagem passa **transparente** sem modificação.

Dentro de uma única regra, o parâmetro `match` controla como os filtros individuais se combinam:

- `:all` - **E** lógico. Cada filtro deve passar para que a regra corresponda.
- `:any` - **OU** lógico. Pelo menos um filtro deve passar para que a regra corresponda.
- `:none` - **NOR** lógico. Nenhum filtro pode passar para que a regra corresponda (correspondência inversa).

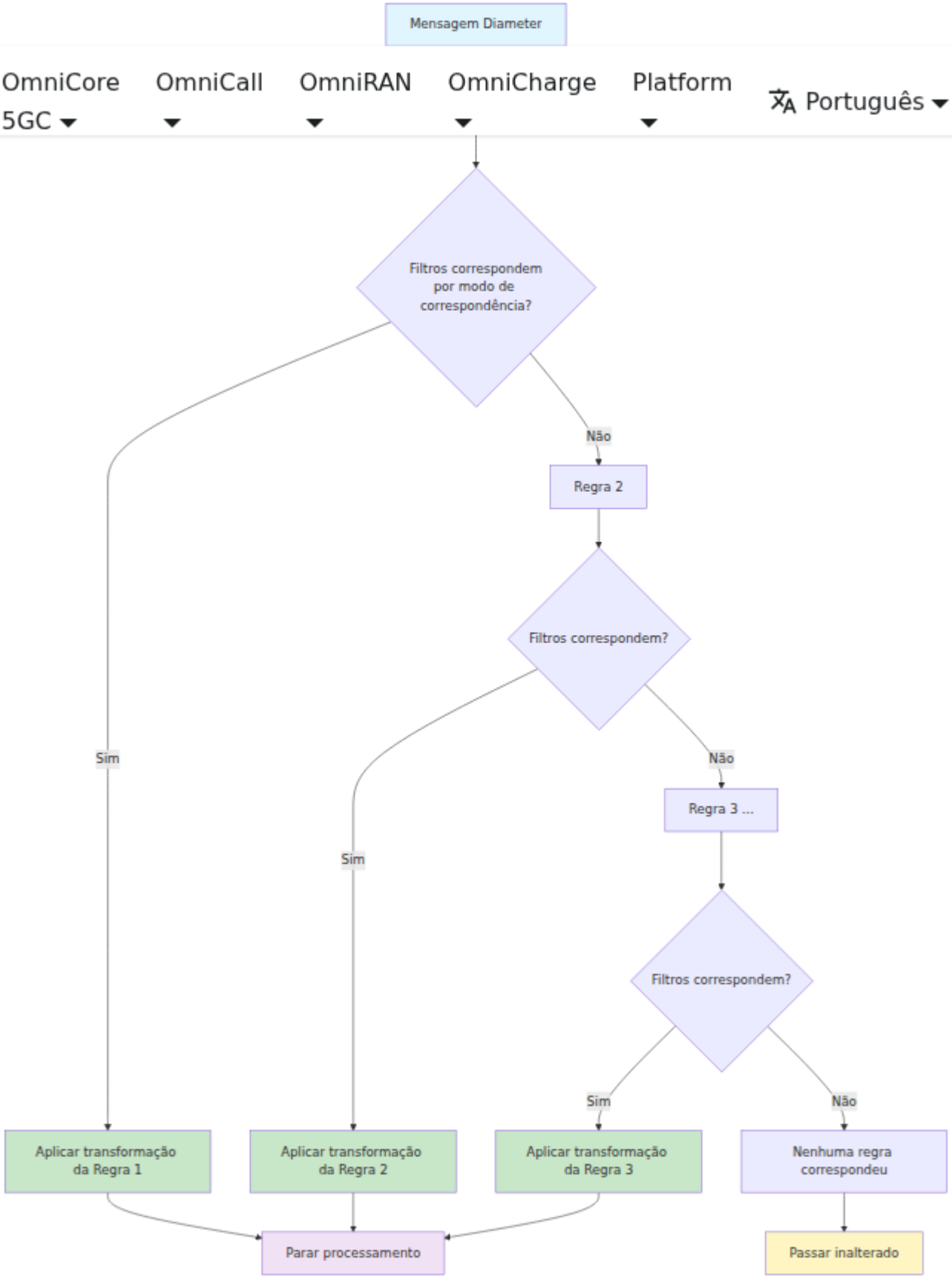


As regras de transformação operam sobre o conteúdo da mensagem. Elas não selecionam um destino. Uma regra pode reescrever um AVP, por exemplo, um

realm ou identidade. O mecanismo de roteamento então usa esse AVP para encaminhar a mensagem.

Avaliação do Modo de Correspondência

Os três modos de correspondência avaliam a mesma lista de filtros de maneira diferente:


Mensagem Diameter
Filtros correspondem por modo de correspondência?
Sim
Não
Regra 2
Filtros correspondem?
Sim
Não
Regra 3 ...
Filtros correspondem?
Sim
Não
Aplicar transformação da Regra 1
Aplicar transformação da Regra 2
Aplicar transformação da Regra 3
Nenhuma regra correspondeu
Parar processamento
Passar inalterado

Filtros

Filtros são as condições que decidem se uma regra se aplica a uma mensagem. A Transformação Avançada fornece seis tipos de filtros.

Filtro	Formato	Descrição
`:packet_type`	`{:packet_type, :request}`	Corresponde à **direção** da mensagem. Aceita apenas `:request` ou `:answer`; uma lista de valores **não** é aceita. Veja Filtragem por Tipo de Pacote.
`:application_id`	`{:application_id, <id>}`	Corresponde ao Application-Id Diameter da mensagem (por exemplo, `16777251` para S6a/S6d). Veja IDs de Aplicação.
`:command_code`	`{:command_code, <code>}`	Corresponde ao Código de Comando Diameter da mensagem (por exemplo, `318` para AIR). Veja Códigos de Comando.
`:avp`	`{:avp, {<code>, <value>}}`	Corresponde a um AVP pelo seu código numérico e valor esperado. Veja Filtragem de AVP.
`:to_peer`	`{:to_peer, "hss01.example.com"}`	Corresponde ao **peer de destino resolvido** de uma solicitação. Corresponde apenas a pacotes de **solicitação** e nunca corresponde a respostas. Aceita uma lista de nomes de host (OU). Veja Filtragem de Peer.

Filtro	Formato	Descrição
`:from_peer`	`{:from_peer, "hss01.example.com"}`	Corresponde ao **peer que respondeu** a uma mensagem. Corresponde apenas a pacotes de **resposta** e nunca corresponde a solicitações. Aceita uma lista de nomes de host (OU). Veja Filtragem de Peer.

Filtragem por Tipo de Pacote

O filtro `:packet_type` restringe uma regra a uma direção de tráfego. Aceita exatamente um dos dois valores:

- **`:request`** - corresponde apenas a mensagens Diameter de **solicitação**.
- **`:answer`** - corresponde apenas a mensagens Diameter de **resposta**.

Diferente de `:application_id`, `:command_code` e `:avp`, o filtro `:packet_type` **não** aceita uma lista de valores.

Filtragem de AVP

O filtro `:avp` tem a estrutura `{avp_id, avp_value}`. O `avp_id` é o código numérico do AVP. O `avp_value` é o valor a ser comparado, expresso como uma string.

Use o filtro `:avp` para **AVPs simples** como `User-Name` e `Origin-Host`. **AVPs agrupados não são suportados** e não correspondem. Valores binários complexos não correspondem.

O filtro corresponde a valores de várias maneiras:

- **String exata** - `{:avp, {1, "999999000000001"}}` corresponde apenas a esse valor exato.

- **Lista de valores** - `{:avp, {1, ["50557...", "505057..."]}}` corresponde se o AVP for igual a **qualquer** valor na lista. Esta é a lógica OR dentro da lista.
- **Expressão regular** - `{:avp, {1, [~r"9999999.*"]}}` corresponde a valores contra um padrão regex.

Use a sintaxe `~r"pattern"` para correspondência de expressões regulares:

Padrão	Corresponde
`~r"999001.*"`	Valores de `User-Name`/IMSI que começam com `999001`
`~r"^310[0-9]{3}.*"`	Valores com prefixos MCC/MNC específicos
`~r".*test$"`	Valores que terminam com `test`

Filtragem de Peer

Os filtros `:to_peer` e `:from_peer` correspondem a um **nome de host de peer**, não ao conteúdo da mensagem. Eles são específicos de direção e complementares:

- **`:to_peer`** corresponde ao **peer de destino resolvido** para o qual o DRA está prestes a encaminhar uma solicitação. Corresponde apenas a pacotes de **solicitação**. Nunca corresponde a uma resposta.
- **`:from_peer`** corresponde ao **peer que respondeu** a uma mensagem. Corresponde apenas a pacotes de **resposta**. Nunca corresponde a uma solicitação.

Ambos os filtros aceitam um único nome de host ou uma **lista de nomes de host**. Com uma lista, o filtro corresponde se o peer for igual a **qualquer** nome de host na lista. Esta é a lógica OR.

Filtro	Exemplo	Corresponde
`:to_peer`	`{:to_peer, "hss01.example.com"}`	Solicitações roteadas para o peer `hss01.example.com`
`:to_peer`	`{:to_peer, ["hss01.example.com", "hss02.example.com"]}`	Solicitações roteadas para **qualquer** peer listado
`:from_peer`	`{:from_peer, "hss01.example.com"}`	Respostas recebidas do peer `hss01.example.com`

Ações de Transformação

O mapa `transform` descreve a transformação. Sua chave `action` seleciona **uma das três** transformações. A **direção** da mensagem determina quais ações são permitidas:

Ação	O que faz	Solicitações	Respostas
`:edit`	Reescreve o **valor** de um AVP existente.	Sim	Não
`:remove`	Exclui o(s) AVP(s) nomeado(s) da mensagem.	Sim	Sim
`:overwrite`	Reescreve AVP(s) por nome, incluindo AVPs **agrupados**, usando um dicionário de decodificação.	Sim	Sim

A Ação `:edit`

```
transform: %{
  action: :edit,
  avps: [{:avp, {1, "999999000000002"}}]
}
```

`:edit` se comporta da seguinte forma:

- Ele **modifica o valor** do AVP nomeado na mensagem Diameter para o valor fornecido.
- Afeta apenas AVPs que **já existem** na mensagem. Se o AVP alvo não estiver presente, o DRA não faz **nenhuma alteração** e passa a mensagem como recebida.
- Cada entrada em `avps` é `{:avp, {<code>, <new_value>}}`. Isso define aquele AVP para o novo valor.
- `:edit` se aplica a **solicitações apenas**. Para reescrever AVPs em respostas, ou para alcançar AVPs agrupados, use `:overwrite`.

A Ação `:remove`

```
transform: %{
  action: :remove,
  avps: [{:avp, {264, :any}}]
}
```

`:remove` se comporta da seguinte forma:

- Ele **exclui** o(s) AVP(s) nomeado(s) em `avps` da mensagem.
- Ele corresponde apenas pelo **código do AVP**. Ignora o valor em cada entrada de `avps` para remoção. Use `:any` como valor para deixar isso explícito.
- Aplica-se tanto a **solicitações quanto a respostas**.

A Ação `:overwrite`

```
transform: %{
  action: :overwrite,
  dictionary: :diameter_gen_3gpp_s6a,
  avps: [{:avp, {1, "999999000000002"}}]
}
```

`:overwrite` se comporta da seguinte forma:

- Ele **reescreve** o(s) AVP(s) nomeado(s) por nome. Usa o `dictionary` fornecido para decodificar e re-encodar a mensagem.
- Ele decodifica a mensagem completa, então `:overwrite` pode alcançar **AVPs agrupados / de registro** que `:edit` não pode.
- Requer uma chave `dictionary`. Esta chave nomeia o dicionário Diameter que descreve a mensagem (por exemplo, `:diameter_gen_3gpp_s6a`).
- Aplica-se tanto a **solicitações quanto a respostas**.

Parâmetros do Mapa de Transformação

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`action`	Atom	Sim	-	A transformação a ser aplicada: `:edit` (reescrever um valor de AVP existente - **apenas solicitações**), `:remove` (excluir AVP(s) por código - solicitações e respostas), ou `:overwrite` (reescrever AVP(s) por nome, incluindo AVPs agrupados - solicitações e respostas).
`avps`	Lista	Sim	-	Lista de alvos de AVP. Para `:edit`/`:overwrite`, cada entrada é `{:avp, {<code>, <new_value>}}` e define aquele AVP para `<new_value>`. Para `:remove`, o DRA usa apenas `<code>` e ignora valor (use `:any`).
`dictionary`	Atom	Apenas para `:overwrite`	`:not_used`	O dicionário Diameter qu decodifica e re-encoda a mensagem para que o DRA possa sobrescrever AVPs nomeados (agrupados), por exemplo, `:diameter_gen_3gpp_s6a` As ações `:edit` e

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
				`:remove` ignoram esta chave.

Configuração

Configure a Transformação Avançada sob `module_advanced_transform` em `config/runtime.exs`. O módulo está desativado por padrão. Ele não faz nada até que você defina `enabled` como `true`.

```
module_advanced_transform: %{
  enabled: true,
  rules: [
    %{
      rule_name: "transform_user_name",
      match: :all,
      filters: [
        {:application_id, 16_777_251},
        {:command_code, 318},
        {:avp, {1, "999999000000001"}}
      ],
      transform: %{
        action: :edit,
        avps: [{:avp, {1, "999999000000002"}}]
      }
    }
  ]
}
```

Parâmetros do Módulo

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`enabled`	Boolean	Não	`false`	Habilita o módulo. Quando `false`, o DRA não avalia regras e todas as mensagens passam inalteradas.
`rules`	Lista	Sim (quando habilitado)	-	Lista ordenada de regras de transformação. O DRA as avalia de cima para baixo. A primeira regra correspondente vence. Veja Parâmetros da Regra.

Parâmetros da Regra

Cada entrada em `rules` é um mapa com as seguintes chaves.

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`rule_name`	String	Sim	-	Identificador único e descritivo para a regra. Suporta clareza operacional e monitoramento.
`match`	Atom	Sim	-	Como os filtros se combinam: `:all` (E), `:any` (OU), ou `:none` (NOR). Veja Como as Regras São Processadas.
`filters`	Lista	Sim	-	Lista de condições de filtro que uma mensagem deve satisfazer de acordo com o modo `match`. Veja Filtros.
`transform`	Mapa	Sim	-	A mudança a ser aplicada quando a regra corresponder. Veja Parâmetros do Mapa de Transformação.

Exemplos

Exemplo 1: Reescrever um User-Name em Solicitações AIR S6a

```
module_advanced_transform: %{
  enabled: true,
  rules: [
    %{
      rule_name: "transform_user_name",
      match: :all,
      filters: [
        {:application_id, 16_777_251},
        {:command_code, 318},
        {:avp, {1, "999999000000001"}}
      ],
      transform: %{
        action: :edit,
        avps: [{:avp, {1, "999999000000002"}}]
      }
    }
  ]
}
```

Como funciona: Com `match: :all`, cada filtro deve passar. A regra corresponde apenas a mensagens de Solicitação de Informação de Autenticação (`318`) do S6a (`16777251`) cuja `User-Name` (AVP `1`) é igual a `999999000000001`. Ao corresponder, a ação `:edit` reescreve `User-Name` para `999999000000002`. Se uma mensagem não carregar um AVP `User-Name`, o DRA não faz nenhuma alteração.

Caso de uso: Corrigir ou normalizar uma identidade de assinante específica enquanto transita pelo DRA. Por exemplo, mapear um IMSI legado para seu valor atual durante uma migração.

Exemplo 2: Reescrita de Realm com Base na Faixa de IMSI

```
module_advanced_transform: %{
  enabled: true,
  rules: [
    %{
      rule_name: "rewrite_destination_realm_roaming_partner",
      match: :all,
      filters: [
        {:avp, {296, "epc.mnc057.mcc505.3gppnetwork.org"}},
        {:avp, {1, [~r"50557.*"]}}
      ],
      transform: %{
        action: :edit,
        avps: [{:avp, {283, "epc.mnc030.mcc310.3gppnetwork.org"}}]
      }
    }
  ]
}
```

Como funciona: A regra corresponde quando o `Origin-Realm` (AVP `296`) é igual ao realm do parceiro de roaming **e** o `User-Name`/IMSI (AVP `1`) começa com `50557`. Em seguida, reescreve o `Destination-Realm` (AVP `283`). O mecanismo de roteamento então encaminha a mensagem para a rede parceira correta. A primeira regra correspondente vence, então coloque faixas de IMSI mais específicas acima de faixas mais amplas.

Caso de uso: Direcionar tráfego de assinantes de roaming ou MVNO para a rede central hospedada correta. O DRA seleciona a rede pela faixa de IMSI e pelo realm de origem.

Exemplo 3: Remover um AVP de Respostas de um Peer Específico

```
module_advanced_transform: %{
  enabled: true,
  rules: [
    %{
      rule_name: "strip_origin_host_from_partner_answers",
      match: :all,
      filters: [
        {:packet_type, :answer},
        {:from_peer, "hss01.partner.example.com"},
        {:command_code, 318}
      ],
      transform: %{
        action: :remove,
        avps: [{:avp, {264, :any}}]
      }
    }
  ]
}
```

Como funciona: O filtro `:packet_type` limita a regra a **respostas**. O filtro `:from_peer` limita-a a respostas de `hss01.partner.example.com`. O filtro `:command_code` limita-a a AIA (`318`). Quando todos os três correspondem, a ação `:remove` exclui o `Origin-Host` (AVP `264`) independentemente de seu valor. A remoção corresponde apenas ao código do AVP.

Caso de uso: Remover um AVP das respostas de uma rede parceira. Use isso quando o parceiro preencher o AVP incorretamente ou quando o AVP não deve ser enviado para baixo.

Exemplo 4: Sobrescrever um AVP Agrupado em Solicitações S6a

```
module_advanced_transform: %{
  enabled: true,
  rules: [
    %{
      rule_name: "overwrite_user_name_s6a",
      match: :all,
      filters: [
        {:packet_type, :request},
        {:application_id, 16_777_251},
        {:command_code, 318}
      ],
      transform: %{
        action: :overwrite,
        dictionary: :diameter_gen_3gpp_s6a,
        avps: [{:avp, {1, "999999000000002"}}]
      }
    }
  ]
}
```

Como funciona: A regra corresponde a solicitações AIR (`318`) do S6a (`16777251`). A ação `:overwrite` decodifica a mensagem com o dicionário `:diameter_gen_3gpp_s6a`, reescreve o(s) AVP(s) nomeado(s) e re-encoda a mensagem. Ela decodifica a mensagem completa, então `:overwrite` pode alcançar AVPs aninhados dentro de estruturas agrupadas/registro que `:edit` não pode tocar.

Caso de uso: Corrigir um AVP que reside dentro de um AVP agrupado, por exemplo, um campo dentro de um grupo de dados de assinatura ou grupo específico de fornecedor. Um simples `:edit` não corresponderia a tal AVP.

Tabelas de Referência

IDs de Aplicação

ID	Interface	Descrição	Referência
16777251	S6a/S6d	Autenticação e assinatura MME/SGSN ↔ HSS	3GPP TS 29.272

Códigos de Comando

Código	Comando	Interface	Referência
318	Solicitação/Resposta de Informação de Autenticação (AIR/AIA)	S6a/S6d	3GPP TS 29.272 §7.2.5
316	Solicitação/Resposta de Atualização de Localização (ULR/ULA)	S6a/S6d	3GPP TS 29.272 §7.2.3

Códigos Comuns de AVP

Código	AVP	Descrição	Referência
1	User-Name	Identidade do assinante (IMSI no S6a).	3GPP TS 29.272 / RFC 6733 §8.14
264	Origin-Host	Identidade Diameter do originador da mensagem.	RFC 6733 §6.3
283	Destination-Realm	O realm para o qual a mensagem é destinada. O DRA o usa para roteamento.	RFC 6733 §6.6
296	Origin-Realm	Realm do originador da mensagem.	RFC 6733 §6.4

Melhores Práticas

1. **Especificidade** - Ordene as regras da mais específica para a mais geral. A primeira regra correspondente vence.
2. **Desempenho** - Coloque as regras mais comumente correspondidas primeiro. Isso reduz a sobrecarga de avaliação.
3. **Teste** - Valide padrões de expressões regulares antes da implantação.
4. **Nomeação** - Use valores descritivos de `rule_name` para clareza operacional e monitoramento.
5. **Monitoramento** - Acompanhe as taxas de correspondência de regras. Isso confirma que as regras se comportam como esperado.

Documentação Relacionada

- Roteamento Avançado - Seleção de destino usando o mesmo modelo de processamento de regras.

Visão Geral da Arquitetura DRA

OmniDRA é um **Agente de Roteamento Diameter (DRA)**. Ele está no centro de uma rede de sinalização Diameter. Ele retransmite mensagens de solicitação e resposta entre clientes, como nós MME/SGSN, e serviços de back-end, como HSS, PCRF e OCS. OmniDRA é construído em Elixir/Erlang na biblioteca OTP `:diameter`. Ele opera como um **Relay Diameter** e encaminha mensagens entre pares. Ele não termina as aplicações que as mensagens transportam.

Índice

1. O que é o DRA
2. Contexto da Rede
3. Pipeline de Processamento de Solicitações
4. Registro de Pares
5. Retransmissão de Mensagens
6. Tabelas de Referência
7. Leitura Adicional

O que é o DRA

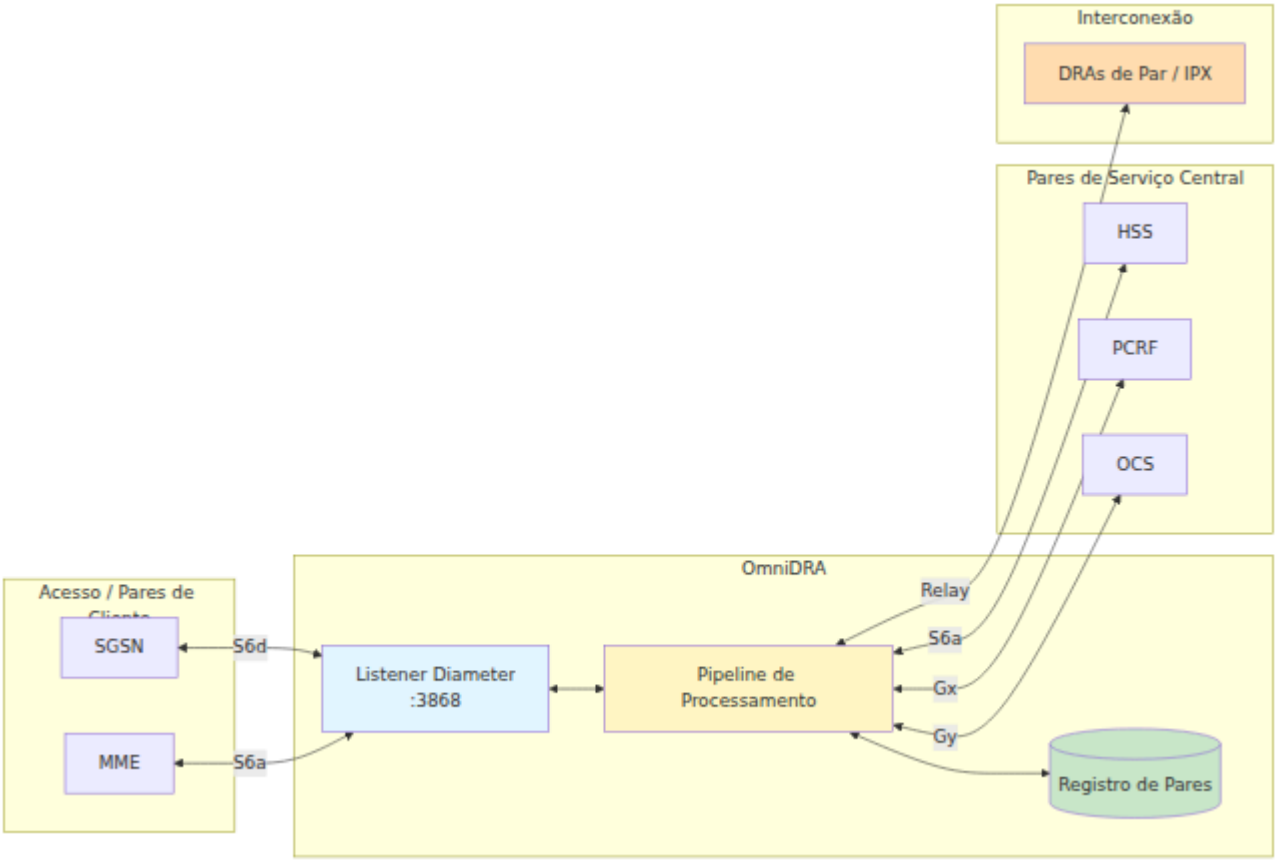
Um **Relay Diameter** recebe mensagens Diameter em um socket de escuta. Ele decide qual par conectado deve lidar com cada mensagem. Em seguida, ele encaminha a mensagem inalterada, exceto pelos AVPs de roteamento que o relay gerencia. OmniDRA opera como um relay, portanto, não implementa a lógica de aplicação das interfaces que transporta. Ele roteia **S6a**, **Gx**, **Gy** e qualquer outra aplicação Diameter de forma transparente. Os pares devem anunciar suporte para a aplicação.

Características principais:

- **Roteamento baseado em padrões** - O roteamento de solicitações segue o mecanismo de prioridade do Protocolo Base Diameter (RFC 6733). Ele seleciona um par com o **Destination-Host**, **Destination-Realm** e **Application-Id**.
- **Correlação de resposta com estado** - As respostas sempre retornam ao longo do caminho reverso de sua solicitação. O DRA correlaciona cada resposta pelo **Hop-by-Hop Identifier**.
- **Módulos de processamento opcionais** - Módulos de direcionamento de roaming, pesquisa de assinantes, roteamento avançado e transformação estendem o comportamento padrão. Eles não o substituem.
- **Pipeline de segurança** - Limitação de taxa, firewall Diameter, sanitização de AVP e ocultação de topologia protegem a rede na borda de sinalização.

Contexto da Rede

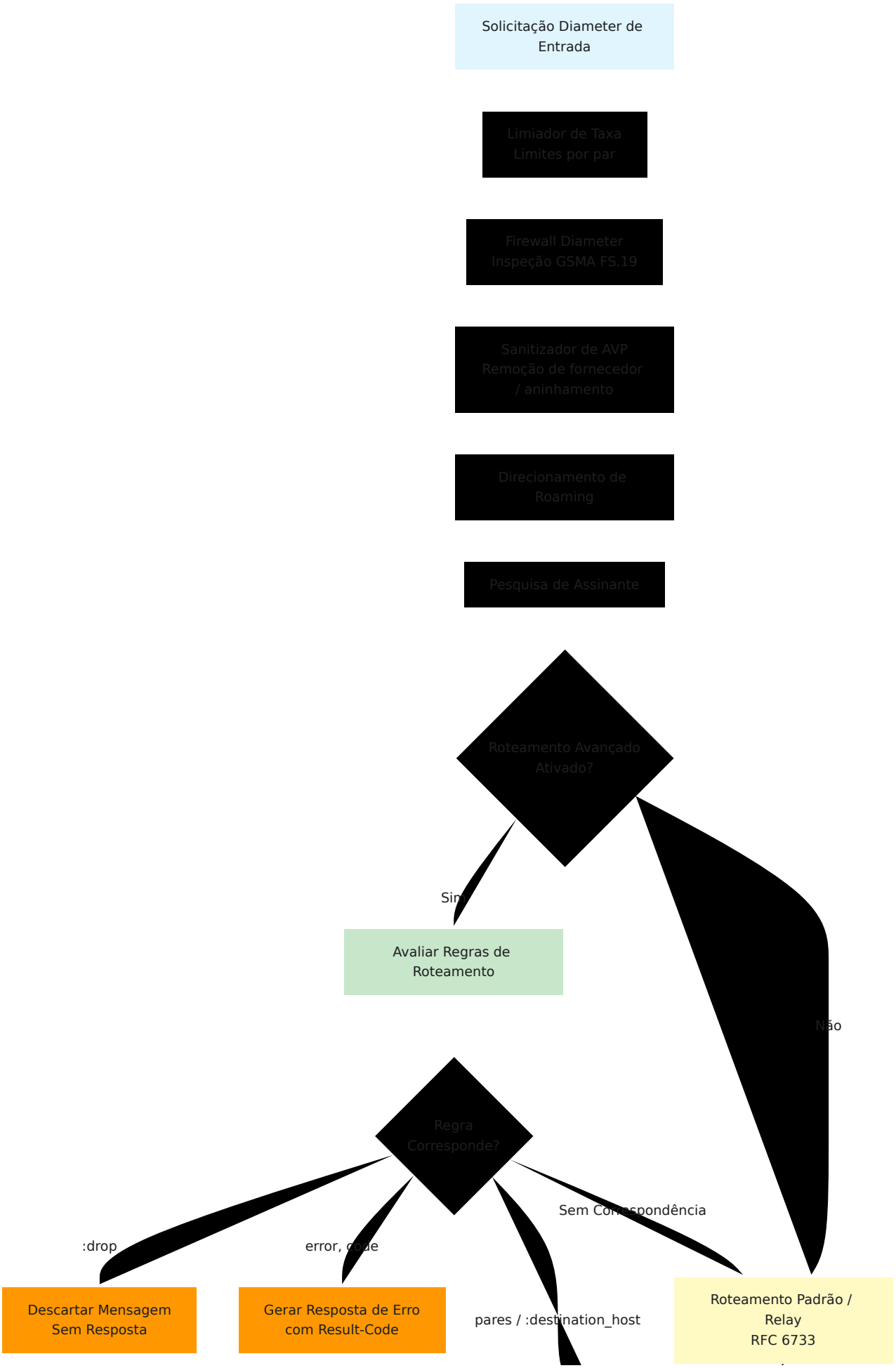
Implante o DRA entre os nós de sinalização do **lado de acesso** (como MME e SGSN) e os serviços do **lado central** (como HSS, PCRF e OCS). Cada par estabelece uma conexão Diameter com o DRA. O DRA nunca origina tráfego de aplicação próprio.



O DRA escuta por conexões Diameter de entrada na porta TCP/SCTP `3868`. Esta é a porta padrão definida pela RFC 6733. Cada par anuncia os domínios e aplicações que suporta durante a **Troca de Capacidades** (CER/CEA). O DRA registra isso em seu registro de pares para uso em decisões de roteamento.

Pipeline de Processamento de Solicitações

Cada solicitação de entrada passa por um pipeline ordenado. As etapas de segurança e direcionamento são executadas primeiro. As etapas de roteamento são executadas em seguida. As respostas ignoram este pipeline. Veja Retransmissão de Mensagens.


Solicitação Diameter de Entrada
Limiador de Taxa Limites por par
Firewall Diameter Inspeção GSMA FS.19
Sanitizador de AVP Remoção de fornecedor / aninhamento
Direcionamento de Roaming
Pesquisa de Assinante
Roteamento Avançado Ativado?
Sim
Avaliar Regras de Roteamento
Não
Regra Corresponde?
:drop
error, code
Sem Correspondência
pares / :destination_host
Descartar Mensagem Sem Resposta
Gerar Resposta de Erro com Result-Code
Roteamento Padrão / Relay RFC 6733

Estágios do Pipeline

Estágio	Propósito
Limiador de Taxa	Impõe limites de taxa de solicitação por par para proteger serviços downstream de sobrecarga.
Firewall Diameter	Inspeciona solicitações contra verificações de camada baixa e Categoria 1/2/3 do GSMA FS.19. Veja Segurança.
Sanitizador de AVP	Remove AVPs de fornecedor não permitidos e impõe limites de profundidade de aninhamento de AVP.
Direcionamento de Roaming	Aplica direcionamento definido pelo operador para influenciar qual rede um assinante em roaming é direcionado.
Pesquisa de Assinante	Resolve o contexto de roteamento específico do assinante antes da avaliação de regras.
Roteamento Avançado	Avalia o roteamento baseado em regras que pode descartar, gerar erro ou direcionar uma solicitação para pares específicos. Veja Roteamento Avançado.

Estágio	Propósito
Roteamento Padrão / Relay	Retorna ao roteamento de prioridade RFC 6733 quando nenhuma regra avançada se aplica. Veja Roteamento Padrão.
Rx Session Binding	Redireciona uma solicitação Rx para o PCRF que mantém sua sessão. Este é o único estágio que é executado *após* o Roteamento Avançado. Uma solicitação `Session-Termination-Request` Rx não carrega nada mais para roteamento. Veja Rx Session Binding.
Ocultação de Topologia	Reescreve ou remove AVPs identificadores (Route-Record, Origin-Host/Realm) em mensagens de saída. Veja Segurança.

Quando o Roteamento Avançado produz um resultado `:drop`, o DRA descarta a solicitação sem resposta. Um resultado `error` gera uma resposta Diameter que carrega o **Result-Code** configurado. Caso contrário, o DRA encaminha solicitações correspondentes para os pares especificados ou para o par **Destination-Host**. Solicitações não correspondentes caem no roteamento padrão.

O Rx Session Binding é a única exceção ao Roteamento Avançado ter a palavra final. Seu escopo é restrito. Ele apenas redireciona uma solicitação que já estava prestes a ser retransmitida, e apenas para Rx. Ele deixa um resultado `:drop`, um resultado `error` e um `Destination-Host` explícito inalterados.

Prioridade de Roteamento Padrão

Quando uma solicitação chega ao roteamento padrão, o DRA seleciona um par usando o mecanismo de prioridade da RFC 6733 §6.1:

1. **Destination-Host AVP (293)** - Se a solicitação contém este AVP, o DRA roteia diretamente para o par nomeado. Este é o mecanismo de maior prioridade. Se esse par não estiver conectado, o roteamento falha.

2. **Destination-Realm AVP (283)** - Se o Destination-Host estiver ausente, o DRA seleciona um par conectado que anuncia o domínio de destino. O DRA faz balanceamento de carga entre os pares correspondentes.
3. **Application-Id** - O DRA filtra os pares candidatos para aqueles que anunciam suporte para a aplicação da mensagem. Os pares anunciam esse suporte durante a Troca de Capacidades.

O comportamento completo de roteamento padrão, incluindo o algoritmo de seleção de pares, está documentado em Roteamento Padrão.

Registro de Pares

O DRA mantém um **registro de pares** em memória. O registro rastreia cada conexão de par e as capacidades que o par anunciou durante a Troca de Capacidades. O registro é a fonte autoritativa para decisões de roteamento. Apenas pares no estado **conectado** são elegíveis para receber tráfego. O DRA exclui automaticamente pares desconectados ou inativos.

Para cada par conectado, o registro registra:

- A **Identidade Diameter** do par (Origin-Host / FQDN).
- Os **domínios** que o par anuncia.
- Os **Application-Ids** que o par suporta.
- O **estado de conexão** atual.

O DRA aprende o suporte a domínios e aplicações dinamicamente a partir de CER/CEA. Para adicionar capacidade a uma interface, conecte um par adicional que anuncie o domínio e a aplicação relevantes. O DRA começa a fazer balanceamento de carga para o novo par automaticamente.

Retransmissão de Mensagens

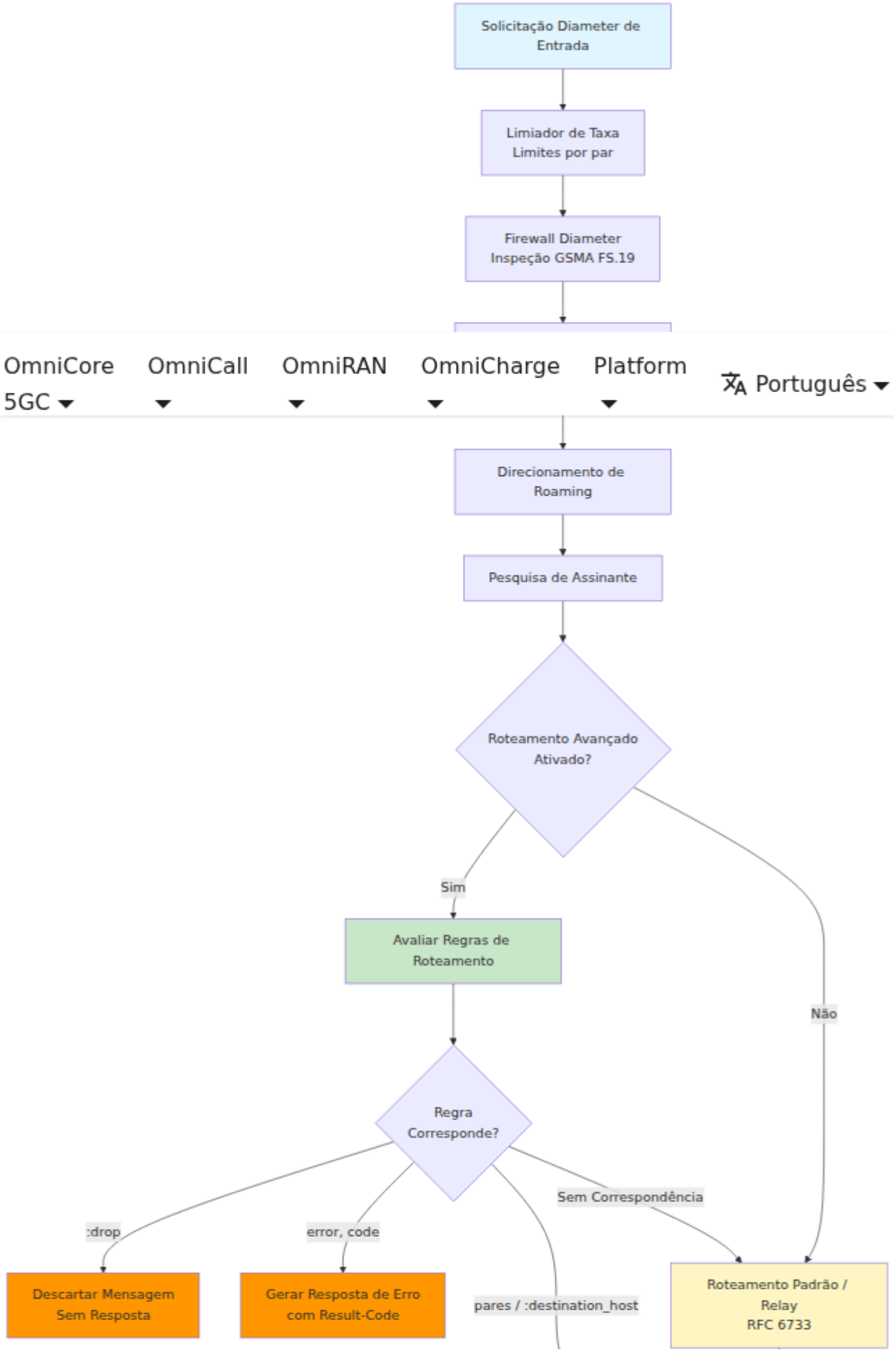
O DRA retransmite duas classes fundamentalmente diferentes de mensagens e as trata de forma diferente.

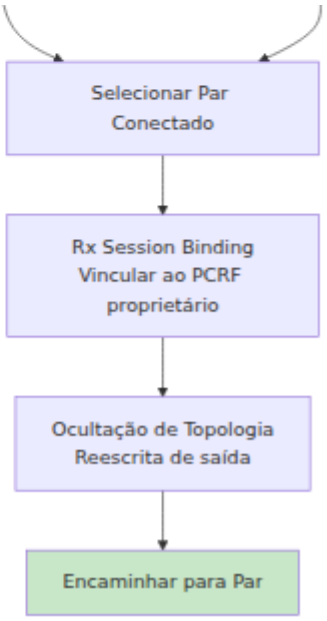
Solicitações

As solicitações percorrem todo o pipeline de processamento. O DRA seleciona um par de destino, atualiza os AVPs de roteamento que gerencia e encaminha a mensagem. O DRA é um relay, portanto, passa a carga útil da aplicação inalterada.

Respostas

As respostas nunca reentram no pipeline de roteamento. Em vez disso, o DRA correlaciona cada resposta à sua solicitação original. O DRA então retorna a resposta ao longo do caminho reverso.


Solicitação Diameter de Entrada
Limiador de Taxa Limites por par
Firewall Diameter Inspeção GSMA FS.19
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Direcionamento de Roaming
Pesquisa de Assinante
Roteamento Avançado Ativado?
Sim
Não
Avaliar Regras de Roteamento
Regra Corresponde?
:drop
error, code
pares / :destination_host
Sem Correspondência
Descartar Mensagem Sem Resposta
Gerar Resposta de Erro com Result-Code
Roteamento Padrão / Relay RFC 6733



As regras de retransmissão para respostas são:

- **Roteamento baseado em sessão** - As respostas sempre seguem o caminho reverso de sua solicitação.
- **Preservação do ID de ponta a ponta** - O **Identificador de Ponta a Ponta** permanece inalterado em cada salto.
- **Roteamento Hop-by-Hop** - O **Hop-by-Hop Identifier** muda a cada salto. O DRA o utiliza para manter o estado de roteamento e para corresponder uma resposta de volta ao par que enviou a solicitação.
- **Sem avaliação de regras** - O DRA não avalia regras de roteamento ou inspeciona conteúdos de AVP para respostas.
- **Correlação com estado** - Tabelas de roteamento internas rastreiam qual par enviou cada solicitação. O DRA limpa esse estado após retransmitir a resposta.

O roteamento de respostas é determinístico. O protocolo Diameter exige que uma resposta retorne ao par original ao longo do caminho de solicitação estabelecido. Isso preserva a gestão correta de sessões e previne loops de roteamento. Veja RFC 6733 §6.2 para detalhes sobre o roteamento de mensagens de resposta.

Tabelas de Referência

IDs de Aplicação 3GPP Comuns

ID	Interface	Descrição	Referência
16777251	S6a/S6d	Autenticação e assinatura MME/SGSN ↔ HSS	3GPP TS 29.272
16777238	Gx	Controle de políticas e cobrança PCEF ↔ PCRF	3GPP TS 29.212
4	Gy	Cobrança online (controle de crédito) ↔ OCS	3GPP TS 32.299

AVPs de Roteamento Comuns

Código	Nome	Descrição	Referência
293	Destination-Host	Alvo de roteamento explícito a nível de host (maior prioridade)	RFC 6733 §6.5
283	Destination-Realm	Domínio usado para selecionar um par conectado	RFC 6733 §6.4
282	Route-Record	Registra a identidade dos relays que uma mensagem percorreu	RFC 6733 §6.7.1

Leitura Adicional

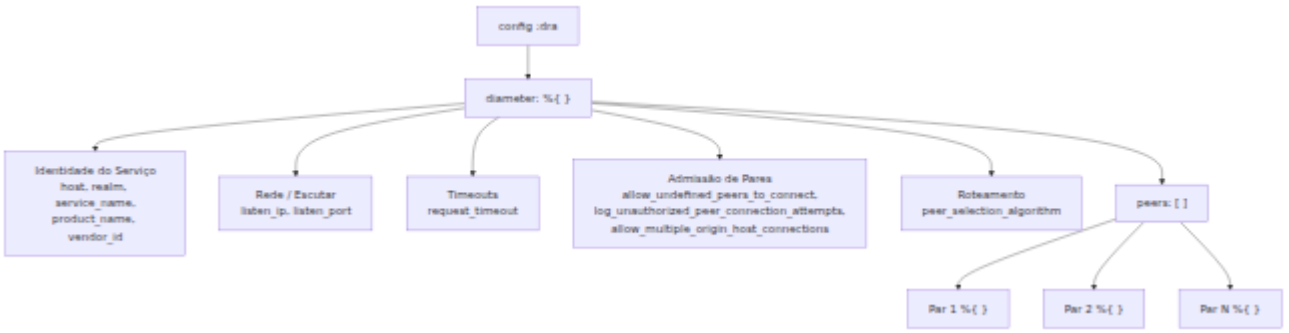
- **Roteamento Padrão** - Roteamento de prioridade RFC 6733, seleção de pares e roteamento de respostas em detalhes.
- **Roteamento Avançado** - Roteamento baseado em regras, resultados de descarte/erro e direcionamento de pares.
- **Configuração** - Configuração base do DRA, transportes e configurações de módulo.
- **Segurança** - Limitação de taxa, firewall Diameter GSMA FS.19, sanitização de AVP e ocultação de topologia.

Configuração Base do DRA

A **Configuração Base do DRA** define a identidade do OmniDRA, ouvintes de rede, timeouts, política de admissão de pares e a lista de pares Diameter aos quais se conecta. Este bloco é a base para cada decisão de roteamento que o agente faz. Declare-o sob `config :dra, diameter: %{...}` em `config/runtime.exs`.

Visão Geral

Todas as configurações base vivem em um único mapa `diameter`. O mapa contém um conjunto de chaves de nível superior que descrevem o próprio DRA. Ele também contém uma lista `peers`. Cada entrada na lista descreve um par Diameter.



Estrutura da Configuração

```
config :dra,
  diameter: %{
    service_name: :omnitouch_dra,
    listen_ip: "127.0.0.2",
    listen_port: 3868,
    host: "dra01",
    realm: "production.omnitouch",
    product_name: "OmniDRA",
    vendor_id: 10415,
    request_timeout: 5000,
    allow_undefined_peers_to_connect: false,
    log_unauthorized_peer_connection_attempts: true,
    allow_multiple_origin_host_connections: false,
    peer_selection_algorithm: :random,
    peers: [
      # Configurações de pares...
    ]
  }
```

Parâmetros de Nível Superior

Essas chaves configuram a própria identidade do DRA, seu ouvinte de rede, timeouts e sua política de admissão e seleção de pares. Cada chave aparece uma vez no mapa `diameter`.

Parâmetro	Tipo	Requerido	P
`service_name`	Atom	Sim	-
`listen_ip`	String ou Lista	Sim	-
`listen_port`	Inteiro	Sim	-
`host`	String	Sim	-
`realm`	String	Sim	-
`product_name`	String	Sim	-

Parâmetro	Tipo	Requerido	P
`vendor_id`	Inteiro	Sim	-
`request_timeout`	Inteiro	Não	`5`
`allow_undefined_peers_to_connect`	Booleano	Sim	`f`
`log_unauthorized_peer_connection_attempts`	Booleano	Sim	`t`
`allow_multiple_origin_host_connections`	Booleano	Sim	`f`

Parâmetro	Tipo	Requerido	P
`peer_selection_algorithm`	Atom	Sim	-
`peer_settle_time_ms`	Inteiro	Não	`0`
`watchdog_timer_ms`	Inteiro	Não	`3`
`peers`	Lista	Sim	-

Tamanho do Buffer do Socket SCTP

O tamanho do buffer do socket é uma chave de nível **superior** `:dra`. Configure-o com `config :dra, diameter_socket_buffer_size:`. Não é um membro do mapa `diameter:`.

```
config :dra, diameter_socket_buffer_size: 4_194_304

config :dra,
  diameter: %{
    # ... configuração base ...
  }
```

Parâmetro	Tipo	Requerido	Padrão	
`diameter_socket_buffer_size`	Inteiro	Não	`4194304` (4 MB)	Tamanh recebim SCTP em nível **su** `:dra,` `diamete` não um `diamete` `:diamet` Multihor

Seleção de Pares

Quando mais de um par configurado corresponde aos critérios de roteamento para uma solicitação, `peer_selection_algorithm` determina qual par a recebe. Esta configuração afeta como o DRA distribui o tráfego e como se comporta durante falhas de pares. Veja Roteamento Padrão para o fluxo completo de decisão de roteamento. Veja Roteamento Avançado para substituições baseadas em regras.

Valor	Comportamento
`:random`	O DRA escolhe um par correspondente aleatoriamente para cada solicitação. Isso espalha a carga entre todos os pares elegíveis.
`:failover`	O DRA prefere o primeiro par correspondente. Ele usa pares posteriores apenas quando os pares de maior prioridade estão indisponíveis.

Configuração de Pares

Cada entrada na lista `peers` é um mapa que descreve um único par Diameter. O mapa define sua identidade, seu endereço de rede, o transporte que usa e se o DRA disca o par ou espera ser discado.

```
%{
  host: "diameter-benchmark-server.production.omnitouch",
  realm: "production.omnitouch",
  ip: "127.0.0.3",
  port: 3868,
  transport: :diameter_tcp,
  initiate_connection: true
}
```

Parâmetros por Par

Parâmetro	Tipo	Requerido	Padrão	Descriç
`host`	String	Sim	-	A Identidac Diameter d (FQDN). Es valor deve correspond exatament sensível a maiúsculas minúsculas Regras de roteamento Roteament Avançado exigem um correspond exata.
`realm`	String	Sim	-	O realm Diameter d
`ip`	String	Sim	-	O endereço primário do usado para conexão.
`additional_ips`	Lista	Não	-	Lista de endereços para uma associação multi-home Quando nã definido, o usa `ip`. Est

Parâmetro	Tipo	Requerido	Padrão	Descriç
				chave se a apenas cor `transport` `:diameter_` Veja Multihomin SCTP.
`port`	Inteiro	Sim	-	A porta Diameter d (tipicamen `3868`).
`transport`	Atom	Sim	-	Protocolo d transporte a conexão: `:diameter_` para TCP, c `:diameter_` para SCTP. a referênci Transportes abaixo.
`source_ip`	String	Não	`listen_ip`	Endereço I local a ser vinculado quando o D disca este (apenas conexões c saída). O p é o `listen` do serviço.

Parâmetro	Tipo	Requerido	Padrão	Descriç
`source_port`	Inteiro	Não	-	Porta local vinculada quando o D disca este (apenas conexões d saída). Qua não definid SO atribui porta efêm Se dois par compartilh o mesmo IP destino e p defina um explícito. C conexão us um socket distinto.
`initiate_connection`	Booleano	Sim	-	Quando `tr` DRA conec ao par (pap cliente). Qu `false`, o D espera que par se cone (papel de servidor). V Modos de Conexão.
`origin_host`	String	Não	serviço `host.realm`	O `Origin-H` que o DRA apresenta

Parâmetro	Tipo	Requerido	Padrão	Descriç
				este par. Es chave é separada d `host` acima é a própria identidade par. Defina chave quar um par esp que o DRA um host específico exemplo, u realm parc O DRA apli substituiçã `CER`/`CEA` e mensagens protocolo b nesta cone Veja Identi por Par.
`origin_realm`	String	Não	serviço `realm`	O `Origin-R` que o DRA apresenta este par. Es chave é separada d `realm` acin que é o pró realm dest Defina esta chave com `origin_hos` por conta

Parâmetro	Tipo	Requerido	Padrão	Descriç
				própria. Ve Identidade Par.

> **Multihoming SCTP:** um par que usa `:diameter_sctp` pode vincular múltiplos endereços locais e remotos para resiliência. Multihoming SCTP descreve as configurações de multihoming e listas de endereços por par.

Transportes

Valor do transporte	Protocolo	Referência
`:diameter_tcp`	TCP	RFC 6733 (Base Diameter)
`:diameter_sctp`	SCTP	RFC 4960 (SCTP) - veja Multihoming SCTP

Modos de Conexão

A chave `initiate_connection` controla se o DRA disca um par ou espera ser discado.

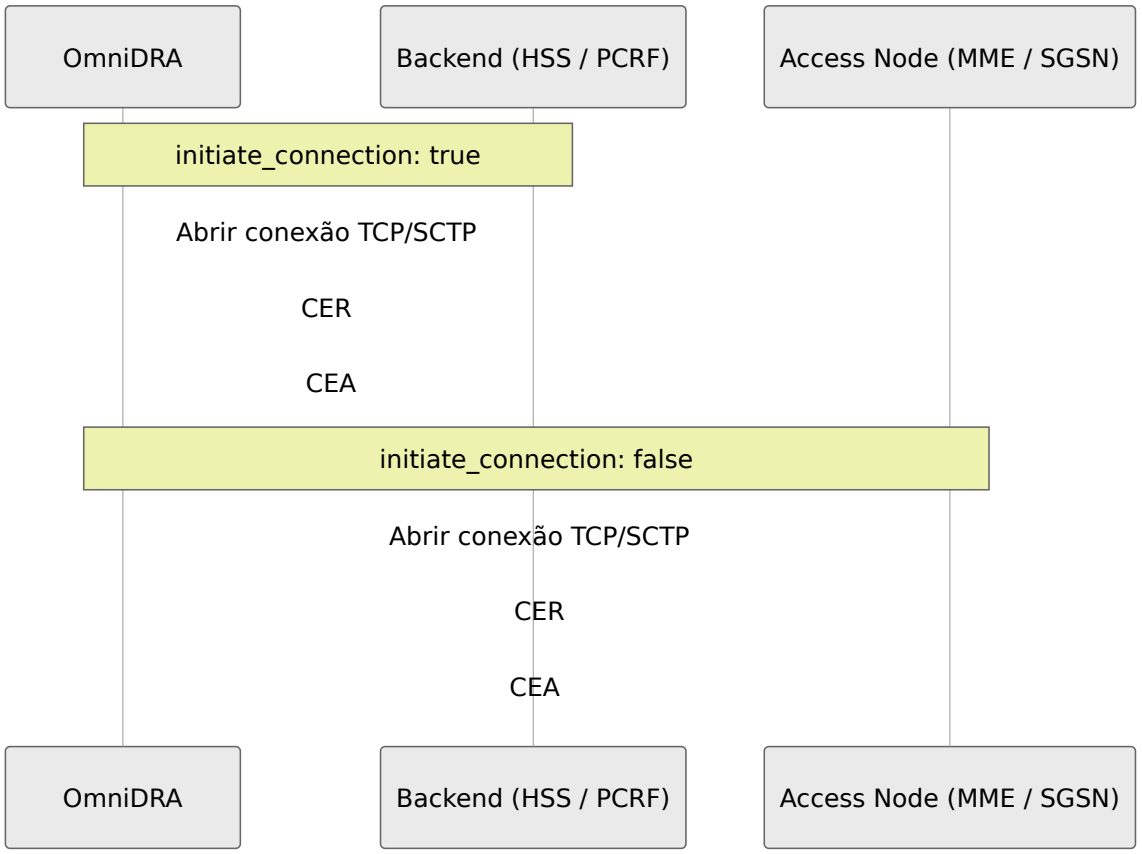
Iniciar Conexão (`initiate_connection: true`)

- O DRA atua como cliente Diameter e inicia a conexão TCP/SCTP com o par.
- Use este modo para sistemas de backend, como um HSS ou PCRF.
- Se o par estiver inacessível, o DRA tenta reconectar.

Aceitar Conexão (`initiate_connection: false`)

- O DRA atua como servidor Diameter e espera que o par se conecte.
- Use este modo para nós do lado de acesso, como um MME, SGSN ou P-GW.

- O par deve estar presente em `peers`, a menos que você defina `allow_undefined_peers_to_connect` como `true`.
- O DRA diferencia pares de entrada pelo `Origin-Host` que eles anunciam no CER. Dois nós Diameter podem compartilhar um IP de origem. O DRA os combina de forma independente quando cada um apresenta um `host` distinto.



Identidade por Par

O DRA tem uma identidade de serviço única. Ele combina `host` e `realm` no `Origin-Host` (por exemplo `dra01.production.omnitouch`) e anuncia `realm` como o `Origin-Realm`. Por padrão, o DRA apresenta essa identidade a cada par.

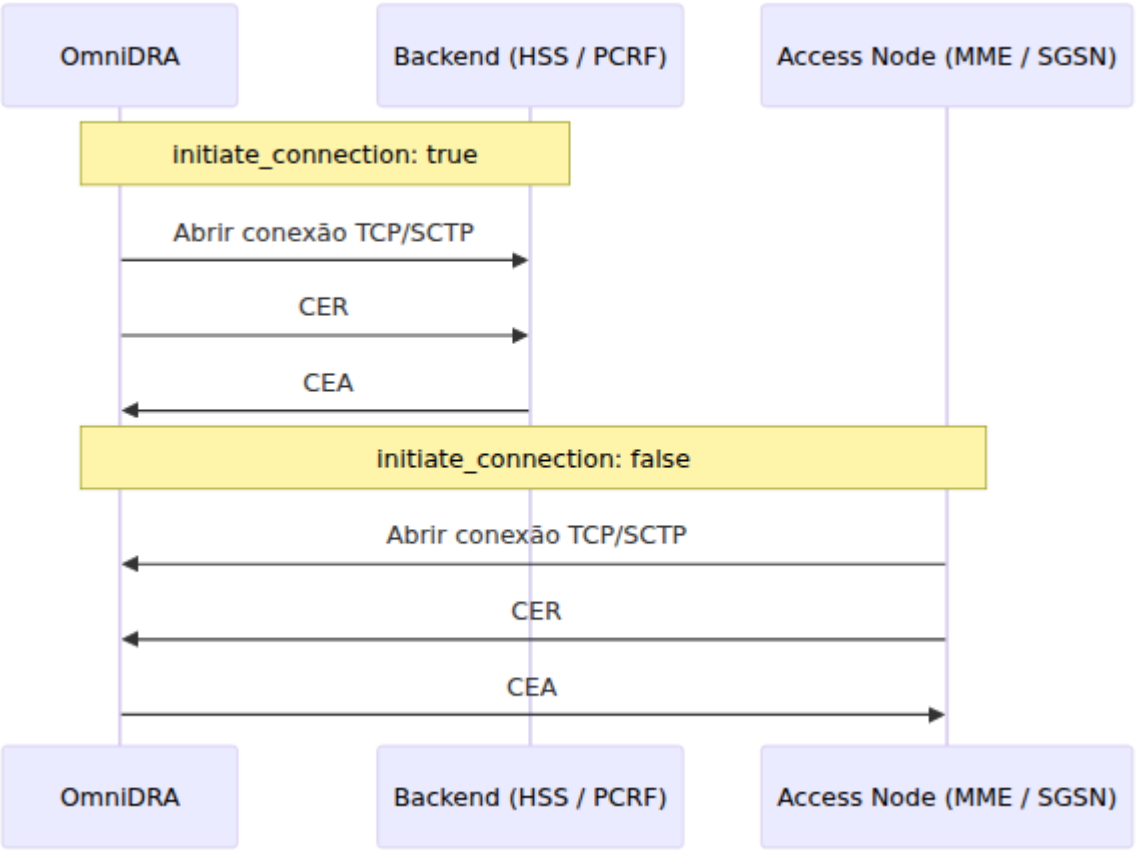
Alguns pares esperam que o DRA apresente uma identidade diferente. Um parceiro de interconexão ou roaming frequentemente exige que o DRA use um host e realm do esquema de nomenclatura do parceiro. Defina `origin_host` e `origin_realm` nesse único par para alterar a identidade que o DRA apresenta nessa conexão.

Essas duas chaves são a identidade que o DRA apresenta **para** o par. Elas são separadas do próprio `host` e `realm` do par, que são a identidade que o par anuncia **para** o DRA. O DRA aplica a substituição apenas a essa conexão. Cada outro par continua a ver a identidade de serviço.

```
peers: [
  %{
    # A própria identidade deste par (o que ele anuncia ao DRA):
    host: "hss.partner.example",
    realm: "partner.example",
    ip: "10.0.0.9",
    port: 3868,
    transport: :diameter_sctp,
    initiate_connection: true,
    # A identidade que o DRA apresenta PARA este par:
    origin_host: "dra-partner.epc.mnc001.mcc001.3gppnetwork.org",
    origin_realm: "epc.mnc001.mcc001.3gppnetwork.org"
  }
]
```

Veja Parâmetros por Par para as definições dos parâmetros `origin_host` e `origin_realm`.

O diagrama abaixo mostra o DRA apresentando sua identidade padrão a um par e uma identidade substituta a um par parceiro.



Comportamento

A substituição altera a identidade na Troca de Capacidades (`CER` / `CEA`) nessa conexão. Ela também altera a identidade nas mensagens de protocolo base nessa conexão: as mensagens de watchdog (`DWR` / `DWA`) e as mensagens de desconexão (`DPR` / `DPA`).

Você pode definir qualquer uma das chaves isoladamente. Uma chave não definida recai sobre o valor do serviço. Se você definir apenas `origin_realm`, o DRA altera apenas o realm e mantém o `Origin-Host` de serviço.

O DRA usa o `Origin-Host` substituto para detecção de loops nessa conexão. Se uma solicitação já carregar um AVP `Route-Record` que corresponda à substituição, o DRA ainda detecta o loop conforme RFC 6733 §6.1.8.

O DRA fixa a identidade quando envia o `CER`. Se você alterar uma substituição e recarregar a configuração em tempo de execução, o DRA restabelece essa conexão para que a nova identidade entre em vigor. Conexões com outros pares não são afetadas.

Exemplo Completo

O seguinte exemplo configura um DRA. O DRA aceita conexões de um MME via SCTP. Ele disca um HSS e um PCRF via TCP.

```
config :dra,
  diameter: %{
    service_name: :mvno_dra,
    listen_ip: "10.100.1.10",
    listen_port: 3868,
    host: "dra01",
    realm: "mvno.example.com",
    product_name: "OmniDRA",
    vendor_id: 10415,
    request_timeout: 5000,
    allow_undefined_peers_to_connect: false,
    log_unauthorized_peer_connection_attempts: true,
    allow_multiple_origin_host_connections: false,
    peer_selection_algorithm: :random,
    peers: [
      # MME - DRA espera que o MME se conecte
      %{
        host: "mme01.operator.example.com",
        realm: "operator.example.com",
        ip: "10.100.2.15",
        port: 3868,
        transport: :diameter_sctp,
        initiate_connection: false
      },
      # HSS - DRA inicia a conexão a partir de uma porta local de
origem explícita
      %{
        host: "hss01.mvno.example.com",
        realm: "mvno.example.com",
        ip: "10.100.3.141",
        port: 3868,
        source_port: 13868,
        transport: :diameter_tcp,
        initiate_connection: true
      },
      # PCRF - DRA inicia a conexão
      %{
        host: "pcrf01.mvno.example.com",
        realm: "mvno.example.com",
        ip: "10.100.3.22",
        port: 3868,
        transport: :diameter_tcp,
        initiate_connection: true
```

```
      }
    ]
  }
```

Notas Operacionais

- **Correspondência de Nome de Host** - O DRA corresponde exatamente e sensivelmente a maiúsculas e minúsculas os valores de `host` dos pares. Regras em Roteamento Avançado devem referenciar a mesma string de `host` que você configura aqui.
- **Troca de Capacidades** - Ao conectar, o DRA e cada par trocam aplicações e identidades suportadas através de CER/CEA. O DRA anuncia os valores de `host`, `realm`, `product_name` e `vendor_id` nesta fase.
- **Vendor-ID 10415** - Este é o padrão `Vendor-Id` 3GPP. É o valor esperado para interfaces Diameter 3GPP.
- **Seleção de Pares** - Quando múltiplos pares correspondem a uma solicitação, `peer_selection_algorithm` decide qual deles o DRA usa. Veja Roteamento Padrão.

Considerações de Segurança

- Defina `allow_undefined_peers_to_connect: false` em produção para que apenas pares conhecidos possam se conectar.
- Ative `log_unauthorized_peer_connection_attempts: true` para registrar tentativas de conexão rejeitadas para monitoramento de segurança.
- Certifique-se de que as regras de firewall estejam alinhadas com o `listen_ip` e `listen_port` configurados.

Documentação Relacionada

- Multihoming SCTP - Transporte SCTP multi-homed e configuração de endereços por par.
- Roteamento Padrão - Como o DRA seleciona pares e roteia solicitações, incluindo o comportamento de `peer_selection_algorithm`.

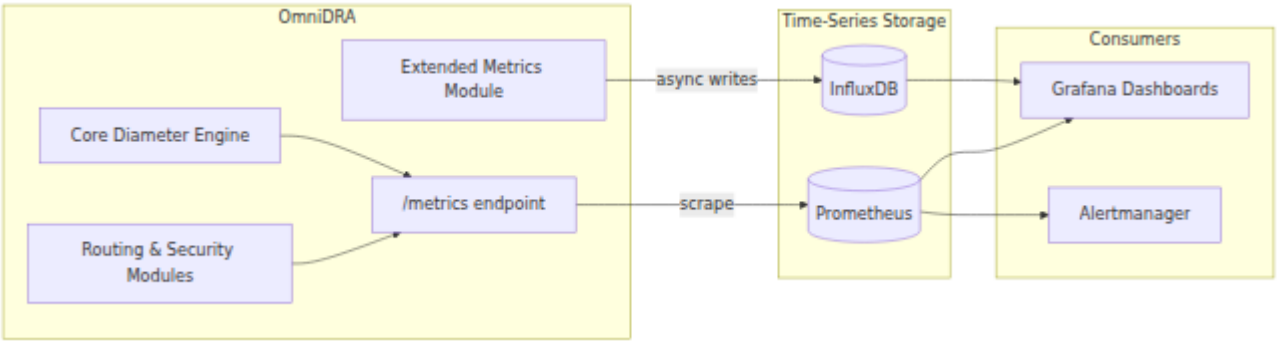
- Roteamento Avançado - Substituições de roteamento baseadas em regras com base nos valores de `host` dos pares.

Métricas & Monitoramento

OmniDRA expõe métricas abrangentes do **Prometheus**. Elas cobrem conectividade entre pares, taxas de mensagens, resultados de sucesso e erro, latência de resposta, tráfego descartado e tentativas de conexão não autorizadas. Esta página documenta cada métrica, seus rótulos e consultas PromQL prontas para uso. Use-as para construir painéis e alertas.

Fluxo de Observabilidade

OmniDRA publica todas as métricas no formato de exposição do Prometheus no endpoint `/metrics`. O Prometheus coleta este endpoint em um intervalo regular e armazena as amostras. Painéis ou ferramentas de alerta então consultam a série temporal resultante.



O Prometheus coleta as **métricas padrão do Prometheus** do `/metrics`. O **Módulo de Métricas Estendidas** publica telemetria de alta cardinalidade, em nível de assinante, diretamente no **InfluxDB**. Isso mantém a contagem de séries do Prometheus gerenciável (veja Módulo de Métricas Estendidas).

> OmniDRA também expõe **métricas de tempo de execução BEAM/Erlang** em `/metrics`. Esta página não as documenta. Todos os contadores são cumulativos. Consulte-os com `rate()` para obter taxas por segundo.

Métricas de Diameter Core

Essas métricas estão sempre disponíveis. Elas descrevem a saúde dos pares e o fluxo de mensagens através do DRA.

Conectividade entre Pares

Status do Par

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

Rótulos:

Rótulo	Descrição
`origin_host`	Identidade Diameter do par (FQDN).
`ip`	Endereço IP do par.

Consultas de exemplo:

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

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

Para diagnosticar pares que flutuam ou nunca alcançam um estado conectado, veja Solução de Problemas.

Aplicações Suportadas

- **Métrica**: `diameter_peer_supported_applications`
- **Tipo**: Gauge

- **Descrição**: As aplicações que um par anunciou durante a troca de capacidades (CER/CEA), quando conhecidas.

Rótulos:

Rótulo	Descrição
`origin_host`	Identidade Diameter do par (FQDN).
`supported_applications`	As aplicações que o par anunciou como suportadas.

Consultas de exemplo:

```
# Aplicações anunciadas por um par específico
diameter_peer_supported_applications{origin_host="hss01.example.com"}
```

Taxas de Mensagens

Contagem de Mensagens

- **Métrica**: `diameter_peer_message_count_total`
- **Tipo**: Counter
- **Descrição**: Número total de mensagens Diameter trocadas com pares.

Rótulos:

Rótulo	Descrição
`origin_host`	Identidade Diameter do par.
`received_from`	Par que enviou a mensagem.
`application_id`	Diameter Application-Id. Veja IDs de Aplicação.
`cmd_code`	Diameter Command-Code. Veja Códigos de Comando.
`application_name`	Nome da aplicação legível por humanos (ex: `3GPP_S6a`).
`cmd_name`	Nome do comando legível por humanos (ex: `AIR`).
`direction`	`request` ou `response`.

Consultas de 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]))
```

Taxas de Sucesso & Erro

Códigos de Resultado de Resposta

- **Métrica**: `diameter_peer_message_result_code_count_total`
- **Tipo**: Counter

- **Descrição**: Número total de respostas Diameter por código de resultado.

Rótulos:

Rótulo	Descrição
`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.

Consultas de 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:

Código	Nome	Referência
`2001`	DIAMETER_SUCCESS	RFC 6733 §7.1.1
`3002`	DIAMETER_UNABLE_TO_DELIVER	RFC 6733 §7.1.3
`3003`	DIAMETER_REALM_NOT_SERVED	RFC 6733 §7.1.3
`3004`	DIAMETER_TOO_BUSY	RFC 6733 §7.1.3
`5001`	DIAMETER_AUTHENTICATION_REJECTED	RFC 6733 §7.1.5
`5004`	DIAMETER_INVALID_AVP_VALUE	RFC 6733 §7.1.5
`5012`	DIAMETER_UNABLE_TO_COMPLY	RFC 6733 §7.1.5

Latência

Atraso de Resposta

- **Métrica**: `diameter_peer_last_response_delay`
- **Tipo**: Gauge
- **Descrição**: Atraso da resposta mais recente em milissegundos (DRA → Par → DRA).

Rótulos:

Rótulo	Descrição
`origin_host`	Solicitante original.
`routed_to`	Par que enviou a resposta.
`application_name`	Nome da aplicação.
`cmd_name`	Nome do comando.

Consultas de exemplo:

```
# Atraso da resposta mais recente do HSS
diameter_peer_last_response_delay{routed_to="hss01.example.com"}

# Maior atraso recente de resposta entre pares S6a
max(diameter_peer_last_response_delay{application_name="3GPP_S6a"})
```

> Este é um gauge que mantém o atraso mais recente por par. Não é um histograma, portanto funções de quantil como `histogram_quantile` não se aplicam.

Requisições Não Respondidas

Contagem de Requisições Não Respondidas

- **Métrica**: `diameter_peer_unanswered_request_count_total`
- **Tipo**: Counter
- **Descrição**: Requisições que o DRA enviou, mas que o par não respondeu dentro do período de timeout.

Rótulos:

Rótulo	Descrição
`origin_host`	Solicitante original.
`routed_to`	Par que não respondeu.
`application_id`	Diameter Application-Id.
`cmd_code`	Diameter Command-Code.
`application_name`	Nome da aplicação.
`cmd_name`	Nome do comando.

Consultas de exemplo:

```
# Taxa de requisições não respondidas
rate(diameter_peer_unanswered_request_count_total[5m])

# Identificar pares problemáticos
topk(5, sum by (routed_to) (
  rate(diameter_peer_unanswered_request_count_total[5m])
))
```

Conexões Não Autorizadas

Tentativas de Conexão Não Autorizadas

- **Métrica**: `diameter_peer_unauthorized_connection_count_total`
- **Tipo**: Counter
- **Descrição**: Tentativas de conexão de pares não autorizados.

Rótulos:

Rótulo	Descrição
`origin_host`	Identidade reivindicada do par não autorizado.
`supported_applications`	Aplicações anunciadas pelo par.
`peer_ip`	Endereço IP da tentativa de conexão.

Consultas de exemplo:

```
# Tentativas de conexão não autorizadas
rate(diameter_peer_unauthorized_connection_count_total[5m])

# Alerta sobre acesso não autorizado
diameter_peer_unauthorized_connection_count_total > 0
```

Para detalhes mais profundos sobre métricas do módulo de segurança (firewall, limitação de taxa, ocultação de topologia, sanitização de AVP), veja Segurança.

Métricas do Módulo de Roteamento Avançado

Essas métricas descrevem os resultados que as regras de roteamento avançadas produzem. Veja a documentação do módulo de segurança em Segurança para telemetria de controle de tráfego relacionada.

Tráfego Descartado

Contagem de Descartes

- **Métrica**: `diameter_advanced_routing_drop_count_total`
- **Tipo**: Counter
- **Descrição**: Requisições que a ação `:drop` do roteamento avançado descartou. O DRA não envia resposta.

Rótulos:

Rótulo	Descrição
`application_id`	Diameter Application-Id.
`cmd_code`	Diameter Command-Code.
`application_name`	Nome da aplicação.
`cmd_name`	Nome do comando.

Consultas de exemplo:

```
# Taxa de descarte por comando
sum by (cmd_name) (
  rate(diameter_advanced_routing_drop_count_total[5m])
)

# Tráfego total descartado
sum(rate(diameter_advanced_routing_drop_count_total[5m]))
```

Quando o DRA descarta tráfego:

- Uma regra de roteamento avançado corresponde a uma ação **drop**.
- O DRA descarta a mensagem.
- O DRA não envia mensagem de resposta ao par solicitante.

Limitação de Taxa & Sobrecarga

Respostas de Erro

- **Métrica**: `diameter_advanced_routing_error_count_total`
- **Tipo**: Counter
- **Descrição**: Respostas de erro que a ação de erro do roteamento avançado gerou.

Rótulos:

Rótulo	Descrição
`result_code`	Código de Resultado Diameter enviado na resposta de erro.
`application_id`	Diameter Application-Id.
`cmd_code`	Diameter Command-Code.
`application_name`	Nome da aplicação.
`cmd_name`	Nome do comando.

Consultas de exemplo:

```
# Respostas de erro por código de resultado
sum by (result_code) (
  rate(diameter_advanced_routing_error_count_total[5m])
)

# Respostas DIAMETER_TOO_BUSY
rate(diameter_advanced_routing_error_count_total{
  result_code="3004"
}[5m])

# Detecção de limitação de taxa ou sobrecarga
diameter_advanced_routing_error_count_total{
  result_code="3004"
} > 100
```

Quando o DRA gera respostas de erro:

- Uma regra de roteamento avançado corresponde a uma ação **error** para um Código de Resultado especificado.
- O DRA gera uma resposta Diameter com o Código de Resultado especificado.

- O DRA envia a resposta de volta ao par solicitante. Ele não encaminha a resposta.

Códigos de Erro Comuns Usados:

Código	Nome	Significado
`3002`	DIAMETER_UNABLE_TO_DELIVER	Falha de roteamento.
`3003`	DIAMETER_REALM_NOT_SERVED	Realm não suportado.
`3004`	DIAMETER_TOO_BUSY	Proteção contra sobrecarga.
`5012`	DIAMETER_UNABLE_TO_COMPLY	Erro geral.

Módulo de Métricas Estendidas

O **Módulo de Métricas Estendidas** fornece telemetria e análises avançadas. Use-o para analisar padrões de tráfego Diameter além das métricas padrão. Atualmente, ele rastreia tentativas de conexão de assinantes LTE. Ele correlaciona pares de mensagens de Solicitação de Informação de Autenticação (AIR) e Resposta (AIA) S6a.

Essas métricas carregam campos de **alta cardinalidade** como IMSI. O módulo, portanto, escreve-os de forma assíncrona no **InfluxDB** e não os expõe no endpoint `/metrics` do Prometheus. Isso mantém a contagem de séries do Prometheus limitada e evita a explosão de métricas.

Configuração

O módulo está desativado por padrão. Configure-o em `config/runtime.exs`:

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

Parâmetro	Tipo	Obrigatório	Padrão
`enabled`	Booleano	Não	`false`
`attach_attempt_reporting_enabled`	Booleano	Não	`false`

Conexão InfluxDB

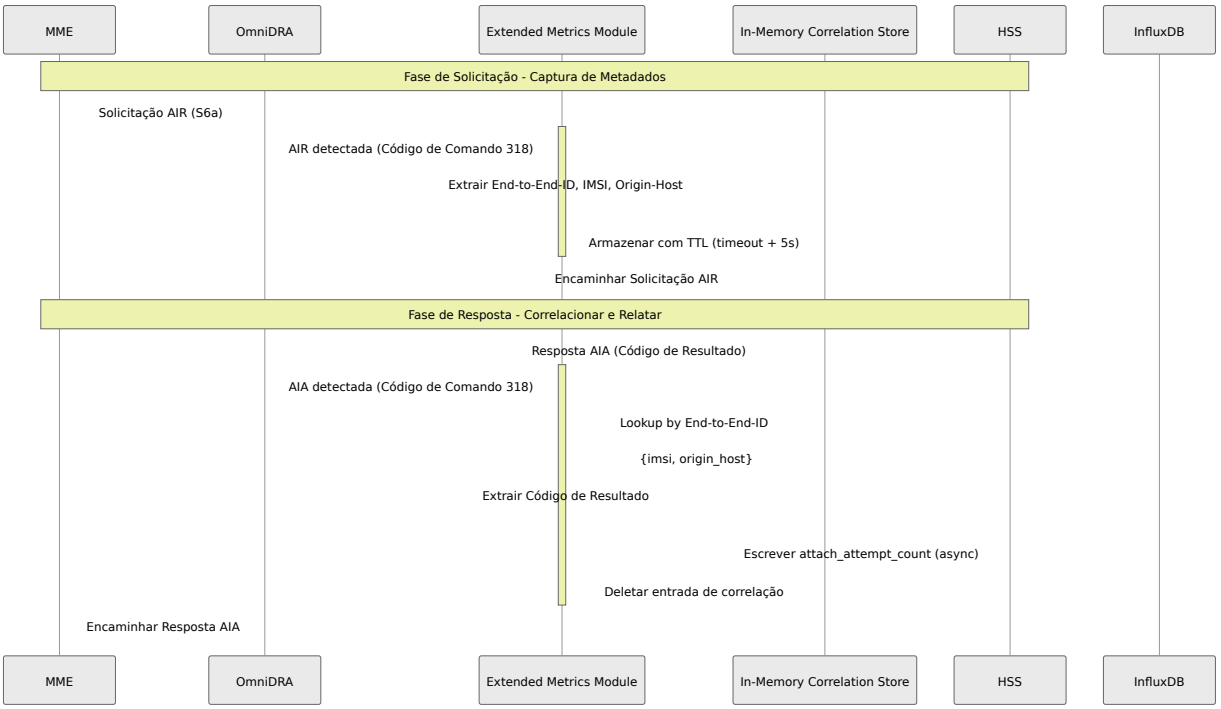
O Módulo de Métricas Estendidas escreve de forma assíncrona no InfluxDB através de uma conexão Instream (Instream). Configure o alvo em `config :dra, DRA.Metrics.InfluxDB`.

```
config :dra, DRA.Metrics.InfluxDB,
  auth: [username: "example_user", password: "example_password"],
  database: "dra",
  host: "127.0.0.15",
  port_udp: 8089,
  writer: Instream.Writer.Line
```

Parâmetro	Tipo	Obrigatório	Padrão	De
`auth`	Lista de palavras-chave	Não	-	Credenci InfluxDB `[usernar` `passworc`
`database`	String	Sim	-	Banco de InfluxDB recebe a métricas
`host`	String	Sim	-	Nome do endereço servidor
`port_udp`	Inteiro	Não	`8089`	Porta UD dados de linha no a porta c do Influx padrão), da API H
`writer`	Módulo	Não	`Instream.Writer.Line`	Módulo e Instream `Instrear` escreve protocolo

Rastreamento de Tentativas de Conexão

O módulo monitora pares AIR/AIA S6a. Ele correlaciona cada solicitação com sua resposta através do Diameter End-to-End-ID. Em seguida, emite uma métrica em nível de assinante. A métrica carrega o par de origem e o código de resultado do HSS.



Medição: `attach_attempt_count`

Campos:

Campo	Descrição
`imsi`	O IMSI do assinante, retirado do AVP User-Name.

Tags:

Tag	Descrição
`origin_host`	O par que originou a solicitação de conexão.
`result_code`	O código de resultado Diameter da resposta do HSS.

Códigos de resultado escritos para esta medição são códigos Diameter padrão:

Código	Nome
`2001`	DIAMETER_SUCCESS
`5001`	DIAMETER_AUTHENTICATION_REJECTED
`5004`	Diameter AVP unsupported

Veja RFC 6733 para a lista completa de códigos de resultado.

Casos de Uso

- **Análise da Taxa de Sucesso de Conexão** - Rastrear tentativas de conexão bem-sucedidas versus falhadas por código de resultado.
- **Solução de Problemas em Nível de IMSI** - Identificar assinantes individuais que têm falhas de conexão.
- **Monitoramento de Desempenho da Rede** - Monitorar padrões de tentativas de conexão por origem (MME/SGSN).
- **Análise de Roaming** - Analisar taxas de sucesso de conexão de roaming inbound.

Notas Operacionais

- As métricas de tentativas de conexão rastreiam apenas pares S6a AIR/AIA (Application-Id `16777251`, Código de Comando `318`).
- Os metadados da solicitação expiram após o timeout de solicitação configurado mais 5 segundos.
- O processamento de métricas é assíncrono. Não bloqueia o fluxo de mensagens.
- O módulo opera independentemente dos módulos de roteamento e transformação.

Tabelas de Referência

Referência de ID de Aplicação

ID	Aplicação	Descrição	Referência
`16777251`	S6a/S6d	MME/SGSN ↔ HSS autenticação e assinatura	3GPP TS 29.272
`16777238`	Gx	PCEF ↔ PCRF controle de política e cobrança	3GPP TS 29.212
`16777216`	Cx	I-CSCF/S-CSCF ↔ HSS registro IMS	3GPP TS 29.229

Referência de Código de Comando

Código	Comando	Aplicação	Referência
`318`	AIR/AIA (Informação de Autenticação)	S6a	3GPP TS 29.272

Notas

- Todas as métricas padrão estão disponíveis no endpoint `/metrics` no formato Prometheus.
- O **Módulo de Métricas Estendidas** fornece métricas adicionais específicas do S6a através do InfluxDB.
- OmniDRA também expõe **métricas de tempo de execução BEAM/Erlang**. Esta página não as documenta.
- Todos os contadores são cumulativos. Consulte-os com `rate()` para obter taxas por segundo.

- Apenas o Módulo de Métricas Estendidas usa tags de alta cardinalidade como IMSI. Isso evita a explosão de métricas.

Documentação Relacionada

- Solução de Problemas - Diagnóstico de problemas de conectividade, roteamento e entrega.
- Segurança - Detalhes sobre métricas do módulo de segurança (firewall, limitação de taxa, ocultação de topologia, sanitização de AVP).

Conectividade de Par e Tempo de Estabelecimento

Este módulo controla quão rapidamente um par Diameter se torna roteável após se conectar. Ele também impede o tráfego para pares que flutuam. Um par flutuante conecta e desconecta repetidamente.

Contexto

Antes que o OmniDRA roteie para um par, o par deve completar a troca de capacidades (CER/CEA) e ser promovido ao serviço. Por padrão, o watchdog do RFC 3539 mantém um par *reconectando* no estado REOPEN. O par permanece lá até passar por sucessivas trocas de watchdog (DWR/DWA). O temporizador do watchdog **Tw** controla essas trocas, aproximadamente 30 segundos com o padrão de 30000 ms. Essa retenção se aplica mesmo que o CER/CEA já tenha sido bem-sucedido.

O OmniDRA remove esse atraso implícito. O watchdog promove pares reconectando imediatamente. O OmniDRA substitui o atraso por um **tempo de estabelecimento**. O tempo de estabelecimento é uma retenção explícita e configurável que se aplica após o par se conectar. Durante a retenção, o par está conectado, mas não é roteável. Se ele desconectar dentro da janela, o OmniDRA cancela a retenção e não envia tráfego para ele.

Configuração

Ambas as chaves são opcionais e estão no mapa de configuração `:diameter`. Se você não definir nenhuma chave, os pares serão roteados assim que se conectarem.

```
config :dra,
  diameter: %{
    # ...configurações de serviço...
    peer_settle_time_ms: 5000,
    watchdog_timer_ms: 30_000
  }
```

Parâmetro	Tipo	Necessário	Padrão	Descrição
`peer_settle_time_ms`	Inteiro	Não	`0`	Quanto tempo (ms) um par recém-conectado deve permanecer ativo antes que o DRA roteie tráfego para ele. `0` roteia imediatamente. Um valor positivo mantém o par por essa duração. Cancela a retenção se o par desconectar durante a janela.
`watchdog_timer_ms`	Inteiro	Não	`30000`	Substituição para o temporizador do watchdog do RFC 3539 **Tw**, o intervalo de vivacidade DWR/DWA. Reduza para uma detecção mais rápida de pares mortos.

Parâmetro	Tipo	Necessário	Padrão	Descrição
				Isso gera mais tráfego de watchdog.

Defina o tempo de estabelecimento não superior ao intervalo de flutuação no pior caso que você deve suportar. Um valor mais longo aumenta a proteção contra flutuações. Também atrasa o retorno de um par saudável após uma reinicialização legítima.

Monitoramento

Um par em sua janela de estabelecimento relata **down** no medidor `diameter_peer_status` (`0` = down/não roteável ainda, `1` = roteável). Ele relata down até se tornar roteável. A métrica, portanto, reflete o que o DRA pode rotear.

O come-up produz estas linhas de log:

Mensagem de log	Quando
`peer connected, settling before routable`	Par conectado, janela de estabelecimento iniciada (inclui `settle_time_ms`).
`peer disconnected during settle window; never became routable`	Par desconectado antes de se estabelecer. O DRA não roteou tráfego.
`destination peer is settling; request not yet routable...`	Um pedido chegou para um par que ainda está se estabelecendo. O DRA respondeu `DIAMETER_UNABLE_TO_DELIVER (3002)`.
`no suitable peer to route request...`	O DRA não pode rotear o pedido para um destino que **não** está se estabelecendo (down, desconhecido ou mal configurado).

As duas últimas linhas distinguem um estabelecimento transitório esperado de um destino genuinamente ausente.

Referência

- RFC 3539 - Máquina de estados do watchdog Diameter e o temporizador do watchdog (Tw).
- RFC 6733 - Protocolo base Diameter (CER/CEA, DWR/DWA).

Direcionamento de Roaming (SoR)

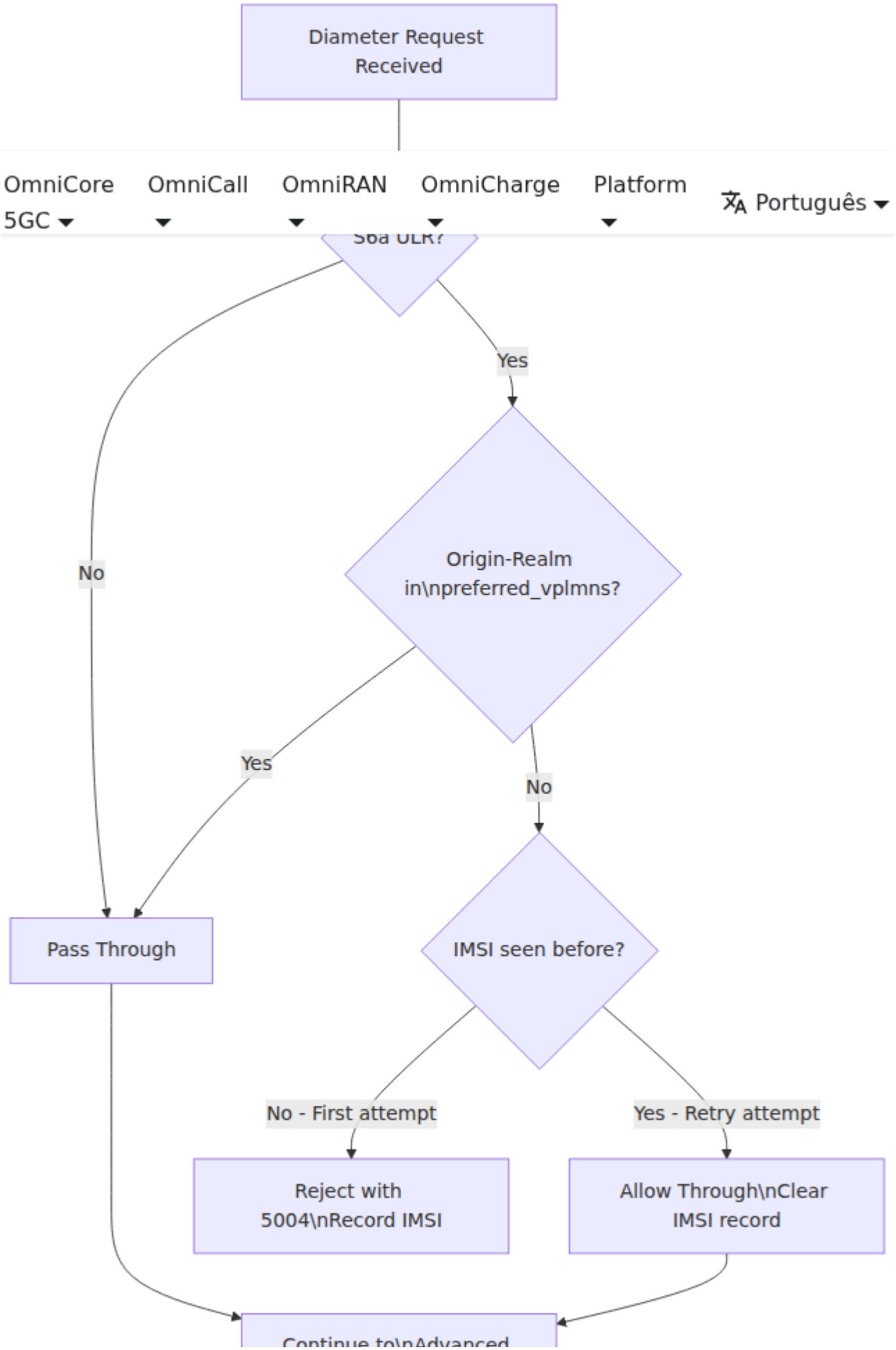
O Direcionamento de Roaming (SoR) permite que o HPLMN influencie a qual VPLMN um assinante em roaming se conecta. Quando um assinante tenta se registrar através de um VPLMN não preferido, o DRA intercepta a S6a Update-Location-Request (ULR). Ele rejeita a ULR com **DIAMETER_ERROR_ROAMING_NOT_ALLOWED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

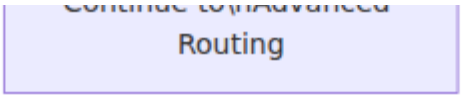
Se a cobertura preferida não estiver disponível e o assinante retornar ao mesmo VPLMN não preferido, o módulo permite o pedido na próxima tentativa.

3GPP TS 29.272 Seção 5.2.1.1 e 3GPP TS 23.122 definem o SoR.

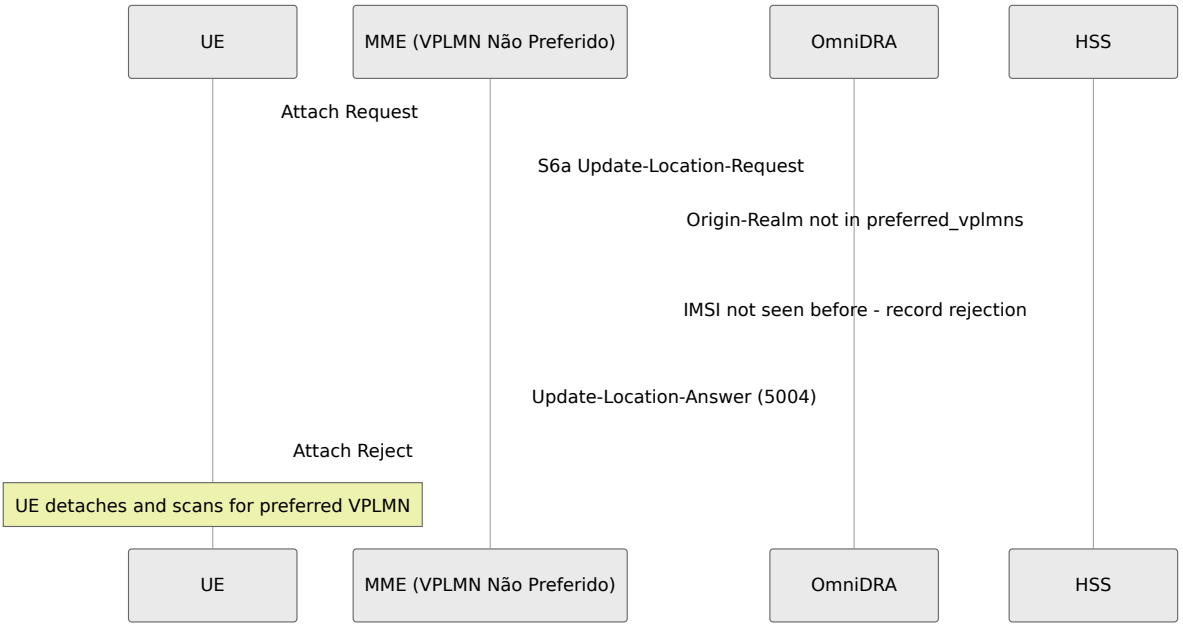
Como Funciona

O módulo opera dentro do pipeline de processamento de solicitações do DRA. Ele é executado **antes** do Roteamento Avançado. O módulo avalia apenas S6a Update-Location-Requests (Application-Id `16777251`, Command-Code `316`). Todas as outras mensagens Diameter passam sem modificações.

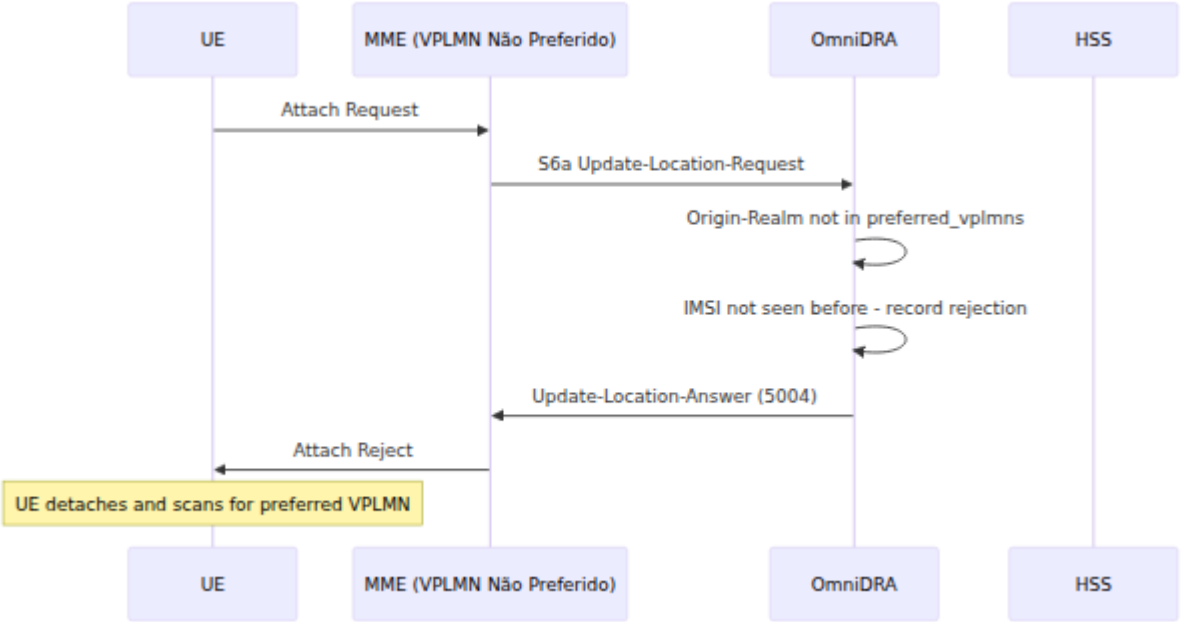

Diameter Request Received
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
S6a ULR?
Yes
No
Origin-Realm in\npreferred_vplmns?
Yes
No
Pass Through
IMSI seen before?
No - First attempt
Yes - Retry attempt
Reject with 5004\nRecord IMSI
Allow Through\nClear IMSI record
Continue to\nAdvanced



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 solicitações. Se o SoR rejeitar uma solicitação, essa mensagem nunca chega ao Roteamento Avançado.

Rastreamento de Assinantes

O módulo rastreia cada assinante rejeitado por IMSI. Cada registro inclui uma contagem de rejeições e um carimbo de data/hora. O módulo limpa registros automaticamente após o TTL configurado expirar.

Quando um IMSI rastreado atinge o limite de `max_rejections` e envia outra ULR, o módulo permite a solicitação. Ele então exclui o registro de rastreamento.

Se nenhuma ULR adicional chegar dentro da janela TTL, o registro expira. O próximo ciclo de limpeza o remove. O módulo então trata uma ULR posterior do mesmo IMSI como uma primeira tentativa e a rejeita novamente.

Configuração

Configure o módulo 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	Requerido	Padrão	
`enabled`	Booleano	Sim	`false`	Habilitar ou d Direcioname `false`, todas modificações
`children`	Lista	Não	`[]`	Processos qu quando habil `[DRA.Router` para que o p IMSI seja exe
`max_rejections`	Inteiro	Não	`1`	Número de v antes que o r Com o padrã primeira ULR
`rejection_ttl_seconds`	Inteiro	Não	`300`	Tempo em se de IMSI rastr remove. Se d novamente d trata a próxi primeira tent controla o int
`rejection_result_code`	Inteiro	Não	`5004`	Código de Re resposta de r DIAMETER_E conforme 3G
`preferred_vplmns`	Lista	Não	`[]`	Lista de strin VPLMNs pref permite uma Realm corres

Exemplos de Configuração

Rejeitar Uma Vez, Depois Permitir

Esse é o comportamento padrão. O módulo rejeita um assinante em um VPLMN não preferido uma vez. Se o assinante retornar, o módulo encaminha a ULR 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 chega de um MME em `epc.mnc003.mcc310.3gppnetwork.org`, que não está na lista preferida. O módulo a rejeita 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, o módulo encaminha a ULR normalmente.

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

Múltiplas Rejeições Antes de Permitir

Use isso quando o operador quiser 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 módulo rejeita o assinante três vezes. Ele então permite a passagem do assinante na quarta tentativa. O TTL se estende para 600 segundos para os ciclos de tentativa adicionais.

Caso de uso: Áreas urbanas densas onde existe cobertura preferida. O UE pode precisar de várias tentativas de conexão para adquiri-la.

Métricas

Rejeições

Métrica: `diameter.roaming_steering.reject.count` **Tipo**: Contador **Descrição**: O DRA incrementa esse contador cada vez que o módulo SoR rejeita uma ULR. **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**: O DRA incrementa esse contador quando o módulo permite uma ULR passar. **Rótulos**:

- `origin_realm` - Origin-Realm do VPLMN de origem

- `reason` - Por que o módulo permitiu a solicitação: `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**: O DRA incrementa esse contador quando o processador retorna uma mensagem de resposta de erro para 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: O módulo rejeita repetidamente assinantes em VPLMNs não preferidos. Eles nunca se conectam com sucesso.

Possíveis causas:

- `rejection_ttl_seconds` é muito curto. O registro de rastreamento expira antes que o assinante tente novamente.
- `max_rejections` é maior do que o número de tentativas que o UE realiza antes de desistir.

Resolução:

1. Aumente `rejection_ttl_seconds` para se adequar ao tempo de tentativa do UE.

2. Confirme que `max_rejections` é um valor que o UE pode alcançar dentro de seu ciclo de tentativas.

Assinantes em VPLMNs Preferidos Sendo Rejeitados

Sintomas: O módulo rejeita ULRs de redes parceiras preferidas com 5004.

Possíveis causas:

- O Origin-Realm do VPLMN preferido não está em `preferred_vplmns`.
- A string Origin-Realm não corresponde exatamente. A correspondência é sensível a maiúsculas e minúsculas.

Resolução:

1. Verifique o Origin-Realm do par nos logs do DRA no momento da conexão.
2. Confirme que a string exata do Origin-Realm está na lista `preferred_vplmns`.

Módulo Não Tendo Efeito

Sintomas: O módulo encaminha ULRs de VPLMNs não preferidos sem rejeição.

Possíveis causas:

- `enabled` está definido como `false`.
- O processo do módulo não está em execução. Verifique o supervisor.

Resolução:

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

Referência

Códigos Diameter

Código	Nome	Descrição	Refer
5004	DIAMETER_ERROR_ROAMING_NOT_ALLOWED	O assinante não pode roaming neste VPLMN	3GPP 29.27 Seçãc

Comandos S6a

Command-Code	Nome	Descrição	Referência
316	Update-Location-Request/Answer (ULR/ULA)	O MME envia isso 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

Vínculo de Sessão Rx

Vínculo de Sessão Rx mantém cada mensagem de uma sessão Rx no PCRF que a possui. Ele existe porque o Rx é a única interface cuja solicitação de término de sessão não carrega nada para roteamento.

O Problema

3GPP TS 29.214 §5.6.5 torna `Destination-Host` **opcional** na `Session-Termination-Request` Rx. Ele define **nenhum** `Subscription-Id` e **nenhum** `Framed-IP-Address` na mensagem. Este segundo ponto é estrutural. Esses AVPs não estão apenas ausentes na prática. Eles não estão na definição da mensagem, então nenhum AF conforme pode fornecê-los.

Se o STR carrega um `Destination-Host` depende, portanto, do AF. Algumas implementações de AF não enviam um. Uma captura representativa mostra esses AVPs:

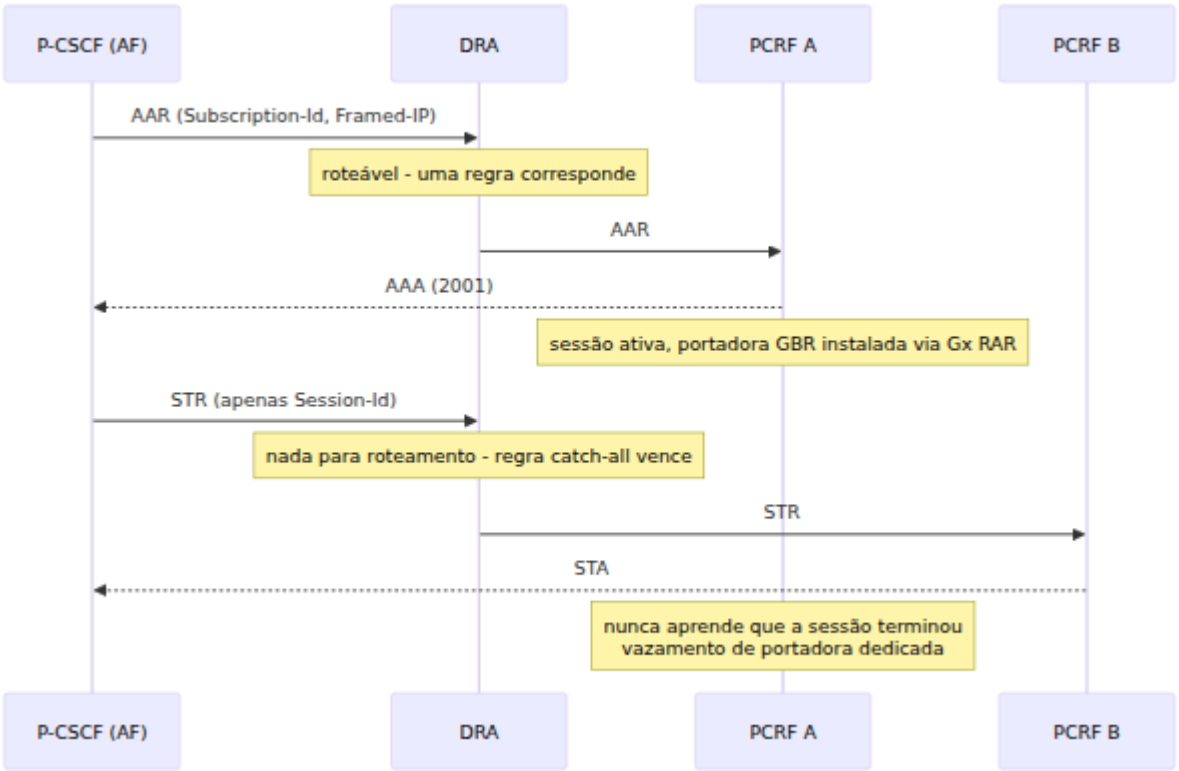
`Session-Id`, `Origin-Host`, `Origin-Realm`, `Destination-Realm`, `Auth-Application-Id`, `Vendor-Specific-Application-Id`, `AF-Application-Identifier`, `Termination-Cause`.

O `Session-Id` é o próprio identificador opaco do AF. De acordo com RFC 6733 §8.8, ele é globalmente único e estável durante a vida da sessão. Ele não identifica nem o assinante nem o servidor.

Compare isso com o `AA-Request` que abriu a sessão. Essa solicitação carrega `Subscription-Id` e `Framed-IP-Address`, então uma regra de Roteamento Avançado pode roteá-la. Compare também com **Gx**. O ABNF do `CC-Request` (TS 29.212 §5.6.2) inclui `[ Destination-Host ]`, `*[ Subscription-Id ]` **e** `[ Framed-IP-Address ]`. É exatamente por isso que o Gx não precisa de um equivalente a este módulo.

TS 29.213 §7.3.5 coloca o ônus sobre o DRA, independentemente. Um cliente "deve ser capaz de enviar cada mensagem de uma sessão para o DRA". Um desvio é apenas algo que "pode ser configurado".

Com um único PCRF, essa assimetria é invisível. Com mais de um PCRF atrás do DRA, não é:



Essa falha é silenciosa. RFC 6733 §8.4.2 não prescreve nenhum código de resultado para um STR que nomeia uma sessão desconhecida. A máquina de estados §8.1 lida com "STR recebido" de *qualquer* estado com "enviar STA, limpar". Um PCRF que responde `2001` para uma sessão que nunca teve é, portanto, conforme, e implementações reais fazem isso. Não espere que códigos de resultado revelem isso. (`5002 DIAMETER_UNKNOWN_SESSION_ID` é prescrito para o **ASA**, §8.5.2, não para o STA.)

O mesmo se aplica, menos visivelmente, a um *pool* de PCRFs intercambiáveis. Uma regra que roteia para `peers: [a, b]` seleciona por mensagem. Um AAR e seu STR podem, portanto, cair em membros diferentes do mesmo pool.

Quando Você Precisa

- Migrando entre PCRFs ou executando dois em paralelo para um teste por assinante.
- Qualquer regra de Roteamento Avançado que envia Rx para mais de um par.

Você **não** precisa disso com um único PCRF. Ele está desativado por padrão.

Como Funciona

Mensagem	Ação
`AA-Answer` (265, `Result-Code` 2001)	Vincular `Session-Id` → o PCRF que respondeu
`AA-Request` (265)	Roteie para o PCRF vinculado
`Session-Termination-Request` (275)	Roteie para o PCRF vinculado
`Session-Termination-Answer` (275)	Libere o vínculo, **apenas** se veio do PCRF vinculado e seu `Result-Code` é final (2xxx ou 5xxx)

Vínculos Seguem Fatos, Não Intenções

O DRA vincula na **resposta**, nunca na solicitação. Uma solicitação apenas diz onde o DRA *tentou* enviar algo. A resposta prova que uma sessão existe naquele nó. Isso tem três consequências:

- Um AAR respondido com um erro, ou nunca respondido, não deixa vínculo.
- O DRA libera uma sessão quando seu STA chega, não quando roteia o STR. RFC 6733 §5.5.4 sanciona explicitamente a retransmissão após uma falha. Uma liberação antecipada redirecionaria a nova tentativa, a própria falha que este módulo previne.
- Uma solicitação nunca reescreve um vínculo. Uma interrupção transitória do par, portanto, não pode migrar silenciosamente uma sessão ativa para um PCRF diferente.

A liberação requer **ambas** as condições. O STA deve vir do par que possui o vínculo, e seu `Result-Code` deve ser final. RFC 6733 §8.4.2 faz com que o servidor libere seus recursos quando envia o STA, então 2xxx e 5xxx são

igualmente finais. Um 3xxx/4xxx é transitório e o AF pode retransmitir, então o vínculo sobrevive para a nova tentativa.

A condição de propriedade não é uma limpeza defensiva. Sem ela, o módulo falha no cenário exato para o qual foi construído. O proprietário se torna inacessível. O STR cai para outro PCRF. Esse PCRF responde `2001` para uma sessão que nunca teve. O DRA então destrói o vínculo com o proprietário enquanto sua portadora ainda está ativa. Isso deixa você em uma situação pior do que sem o módulo, porque a desmontagem agora *parece* fixada. A condição de propriedade é o espelho de `insert_new` no lado do vínculo. Um segundo PCRF não pode roubar um vínculo nem destruí-lo.

O Que Não Irá Substituir

O módulo apenas redireciona o roteamento longe de uma decisão de retransmissão. Tudo o mais é um resultado deliberado, e o módulo o deixa em paz:

Resultado	Fonte	Honrado
`:discard`	uma regra `route: :drop`	sim
`{:answer_message, code}`	uma regra `route: {:error, code}`	sim
`{:answer_message, 3002}`	Roteamento Avançado falhando fechado sem par conectado	sim
`Destination-Host` explícito	o AF nomeando seu servidor (RFC 6733 §6.1.5)	sim

O custo é que o DRA não pode desmontar uma sessão enquanto cada par que suas regras apontam estiver inacessível. Este é o comércio correto. Um operador que esvazia um nó, ou falha fechado intencionalmente, não pede para ser contornado.

Ordenação

Esta é a única etapa que roda **depois** do Roteamento Avançado. Ela precisa rodar lá. Um STR só pode corresponder a uma regra catch-all, e o vínculo deve ser capaz de superá-la.

O observador do caminho de resposta roda **antes** do Transformação Avançada. A Transformação Avançada pode reescrever ou remover `Session-Id` e `Result-Code` nas respostas. O vínculo deve se basear no `Session-Id` que o AF realmente usou. O módulo é deliberadamente insensível ao ocultamento de topologia. Ele lê o par que responde de suas capacidades Diameter, não de um AVP `Origin-Host`. Uma reescrita de AVP, portanto, não pode enganá-lo.

Configuração

```
config :dra,
  module_rx_session_binding: %{
    enabled: true,
    children: [DRA.Router.Modules.RxSessionBinding],
    ttl_seconds: 43_200
  }
```

Chave	Propósito
`enabled`	Desativado por padrão. Independente de `module_subscriber_lookup`. Os dois resolvem problemas diferentes, e você pode querer apenas um deles.
`ttl_seconds`	Protetor contra vazamentos para sessões cujo STA nunca chega. Não é um temporizador de sessão.

Defina `ttl_seconds` **acima** do `Authorization-Lifetime` do AF. O DRA então nunca desvincula uma chamada longa no meio da sessão. Um P-CSCF que anuncia `Authorization-Lifetime: 36000` precisa de mais de 10 horas. O padrão de 12 horas é adequado. Limpezas ocorrem a cada `ttl_seconds / 2` (mínimo de 60s). O DRA recupera um vínculo vazado dentro de 1,5 × TTL.

Observabilidade

Três eventos de telemetria, todos carregando `session_id`:

Evento	Metadados
`[:diameter, :rx_session_binding, :bind]`	`serving_host`
`[:diameter, :rx_session_binding, :route]`	`serving_host`
`[:diameter, :rx_session_binding, :release]`	-

Um quarto evento, `[:diameter, :rx_session_binding, :reap]`, carrega uma `count`. O DRA emite isso apenas quando a limpeza de TTL encontra vínculos expirados. Uma contagem de reap não zero significa que STAs estão desaparecendo. Alerta sobre essa contagem.

O DRA expõe uma contagem de vínculos ativos e pode listá-los como `{session_id, origin_host, inserted_at}`.

Limitações

- **Vínculos são em memória e locais ao nó.** Uma reinicialização do DRA, ou uma falha de um irmão sob o supervisor `:one_for_all`, os elimina. Sessões em andamento então voltam ao que as regras dizem para o resto da chamada. O DRA não compartilha vínculos entre DRAs. Um AF deve, portanto, alcançar o mesmo DRA para ambas as metades de uma sessão.
- **O DRA vincula o par adjacente.** No modo de retransmissão, o DRA pode anexar um Route-Record, mas não inserir um `Destination-Host` (RFC 6733 §6.1.9). O DRA, portanto, não pode fixar um PCRF que está atrás de outro agente. Apenas esse agente pode.
- **Uma AA-Answer tardia pode ressuscitar uma sessão liberada.** Se o DRA processar a resposta de uma atualização de AAR após o STA, `insert_new` recria a linha. A sessão acabou, então esta é uma linha vazada, não um redirecionamento incorreto. Identificadores de sessão não

são reutilizados (RFC 6733 §8.8). O valor de `ttl_seconds` limita a linha, e ela aparece no contador `:reap`.

- **O DRA não pode liberar uma sessão enquanto o Roteamento Avançado falha fechado** para Rx. O resultado `{:answer_message, 3002}` é indistinguível de um `route: {:error, 3002}` deliberado de um operador. Mantenha todos os PCRFs que possuem sessões ativas na lista de pares da regra. Esse caso, então, não surge.
- **`children` é obrigatório na configuração.** O `DRA.Supervisor` inicia os processos de um módulo apenas quando a configuração contém uma lista `children`. Com `enabled: true` e sem `children`, o GenServer nunca inicia. Cada ponto de entrada degrada para "sem vínculo", e nada o avisa.
- **O DRA vincula apenas a direção AF → PCRF.** `RAR` e `ASR` iniciados pelo PCRF carregam um `Destination-Host` obrigatório (RFC 6733 §8.3.1, §8.5.1) e roteiam com base nisso. Eles não precisam de vínculo.
- **O DRA vincula apenas o id da aplicação Rx Rel-7+ (16777236).** Ele não vincula sessões no id Rx/Gq legado Rel-6 (16777229).
- **O DRA emite telemetria, mas não a exporta.** Os quatro eventos acima não têm definições de `Telemetry.Metrics`, então nada chega ao Prometheus hoje. Esta é a mesma lacuna que os eventos `subscriber_lookup` pré-existentes têm. Use a contagem de vínculos ativos de um shell remoto até que isso seja conectado.

Multihoming SCTP

Multihoming SCTP fornece redundância de rede para transporte Diameter. Ele permite que cada ponto final se vincule a vários endereços IP dentro de uma única associação. Se o caminho de rede primário falhar, o SCTP automaticamente muda para um caminho alternativo. Isso não interrompe a sessão Diameter.

Por Que o Multihoming É Importante

RFC 6733 define o Diameter sobre SCTP. O comportamento de multihoming descrito aqui é uma capacidade nativa do **SCTP** em si. RFC 4960 especifica isso. O TCP vincula uma associação a um único par de IPs de origem/destino. O SCTP permite que uma associação abranja vários endereços IP de cada lado. Isso proporciona resiliência na camada de transporte do OmniDRA contra a perda de um caminho de rede individual, NIC, switch ou segmento roteado. A conexão Diameter não é encerrada ou restabelecida.

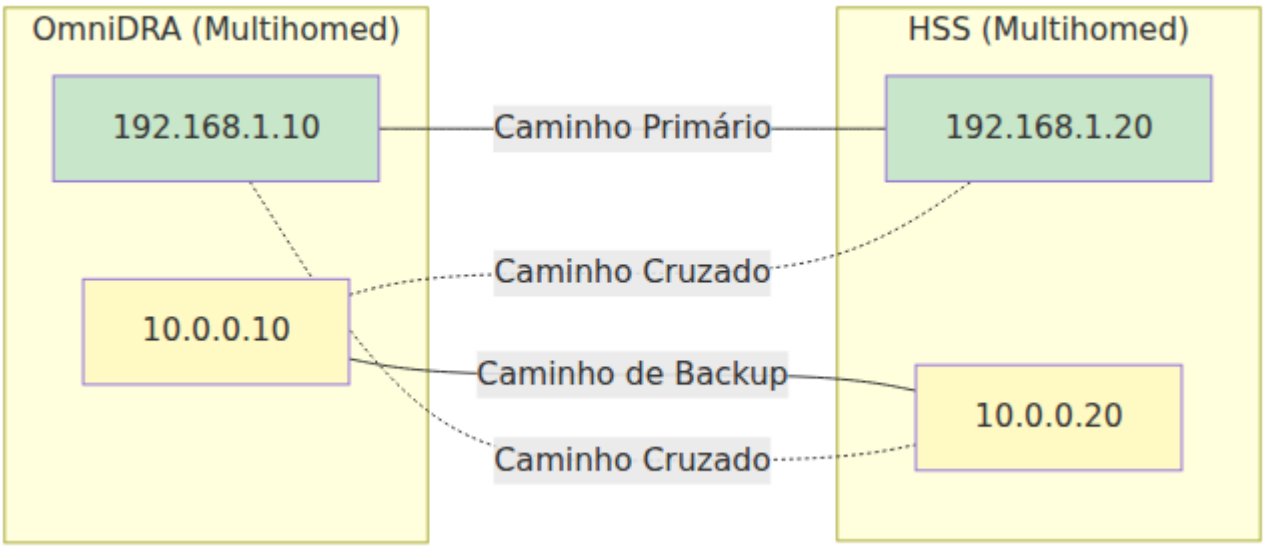
Características principais:

- **Heartbeats** do SCTP monitoram todos os caminhos de rede na associação continuamente.
- **Failover automático** ocorre no nível do kernel se o caminho primário se tornar inacessível.
- A troca de caminho não interrompe a sessão Diameter. A associação e seu estado Diameter persistem.
- O kernel do sistema operacional seleciona o caminho automaticamente.

Como Funciona

OmniDRA e um par estabelecem uma associação multihomed. Cada ponto final anuncia vários endereços IP. O kernel seleciona um caminho primário para transmissão de dados. Ele trata os caminhos restantes como rotas de espera.

Se o caminho primário falhar, o kernel promove automaticamente uma rota de espera.



No diagrama acima, o **Caminho Primário** (verde) transporta tráfego em condições normais. O **Caminho de Backup** (amarelo) e os **Caminhos Cruzados** (tracejados) fazem parte da mesma associação SCTP. Eles permanecem disponíveis para failover imediato.

Configuração

Configure o multihoming em `config/runtime.exs` em dois lugares. O primeiro são os endereços de escuta locais aos quais o OmniDRA se vincula. O segundo são os endereços remotos de cada par multihomed. Consulte a Referência de Configuração para a estrutura completa de configuração.

Endereços de Escuta DRA

O campo `listen_ip` controla quais endereços IP locais o OmniDRA se vincula para conexões de entrada. Ele aceita uma única string de IP (compatível com versões anteriores) ou uma lista de strings de IP. Use uma lista para habilitar o multihoming.

```
%{
  # 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
}
```

Parâmetro	Tipo	Requerido	Padrão	Descrição
`listen_ip`	String ou Lista de Strings	Sim	-	Endereço(s) IP local(is) ao qual o OmniDRA se vincula para conexões de entrada. Uma única string vincula um endereço. Uma lista de strings habilita o multihoming SCTP em todos os endereços listados. Para **transporte TCP**, o OmniDRA usa apenas o primeiro IP da lista. Para **transporte SCTP**, o OmniDRA vincula todos os IPs listados à associação.
`listen_port`	Inteiro	Sim	-	Porta local para conexões Diameter de entrada. A porta padrão é 3868 conforme RFC 6733.

Compatibilidade com versões anteriores: Uma única string de IP continua totalmente suportada. As configurações existentes de um único IP continuam a funcionar inalteradas. Para optar pelo multihoming, forneça uma lista.

Configuração do Par

Cada entrada de par declara o(s) endereço(s) remoto(s) do par. O campo `ip` contém o endereço primário e é sempre necessário. O campo opcional `additional_ips` lista todos os endereços para a associação multihomed.

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

Parâmetro	Tipo	Requerido	Padrão	Descrição
`host`	String	Sim	-	Identidade Diameter do pa (FQDN). Deve corresponder exatamente à identidade configurada do par.
`realm`	String	Sim	-	Reino Diameter do par. O DRA utiliza para decisões de roteamento.
`ip`	String	Sim	-	Endereço IP primário do par Sempre necessário, incluindo para compatibilidad com versões anteriores de configurações um único IP.
`additional_ips`	Lista de Strings	Não	-	Todos os endereços IP para a associação SCT multihomed. Este campo é opcional. Se você omitir, o OmniDRA usar

Parâmetro	Tipo	Requerido	Padrão	Descrição
				apenas `ip`. Par SCTP, inclua o primário nesta lista. Para TCP, OmniDRA ignor este campo, po o TCP não suporta multihoming.
`port`	Inteiro	Sim	-	Porta Diameter do par. A porta padrão é 3868 conforme RFC 6733.
`transport`	Átomo	Sim	-	Protocolo de transporte para a conexão do par: `:diameter_sct` ou `:diameter_tcp` O multihoming requer `:diameter_sct`
`initiate_connection`	Booleano	Sim	-	Quando `true`, OmniDRA inicia a conexão com par. Quando `false`, o OmniDRA aguarda o par conectar.

Compatibilidade com versões anteriores: O campo `ip` sozinho descreve um par de um único IP. Se você omitir `additional_ips`, o par permanecerá com um único IP. Portanto, as configurações existentes de pares não precisam de alterações.

Exemplo Completo

A configuração a seguir vincula o OmniDRA a dois endereços locais. Ele se conecta a um HSS multihomed e a um MME de um único IP.

```
config :dra,
  diameter: %{
    service_name: :omnitouch_dra,
    listen_ip: ["192.168.1.10", "10.0.0.10"],  # DRA Multihomed
    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 Multihomed
      %{
        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,
        initiate_connection: true
      },
      # MME de um único IP (compatível com versões anteriores)
      %{
        host: "mme01.example.com",
        realm: "example.com",
        ip: "192.168.1.30",
        port: 3868,
        transport: :diameter_sctp,
        initiate_connection: false
      }
    ]
  }
```

As tabelas acima documentam `listen_ip`, `listen_port` e todos os parâmetros do par. Os parâmetros restantes de nível superior controlam a identidade Diameter do OmniDRA e o comportamento de seleção de pares. A Referência de Configuração os documenta:

Parâmetro	Tipo	Requerido	Padrão	
`service_name`	Átomo	Sim	-	
`host`	String	Sim	-	
`realm`	String	Sim	-	
`product_name`	String	Sim	-	
`vendor_id`	Inteiro	Sim	-	
`request_timeout`	Inteiro	Não	`5000`	

Parâmetro	Tipo	Requerido	Padrão	
`peer_selection_algorithm`	Átomo	Sim	-	
`allow_undefined_peers_to_connect`	Booleano	Sim	`false`	
`peers`	Lista	Sim	-	

Buffers de Socket SCTP

O OmniDRA define um buffer de recebimento e envio de socket explícito em cada transporte SCTP. Ele usa a chave de nível superior `diameter_socket_buffer_size` (padrão 4 MB). Ao contrário de `listen_ip` e `peers`, esta chave está no **nível superior** da configuração da aplicação `:dra`. Ela **não** está dentro do mapa `diameter:`.

Sem um buffer explícito, o socket SCTP herda um pequeno `SO_RCVBUF` padrão. Isso fixa a janela de recebimento SCTP em aproximadamente um MTU e causa paradas de janela zero sob carga. O kernel do Linux limita `SO_RCVBUF` a `net.core.rmem_max`. Ele limita o buffer de envio a `net.core.wmem_max`. Para usar tamanhos de buffer maiores do que esse limite, aumente esses sysctls no host.

```
config :dra,
  # Chave de nível superior - NÃO dentro do mapa diameter:.
Aplicada a cada transporte SCTP.
  diameter_socket_buffer_size: 4 * 1024 * 1024,  # 4 MB
(recbuf/sndbuf)
  diameter: %{
    # ... configuração de serviço e pares ...
  }
```

Parâmetro	Tipo	Requerido	Padrão	
`diameter_socket_buffer_size`	Inteiro	Não	`4194304` (4 MB)	Tamanh envio dc transpor quanto `:diamet` Valores `net.cor` exigem do kerne

Requisitos

O multihoming SCTP precisa dessas condições em ambos os pontos finais:

- O **módulo do kernel SCTP** deve estar carregado no host (o pacote `lksctp-tools` no Linux).
- Todos os endereços IP configurados devem ser roteáveis para e a partir do par.

- As regras de firewall devem permitir tráfego SCTP em cada endereço IP configurado.
- Configure ambos os pontos finais para multihoming para alcançar total redundância. Um OmniDRA multihomed conectado a um par de um único IP ganha redundância de caminho apenas do lado do OmniDRA.

Limitações

- **O TCP não suporta multihoming.** Com o transporte `:diameter_tcp`, o OmniDRA usa apenas o IP primário. `listen_ip` usa o primeiro IP da lista, e o OmniDRA ignora `additional_ips` do par.
- **O tempo de failover de caminho** depende dos parâmetros SCTP do kernel, como o intervalo de heartbeat e os limiares de retransmissão. Você ajusta isso fora do OmniDRA no nível do sistema operacional.

Referências

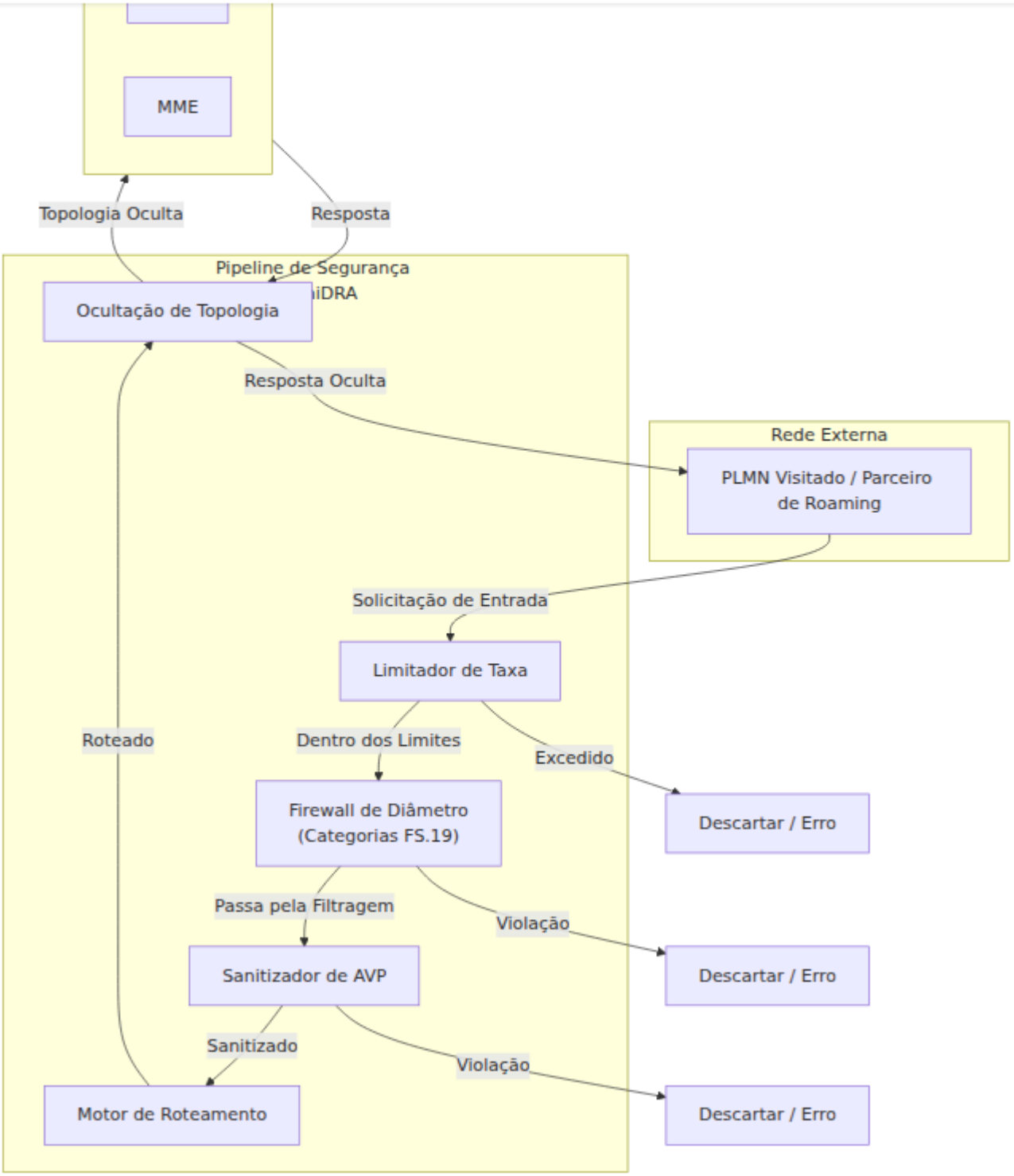
- RFC 4960 - Stream Control Transmission Protocol (SCTP)
- RFC 6733 - Diameter Base Protocol
- Referência de Configuração

Segurança do Diâmetro

OmniDRA fornece um conjunto abrangente de módulos de segurança do Diâmetro. Eles estão 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.

Você configura cada módulo de forma independente. Você pode habilitar ou desabilitar cada módulo sem afetar os outros. Cada módulo 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. Se uma etapa rejeitar uma mensagem, o DRA nunca a encaminha para as etapas posteriores.

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

Ações

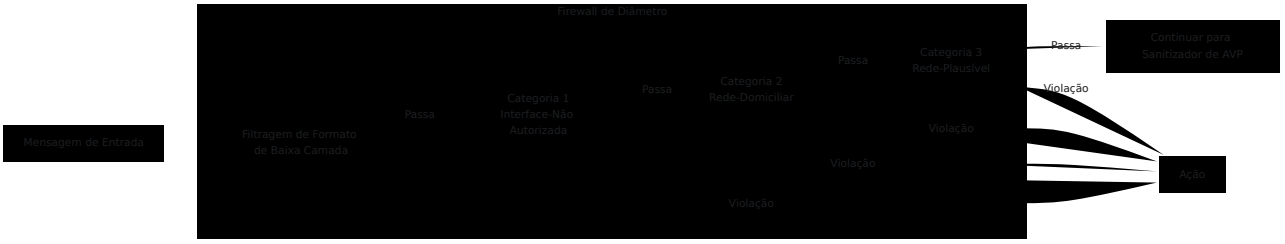
Todos os módulos de segurança suportam ações configuráveis quando detectam uma violação. Você configura as ações por módulo e por categoria para o Firewall de Diâmetro.

Ação	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	Nenhuma resposta enviada
`:log_only`	Registrar a violação, mas permitir a mensagem passar	Mensagem continua

---

Firewall de Diâmetro

O Firewall de Diâmetro implementa as quatro camadas de filtragem definidas na **Seção 3.3 do GSMA FS.19**. Ele também implementa a categorização de pacotes independente de protocolo da **Seção 7 do FS.21**. O firewall avalia mensagens através de cada camada em 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 spoofing. Não precisa entender a semântica de 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	Confirma 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 uma mensagem de resposta que contém um AVP Destination-Host. Esta é uma violação de protocolo que pode indicar evasão de filtro.
Validação de AVPs Obrigatórios	Seção 3.3.5	Confirma que os AVPs obrigatórios estão presentes para cada código de comando. Por exemplo, um 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 aceita apenas IDs de Aplicação do Diâmetro e Códigos de Comando autorizados em cada interface de interconexão. Isso previne o acesso externo a interfaces que são apenas internas. Por exemplo, bloqueia comandos Sh sobre uma interface S6a. Também impõe que um parceiro de roaming envie apenas os tipos de mensagem que seu acordo de roaming cobre.

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

Você pode configurar listas brancas globalmente, o que as aplica a todos os pares, ou por par. Uma lista branca por par permite 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	Solicitação/Resposta de Atualização de Localização	MME → HSS	3GPP TS 29.272 §7.2.3
317	Solicitação/Resposta de Cancelamento de Localização	HSS → MME	3GPP TS 29.272 §7.2.7
318	Solicitação/Resposta de Informação de Autenticação	MME → HSS	3GPP TS 29.272 §7.2.5
319	Solicitação/Resposta de Inserção de Dados do Assinante	HSS → MME	3GPP TS 29.272 §7.2.9
320	Solicitação/Resposta de Exclusão de Dados do Assinante	HSS → MME	3GPP TS 29.272 §7.2.11
321	Solicitação/Resposta de Purga de UE	MME → HSS	3GPP TS 29.272 §7.2.13
323	Solicitação/Resposta de Notificação	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

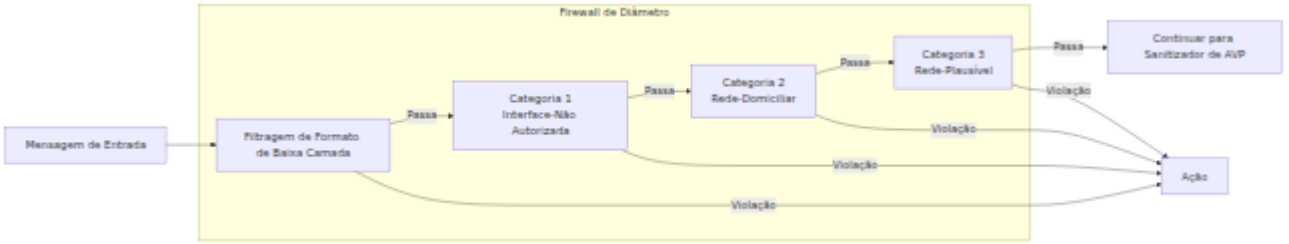
Referência GSMA: Seção 3.3.6 do FS.19, Seção 7.2.2 do FS.21

A filtragem da Categoria 2 protege os assinantes domiciliares contra mensagens que os visam a partir da interconexão. O firewall inspeciona mensagens em códigos de comando protegidos para identidade do assinante (IMSI no AVP User-Name, MSISDN no AVP 701). Se a identidade corresponder a um prefixo de assinante domiciliar, o firewall rejeita a mensagem. O tráfego legítimo para assinantes domiciliares se origina de dentro da rede domiciliar, não de pares externos.

Isso previne ataques onde uma entidade externa envia atualizações de localização, solicitações de dados do assinante ou consultas de autenticação que visam os próprios assinantes da operadora.

```
# 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 de 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. Ela rastreia a última rede vista para cada assinante (IMSI). Quando uma Solicitação de Atualização de Localização chega de uma rede visitada diferente da anterior, o firewall compara o tempo decorrido com um limite configurável. Uma mudança de domínio que ocorre mais rápido do que o fisicamente possível indica um ataque potencial, como uma atualização de localização falsificada.

Este módulo mantém rastreamento por estado por IMSI em uma tabela ETS. Ele limpa entradas obsoletas automaticamente. Remove entradas com mais de 24 horas periodicamente.

Verificações realizadas:

Verificação	Referência FS.19	Descrição
Verificação de Localização Anterior	Seção 3.3.7.1	Compara o Origin-Realm do ULR atual com o último Origin-Realm visto para esse IMSI
Verificação de Velocidade/Tempo	Seção 3.3.7.2	Marca uma mudança de domínio que ocorre 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,
  # Segundos mínimos entre ULRs de diferentes domínios para o
mesmo IMSI
  min_time_between_updates_seconds: 2
}
```

Configuração

Habilite o Firewall de Diâmetro no nível superior. Configure cada camada de filtragem de forma independente. Combine os trechos de configuração de cada seção de categoria acima 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	
`enabled`	Booleano	Sim	`false`	Habilitar ou d Quando `fals`
`children`	Lista	Não	`[]`	Processos qu habilitado. Pa `[DRA.Router` O rastreame requer isso.
`home_imsi_prefixes`	Lista	Não	`[]`	Lista de strin assinantes d Exemplo: `["`
`home_msisdn_prefixes`	Lista	Não	`[]`	Lista de strin assinantes d Exemplo: `["`

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	Obrigat►►rio	Padrão	Des
`enabled`	Booleano	Não	`true`	Habilitar ou desabil Categoria 1.
`action`	Ação	Não	`{:error, 3007}`	Ação a ser tomada detecta uma violaç padrão responde co DIAMETER_APPLICA
`whitelists`	Mapa	Sim	-	Mapa de nome de h para combinações p Aplicação / Código chave `:all` fornece padrão. Entradas es têm prioridade.

Parâmetros da Categoria 2

Parâmetro	Tipo	Obrigatório	Padrão	Descr
`enabled`	Booleano	Não	`true`	Habilita desabili filtrager Categor
`action`	Ação	Não	`{:error, 3002}`	Ação a tomada quando mensag visa um assinan domicili partir d intercor
`protected_command_codes`	Lista	Não	`[]`	Códigos comanc acionan verifica de identida do assir domicili Normalı os códi de com S6a que transpo a identi do assir

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]`	O firewall detectou e bloqueou uma violação de formato de baixa camada
`[:diameter, :firewall, :category_1, :block]`	O firewall detectou e bloqueou uma violação da Categoria 1
`[:diameter, :firewall, :category_2, :block]`	O firewall detectou e bloqueou uma violação da Categoria 2
`[:diameter, :firewall, :category_3, :block]`	O firewall detectou e bloqueou uma violação da Categoria 3
`[:diameter, :firewall, :pass, :count]`	A mensagem passou por todas as verificações do firewall

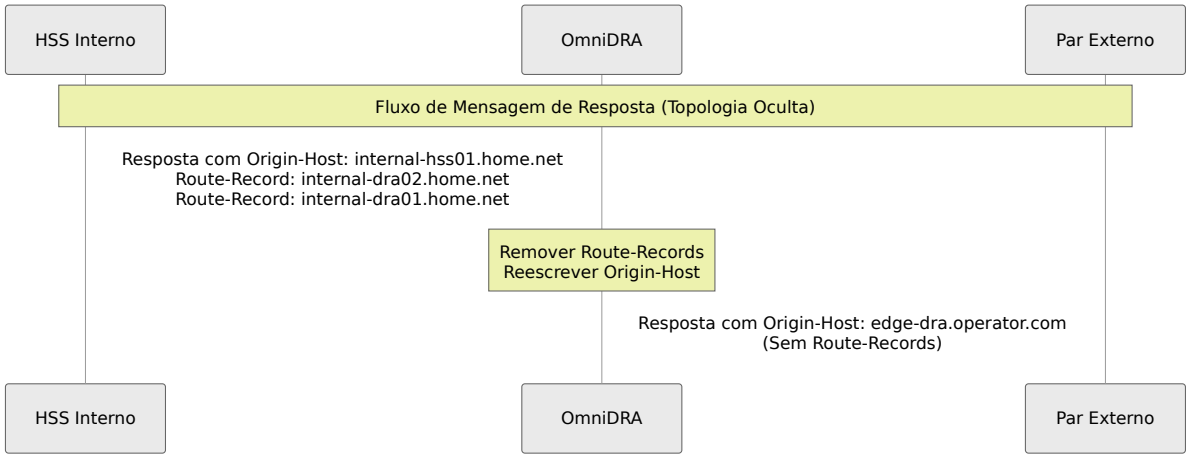
Todos os eventos incluem os seguintes metadados: `origin_host`, `application_id`, `command_code`, `application_name`, `command_name` e `reason`.

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 previne a exposição da topologia interna da rede a pares externos. Quando mensagens do Diâmetro atravessam múltiplos nós internos, elas acumulam AVPs Route-Record. Elas também carregam valores Origin-Host/Origin-Realm que revelam nomes de host internos, estrutura de rede e contagens de nós. Um atacante pode usar essas informações para mapear a rede interna para ataques direcionados.

A ocultação de topologia opera no **caminho de saída**. Ela é executada após o DRA tomar decisões de roteamento e antes de encaminhar mensagens 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 de domínio interna
Controle por Par	-	Aplicar ocultação de topologia seletivamente, 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 de domínio interna
  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 que o DRA as encaminhe 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 que o módulo considera externos. O módulo aplica a ocultação de topologia 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`). O módulo usa um valor de string como está.

Eventos de Telemetria

Evento	Descrição
`[:diameter, :topology_hiding, :route_record, :stripped]`	O módulo removeu AVPs Route-Record. A medição inclui a `count` de AVPs removidos.
`[:diameter, :topology_hiding, :origin_host, :rewritten]`	O módulo reescreveu o AVP Origin-Host
`[:diameter, :topology_hiding, :origin_realm, :rewritten]`	O módulo reescreveu o AVP Origin-Realm

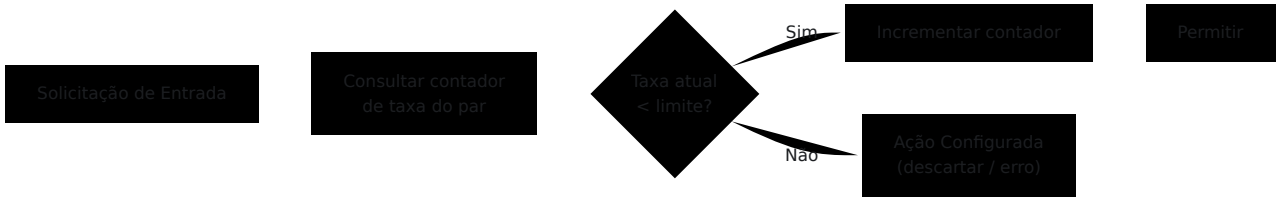
Limitador de Taxa

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

O Limitador de Taxa impõe limites de taxa de mensagens por par. Ele protege contra ataques volumétricos e inundações de mensagens. Opera como a primeira verificação de segurança no pipeline, antes de qualquer análise de mensagem ou inspeção de conteúdo. Isso elimina carga excessiva o mais cedo possível.

O Limitador de Taxa rastreia limites de taxa por par com 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`
`children`	Lista	Não	`[]`
`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 o padrão	Máximo de solicitações por segundo para este par específico.
`action`	Ação	Não	Herda o padrão	Ação quando este par excede seu limite de taxa.

Eventos de Telemetria

Evento	Descrição
`[:diameter, :rate_limiter, :throttled]`	O módulo limitou a taxa de uma mensagem. As medições incluem `current_rate` e `limit`.
`[:diameter, :rate_limiter, :allowed]`	Uma mensagem estava dentro dos limites de taxa.

Sanitizador de AVP

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

O Sanitizador de AVP valida e sanitiza AVPs do Diâmetro na fronteira de interconexão. Ele aborda as recomendações da Seção 3.6 do FS.21. Essas

recomendações lidam 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 incluídos na lista branca. Previne a injeção de AVPs proprietários que podem acionar comportamentos não intencionais 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 que usam 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 Sanit
`action`	Ação	Não	`{:error, 3002}`	Ação quan detec de pr aninh remo forne os AV não b mens
`allowed_vendor_ids`	Lista	Não	`[0, 10415, 13019, 5535]`	Lista forne perm interc módu de ou forne
`strip_unknown_vendor_avps`	Booleano	Não	`true`	Habil de AV forne estão `allo`
`max_avp_nesting_depth`	Inteiro	Não	`10`	Profu máxi aninh agrup

Parâmetro	Tipo	Obrigatório	Padrão	
				confi a um que e profu

IDs de Fornecedor Comuns:

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

Eventos de Telemetria

Evento	Descrição
`[:diameter, :avp_sanitizer, :unknown_vendor, :stripped]`	O módulo removeu um ou mais AVPs de fornecedores desconhecidos
`[:diameter, :avp_sanitizer, :nesting_depth, :violation]`	Uma mensagem excedeu a profundidade máxima de aninhamento de AVP
`[:diameter, :avp_sanitizer, :pass, :count]`	A mensagem passou por todas as verificações de sanitização

Recomendações de Implantação

Ordem Recomendada de Habilitação

Quando você implanta os módulos de segurança pela primeira vez, habilite-os um de cada vez. Isso evita interrupções no tráfego ao vivo:

1. **Limitador de Taxa** - Comece com limites generosos e monitore padrões de tráfego. Aperte os limites após entender as taxas básicas.
2. **Sanitizador de AVP** - Baixo risco de falsos positivos. Habilite as verificações de remoção e aninhamento.
3. **Firewall de Diâmetro (Categoria 1)** - Defina listas brancas com base em acordos de roaming. Comece com combinações conhecidas de ID de Aplicação / Código de Comando.
4. **Firewall de Diâmetro (Baixa Camada)** - Habilite as verificações de conformidade de protocolo.
5. **Firewall de Diâmetro (Categoria 2)** - Configure os prefixos de 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-o com a ação `:log_only`. Observe padrões de localização antes de impor.

Defesa em Profundidade

Esses módulos implementam o princípio de **defesa em profundidade** da 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	Sanitizador de AVP
Alvo de Assinante Domiciliar	Categoria 2	Categoria 1 (restrição de interface)
Spoofing de Localização	Categoria 3	Categoria 2 (verificação de assinante domiciliar)
Descoberta de Topologia	Ocultação de Topologia	-
Injeção de AVP	Sanitizador de AVP	Filtragem de Baixa Camada
Evasão de Protocolo (Duplicação de AVP)	Filtragem de Baixa Camada	Sanitizador 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	-
Sanitização 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

Roteamento de Diâmetro Padrão

Roteamento de Diâmetro Padrão é o comportamento padrão de encaminhamento de solicitações do DRA. O DRA o aplica quando nenhuma regra de Roteamento Avançado corresponde a uma solicitação recebida. Ele segue o mecanismo de roteamento baseado em prioridade definido no Protocolo Base de Diâmetro (RFC 6733). Ele seleciona um destino a partir do conteúdo do AVP e do conjunto de pares conectados do DRA.

Quando o Roteamento Padrão se Aplica

O DRA sempre tenta o roteamento padrão como seu comportamento base. Se o módulo de Roteamento Avançado estiver habilitado e uma de suas regras corresponder à solicitação, essa regra decide o resultado em vez disso. A regra pode roteá-la para pares nomeados, roteá-la por `Destination-Host`, gerar uma resposta de erro ou descartar a solicitação. Quando o Roteamento Avançado está desativado, ou nenhuma regra corresponde, a solicitação cai na precedência de roteamento padrão descrita aqui.

Uma exceção estreita se aplica após ambas as etapas. Quando habilitado, o Rx Session Binding redireciona uma solicitação Rx para o PCRF que mantém sua sessão. Ele apenas substitui uma decisão de retransmissão. Um resultado `:drop`, uma resposta de erro e um `Destination-Host` explícito permanecem.

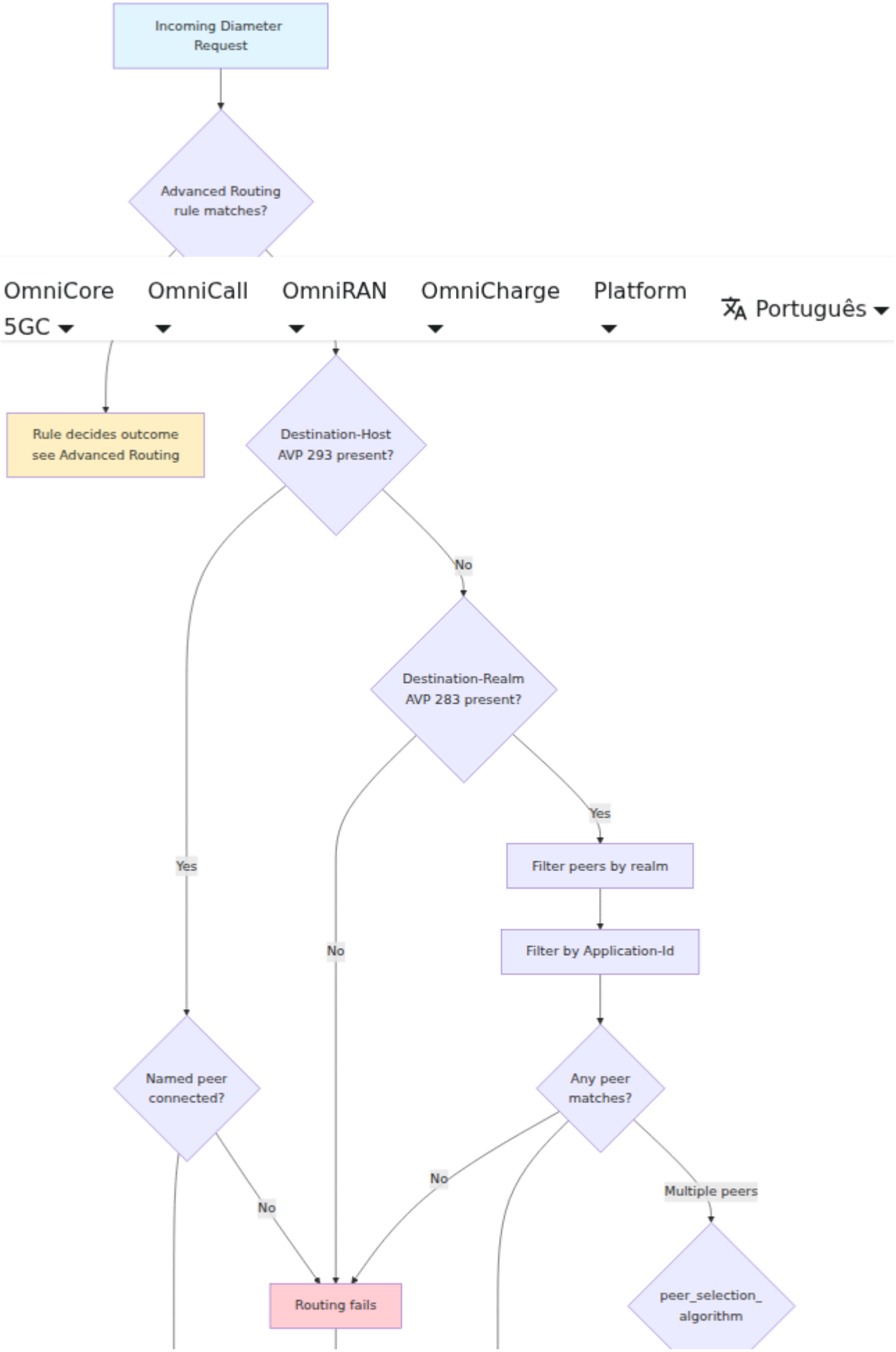
O Roteamento Avançado descreve as substituições baseadas em conteúdo. Essas substituições sobrepõem o roteamento com base em valores de AVP, como padrões de IMSI, tradução de domínio e decisões de pares de origem. Essa página também detalha a **afinidade de sessão** através do AVP `Destination-Host`.

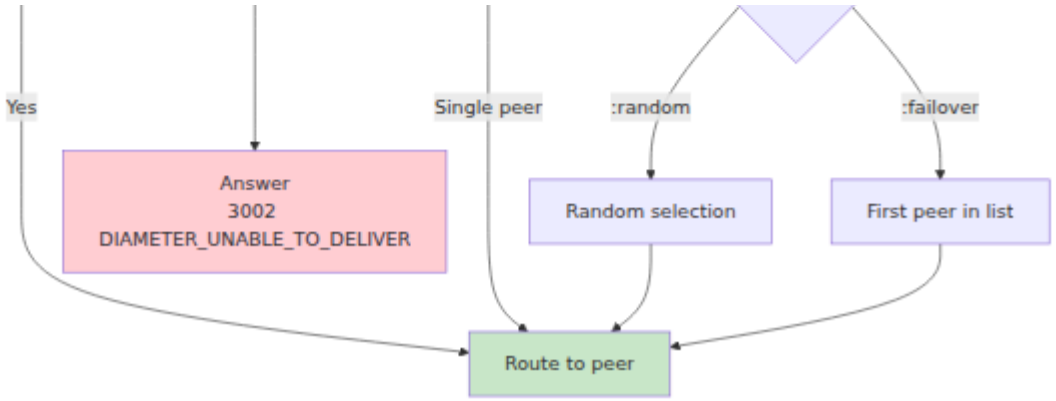
Precedência da Decisão de Roteamento

O roteamento padrão avalia cada solicitação contra uma ordem de precedência fixa. O primeiro mecanismo aplicável vence.

1. **AVP `Destination-Host` (293)** - Este é o mecanismo de maior prioridade. Se a solicitação contém um AVP `Destination-Host`, o DRA roteia diretamente para esse host. Isso fornece controle de roteamento explícito em nível de host. Se o par nomeado não estiver conectado, o roteamento falha.
2. **AVP `Destination-Realm` (283) + `Application-Id`** - Se o `Destination-Host` estiver ausente, o DRA seleciona um par conectado que anuncia suporte para o **domínio** alvo. O DRA então filtra os pares candidatos pelo **Application-Id** da mensagem. Apenas os pares que anunciam suporte para essa aplicação permanecem. Os pares anunciam esse suporte durante a Troca de Capacidades. Quando mais de um par corresponde, o algoritmo de seleção de pares configurado escolhe entre eles.
3. **Falha de retransmissão / roteamento** - Se nem um `Destination-Host` acessível nem um par correspondente de domínio mais aplicação estiver disponível, o DRA não pode entregar a solicitação. O DRA retorna uma resposta `3002 DIAMETER_UNABLE_TO_DELIVER`.

O diagrama de fluxo a seguir mostra todo o caminho de decisão. Ele inclui o ponto em que o Roteamento Avançado pode interceptar a solicitação antes que o roteamento padrão seja executado.


Incoming Diameter Request
Advanced Routing rule matches?
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Rule decides outcome see Advanced Routing
Destination-Host AVP 293 present?
No
Destination-Realm AVP 283 present?
Yes
Filter peers by realm
Filter by Application-Id
Yes
No
Named peer connected?
Any peer matches?
No
No
Multiple peers
Routing fails
peer_selection_ algorithm



Apenas pares no estado **conectado** são elegíveis em qualquer estágio. O DRA exclui automaticamente pares desconectados ou inativos da filtragem de domínio e aplicação. Uma solicitação `Destination-Host` direcionada a um par desconectado falha imediatamente.

Seleção de Pares

Quando múltiplos pares conectados correspondem aos critérios de domínio e Application-Id, a configuração global `peer_selection_algorithm` determina qual par recebe a solicitação. Esta é uma configuração de serviço em toda a extensão. Você não a configura por domínio ou por par.

```
config :dra, :service, %{
  # ... DRA identity and network settings ...
  peer_selection_algorithm: :random
}
```

Parâmetro	Tipo	Requerido	Padrão	Descriçã
`peer_selection_algorithm`	Atom	Não	`:random`	Estratégia de balanceamen de carga par quando múltiplos par conectados correspondem uma solicitaç `:random` seleciona um par aleatório conjunto correspondem `:failover` sempre seleciona o primeiro par lista configurada. usa os pares restantes apenas quan o primeiro nã está disponív

- **`:random`** - Distribui a carga uniformemente entre todos os pares correspondentes. Use isso quando os pares forem intercambiáveis e de custo igual.
- **`:failover`** - Prefere um único par primário. Ele trata a lista de pares ordenada como uma sequência de prioridade. Use isso para designs ativo/standby. Um par específico transporta tráfego enquanto estiver conectado.

Consulte Configuração para o bloco completo de configuração do serviço e definições de pares.

Quando Nenhum Par Está Disponível

A avaliação de precedência pode falhar em identificar um destino acessível. Isso acontece quando o par `Destination-Host` está desconectado, quando nenhum par conectado anuncia o domínio alvo, quando nenhum par correspondente suporta o Application-Id, ou quando a solicitação carece dos AVPs necessários para tomar uma decisão. Em cada um desses casos, o DRA não pode encaminhar a solicitação.

O DRA então retorna uma resposta com `Result-Code` definido como `3002 DIAMETER_UNABLE_TO_DELIVER`, conforme definido na RFC 6733 7.1.5. Isso informa ao par de origem que o DRA recebeu a solicitação, mas não conseguiu entregá-la a um próximo salto adequado. O originador pode então aplicar sua própria lógica de nova tentativa ou failover.

Referência

Códigos de Resultado

Código	Nome	Descrição	Referência
2001	DIAMETER_SUCCESS	O DRA completou a solicitação com sucesso	RFC 6733 §7.1.1
3002	DIAMETER_UNABLE_TO_DELIVER	O DRA não conseguiu entregar a solicitação a um próximo salto adequado. Nenhum par conectado correspondente estava disponível	RFC 6733 §7.1.3

Códigos AVP Comuns

Código	Nome do AVP	Tipo	Uso	Referência
1	User-Name	UTF8String	Identificador do assinante (IMSI em 3GPP)	RFC 6733 §8.14
264	Origin-Host	DiameterIdentity	Nome do host do par de origem	RFC 6733 §6.3
268	Result-Code	Unsigned32	Código de resultado padrão	RFC 6733 §7.1
283	Destination-Realm	DiameterIdentity	Domínio alvo	RFC 6733 §6.6
293	Destination-Host	DiameterIdentity	Host alvo (opcional)	RFC 6733 §6.5
296	Origin-Realm	DiameterIdentity	Domínio de origem	RFC 6733 §6.4
297	Experimental-Result	Grouped	Código de resultado específico do fornecedor	RFC 6733 §7.6

Códigos de Comando Comuns

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

Código	Nome do Comando	Descrição	Referência
257	CER/CEA	Capabilities-Exchange-Request/Answer	RFC 6733 §5.3
258	RAR/RAA	Re-Auth-Request/Answer	RFC 6733 §8.3
274	ASR/ASA	Abort-Session-Request/Answer	RFC 6733 §8.5
275	STR/STA	Session-Termination-Request/Answer	RFC 6733 §8.4
280	DWR/DWA	Device-Watchdog-Request/Answer	RFC 6733 §5.5
282	DPR/DPA	Disconnect-Peer-Request/Answer	RFC 6733 §5.4
316	ULR/ULA	Update-Location-Request/Answer (S6a)	3GPP TS 29.272
317	CLR/CLA	Cancel-Location-Request/Answer (S6a)	3GPP TS 29.272
318	AIR/AIA	Authentication-Information-Request/Answer (S6a)	3GPP TS 29.272
321	PUR/PUA	Purge-UE-Request/Answer (S6a)	3GPP TS 29.272

IDs de Aplicação Comuns do 3GPP

Os pares anunciam seus Application-Ids suportados durante a Troca de Capacidades. O roteamento padrão filtra os pares candidatos pelo Application-

Id da solicitação.

Application-Id	Interface	Descrição	Referência
16777251	S6a/S6d	MME/SGSN ↔ HSS autenticação e dados de assinatura	3GPP TS 29.272
16777252	S13/S13'	MME ↔ EIR verificação de identidade do equipamento	3GPP TS 29.272
16777238	Gx	PCEF ↔ PCRF controle de política e cobrança	3GPP TS 29.212
16777267	S9	PCRF de origem ↔ PCRF visitado política de roaming	3GPP TS 29.215
16777272	S6b	PDN-GW ↔ 3GPP AAA para acesso não 3GPP	3GPP TS 29.273
16777216	Cx	I-CSCF/S-CSCF ↔ HSS registro IMS	3GPP TS 29.229
16777217	Sh	AS ↔ HSS dados do usuário IMS	3GPP TS 29.329
16777255	SLg	MME/SGSN ↔ GMLC serviços de localização	3GPP TS 29.172
16777291	SLh	GMLC ↔ HSS informações do assinante de localização	3GPP TS 29.173
16777302	Sy	PCRF ↔ OCS vinculação de sessão	3GPP TS 29.219
16777310	S6m	MTC-IWF ↔ HSS/HLR para dispositivos M2M	3GPP TS 29.336

Application-Id	Interface	Descrição	Referência
16777312	S6c	SMS-SC/IP-SM-GW ↔ HSS roteamento de SMS	3GPP TS 29.338
16777345	S6t	SCEF ↔ HSS monitoramento de eventos	3GPP TS 29.336
16777236	Rx	AF ↔ PCRF autorização de mídia	3GPP TS 29.214

As aplicações de cobrança (Gy/Ro) seguem 3GPP TS 32.299.

Documentação Relacionada

- Configuração - Referência completa de configuração de serviço e pares.
- Roteamento Avançado - Regras de roteamento baseadas em conteúdo, afinidade de sessão via `Destination-Host`, e substituições baseadas em regras que têm precedência sobre o roteamento padrão.

Função de Consulta de Assinante (SLF)

A Função de Consulta de Assinante (SLF) aprende dinamicamente qual elemento de rede atende cada assinante. Ela observa o tráfego de sinalização Diameter que passa pelo DRA. Utiliza os vínculos aprendidos para rotear solicitações posteriores, como consultas de serviço de localização, diretamente para o nó de atendimento correto. Não precisa de regras de roteamento estáticas.

Isso é útil para solicitações de serviço de localização SLg/SLh. O GMLC precisa alcançar o MME que atende um determinado assinante, mas não sabe qual MME é esse.

3GPP TS 29.172 e 3GPP TS 29.173 descrevem o conceito de SLF para serviços de localização. 3GPP TS 29.272 define o registro do assinante.

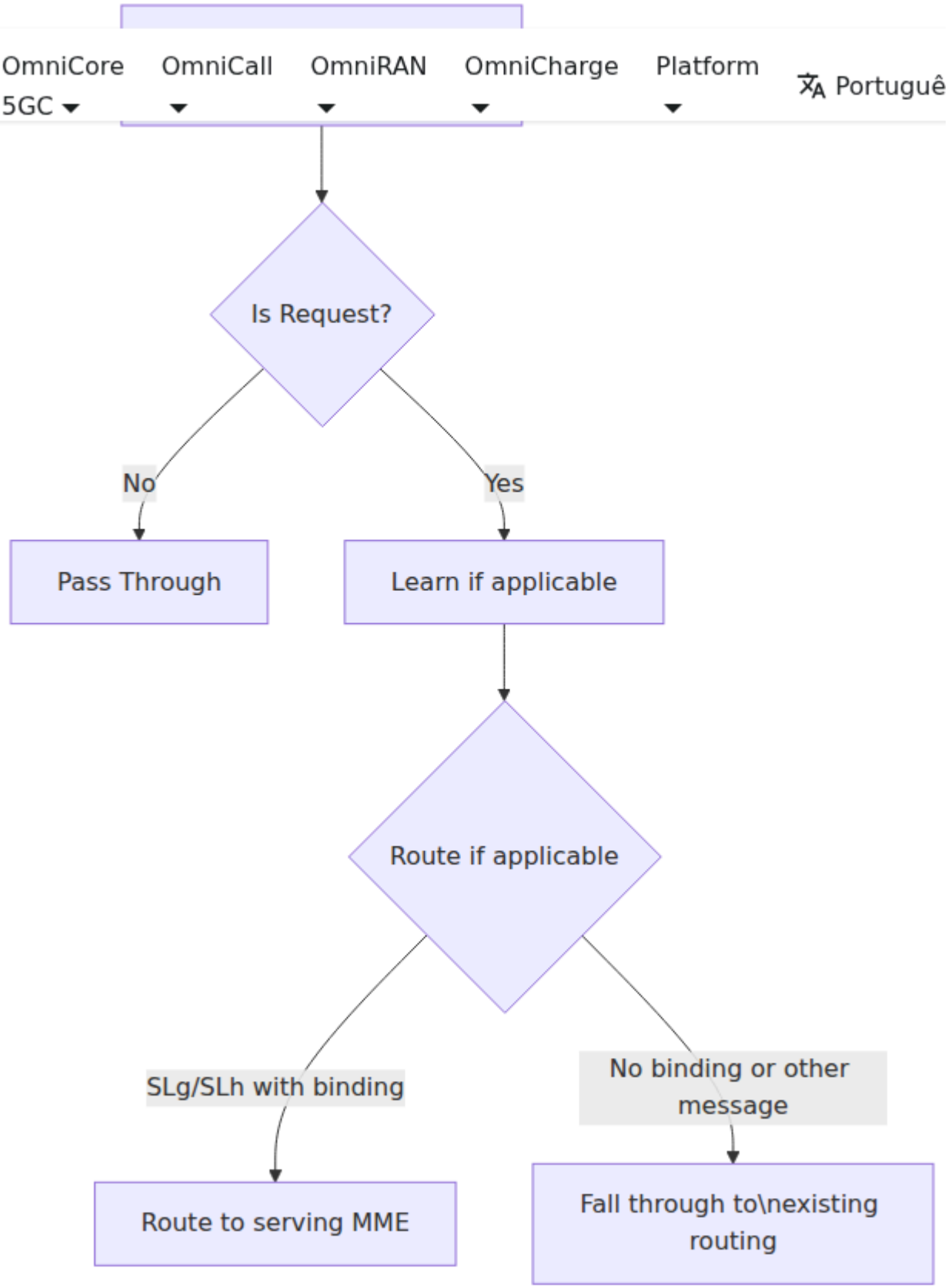
Como Funciona

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

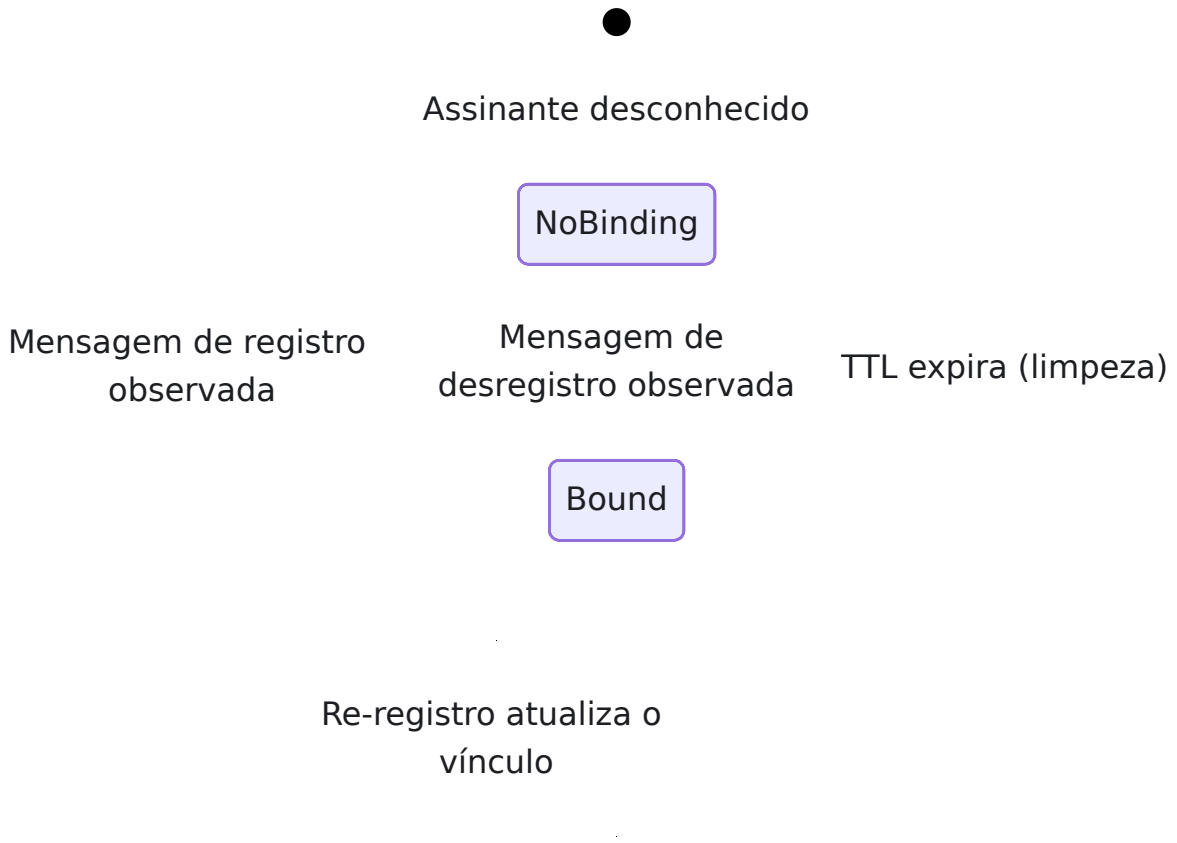
Durante o **aprendizado**, o módulo observa mensagens de registro e desregistro que fluem através do DRA. Quando um assinante se registra (por exemplo, através de um S6a Update-Location-Request), o módulo registra qual elemento de rede agora atende esse assinante.

Durante o **roteamento**, uma solicitação pode chegar que deve alcançar o nó de atendimento de um assinante (por exemplo, um SLg Provide-Location-Request). O módulo consulta o vínculo e roteia diretamente para o par correto.

Se nenhum vínculo existir para um assinante, a solicitação cai na lógica de roteamento existente, incluindo Roteamento Avançado.

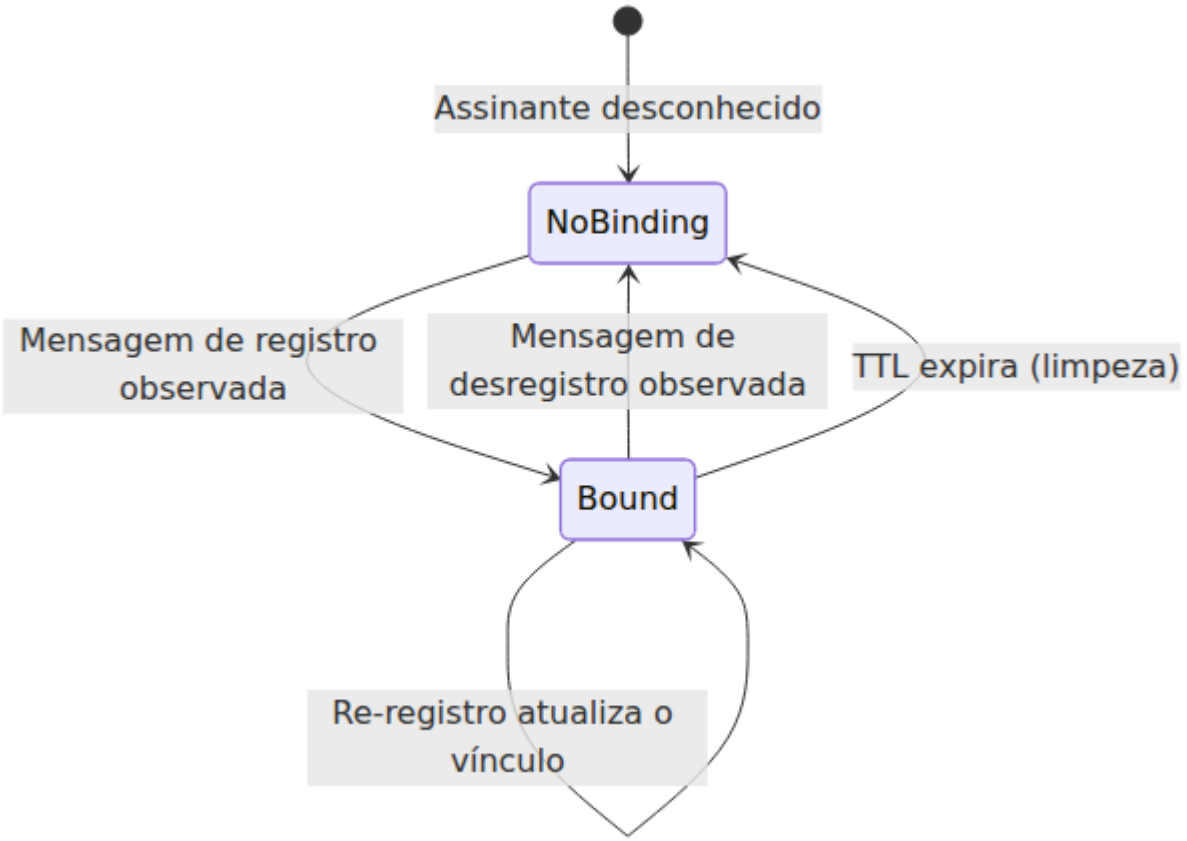

Is Request?
No
Yes
Pass Through
Learn if applicable
Route if applicable
SLg/SLh with binding
No binding or other message
Route to serving MME
Fall through to\nexisting routing

Ciclo de Vida do Vínculo



Aprendizado: Mensagens que Criam Vínculos

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



Interface	Mensagem	Código-Comando	Vínculo Criado	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 Vínculos

Interface	Mensagem	Código-Comando	Vínculo Removido
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 o módulo remove o último vínculo para um assinante, ele exclui todo o registro do assinante da tabela.

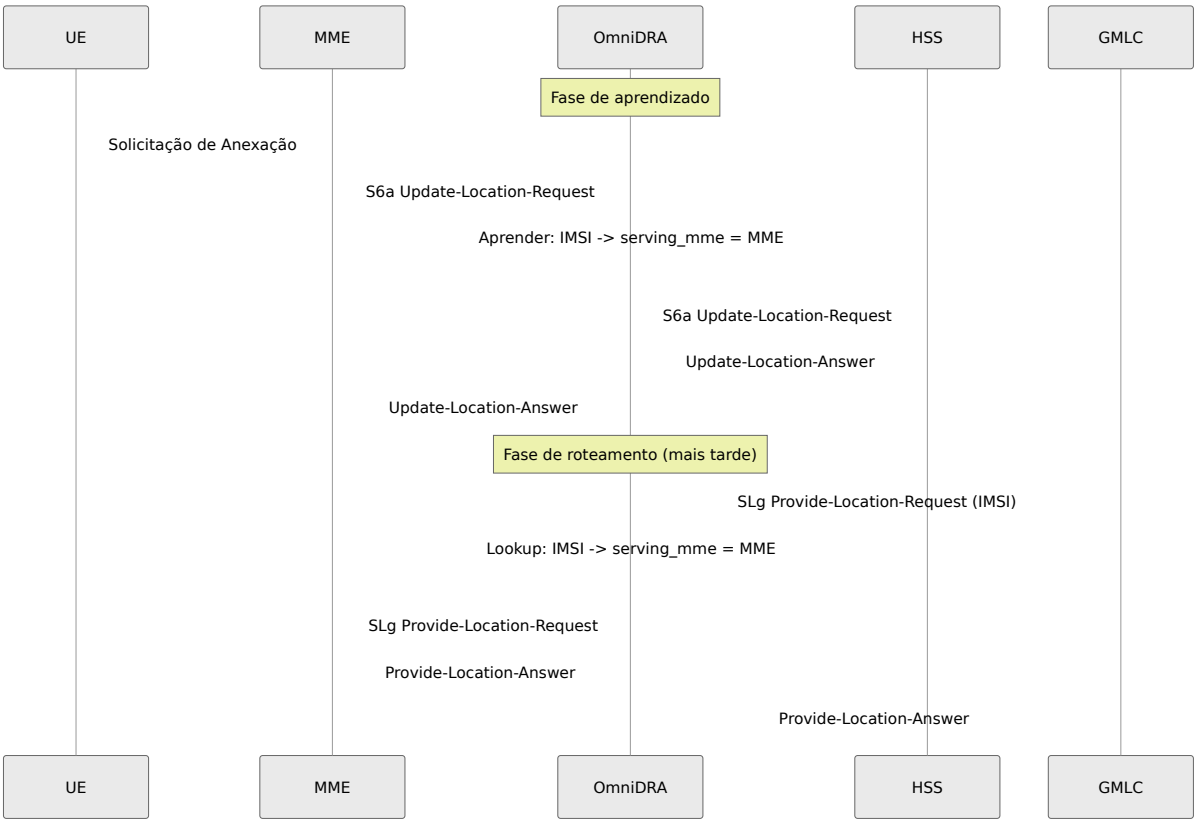
Roteamento: Mensagens que Usam Vínculos

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

O DRA retransmite uma solicitação roteada por um vínculo aprendido para o nó de atendimento. Ele utiliza o tempo limite de retransmissão base da

configuração `diameter.request_timeout` (veja Conectividade de Par ou Configuração). O módulo não define seu próprio tempo limite para essas solicitações roteadas.

Fluxo de Exemplo: Solicitação de Serviço de Localização



Posição do Pipeline

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

Configuração

Configure o módulo sob `module_subscriber_lookup` em `config/runtime.exs`.

```
module_subscriber_lookup: %{
  # Habilitar ou desabilitar o módulo
  enabled: false,

  # Quanto tempo manter vínculos antes de expirar (segundos)
  binding_ttl_seconds: 86400
}
```

Parâmetros

Parâmetro	Tipo	Requerido	Padrão	
`enabled`	Booleano	Sim	`false`	Habilitar ou des Consulta de As todas as solicit modificação e c vínculos.
`children`	Lista	Não	`[]`	Processos que quando habilita `[DRA.Router.M` para que o proc seja executado
`binding_ttl_seconds`	Inteiro	Não	`86400`	Tempo em segu de assinante ex remove. O padr executada na m com um mínim

Exemplos de Configuração

Implantação Padrão

Use isso para a maioria das redes onde os assinantes se re-registram dentro de 24 horas.

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

Como funciona: O módulo aprende vínculos MME/PGW/CSCF a partir do tráfego que passa e os mantém por 24 horas. O DRA roteia automaticamente solicitações SLg e SLh para o MME de atendimento correto. O módulo atualiza um vínculo sempre que observa uma nova mensagem de registro para o mesmo assinante.

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

Ambiente de Alta Rotatividade

Use isso para redes com mobilidade frequente de assinantes. Vínculos obsoletos devem expirar rapidamente.

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

Como funciona: Vínculos expiram após 1 hora. Use isso quando os assinantes se movem frequentemente entre MMEs. Vínculos obsoletos poderiam causar 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 Vínculo

Métrica: `diameter.subscriber_lookup.binding.update` **Tipo**: Contador **Descrição**: O DRA incrementa este contador cada vez que cria ou atualiza um vínculo de assinante. **Rótulos**:

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

Exclusões de Vínculo

Métrica: `diameter.subscriber_lookup.binding.delete` **Tipo**: Contador **Descrição**: O DRA incrementa este contador cada vez que uma mensagem de desregistro remove um vínculo de assinante. **Rótulos**:

- `imsi` - IMSI do assinante
- `binding_type` - Tipo de vínculo removido

Solicitações Roteadas

Métrica: `diameter.subscriber_lookup.route.count` **Tipo**: Contador **Descrição**: O DRA incrementa este contador cada vez que roteia uma solicitação SLg/SLh com um vínculo aprendido. **Rótulos**:

- `imsi` - IMSI do assinante
- `binding_type` - Tipo de vínculo que o DRA usou para roteamento (atualmente sempre `serving_mme`)
- `serving_host` - Origin-Host do par para o qual o DRA roteou a solicitação

Solução de Problemas

Solicitações de Localização Não Sendo Roteadas para o MME de Atendimento

Sintomas: O DRA não roteia SLg Provide-Location-Requests ou SLh LCS-Routing-Info-Requests para o MME esperado. Elas caem no roteamento padrão em vez disso.

Possíveis causas:

- O assinante não se registrou através deste DRA, portanto, nenhum vínculo existe.
- O vínculo expirou. O TTL foi excedido.
- O par MME de atendimento não está conectado a este DRA.
- `enabled` está definido como `false`.

Resolução:

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

Vínculos Não Sendo Aprendidos

Sintomas: A tabela de vínculos permanece vazia apesar do tráfego S6a/Gx/Cx passar 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. Confirme `enabled: true` na configuração.
2. Confirme que o DRA foi reiniciado após a alteração de configuração.
3. Verifique os logs de depuração do DRA para entradas `SubscriberLookup: Learned`. Estas confirmam a atividade de vínculo.

Vínculos Obsoletos Causando Solicitações Mal Roteadas

Sintomas: O DRA roteia solicitações de localização para um MME que não atende mais 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. Confirme que todo o tráfego de desregistro S6a flui através deste DRA.

Referência

IDs de Aplicação

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

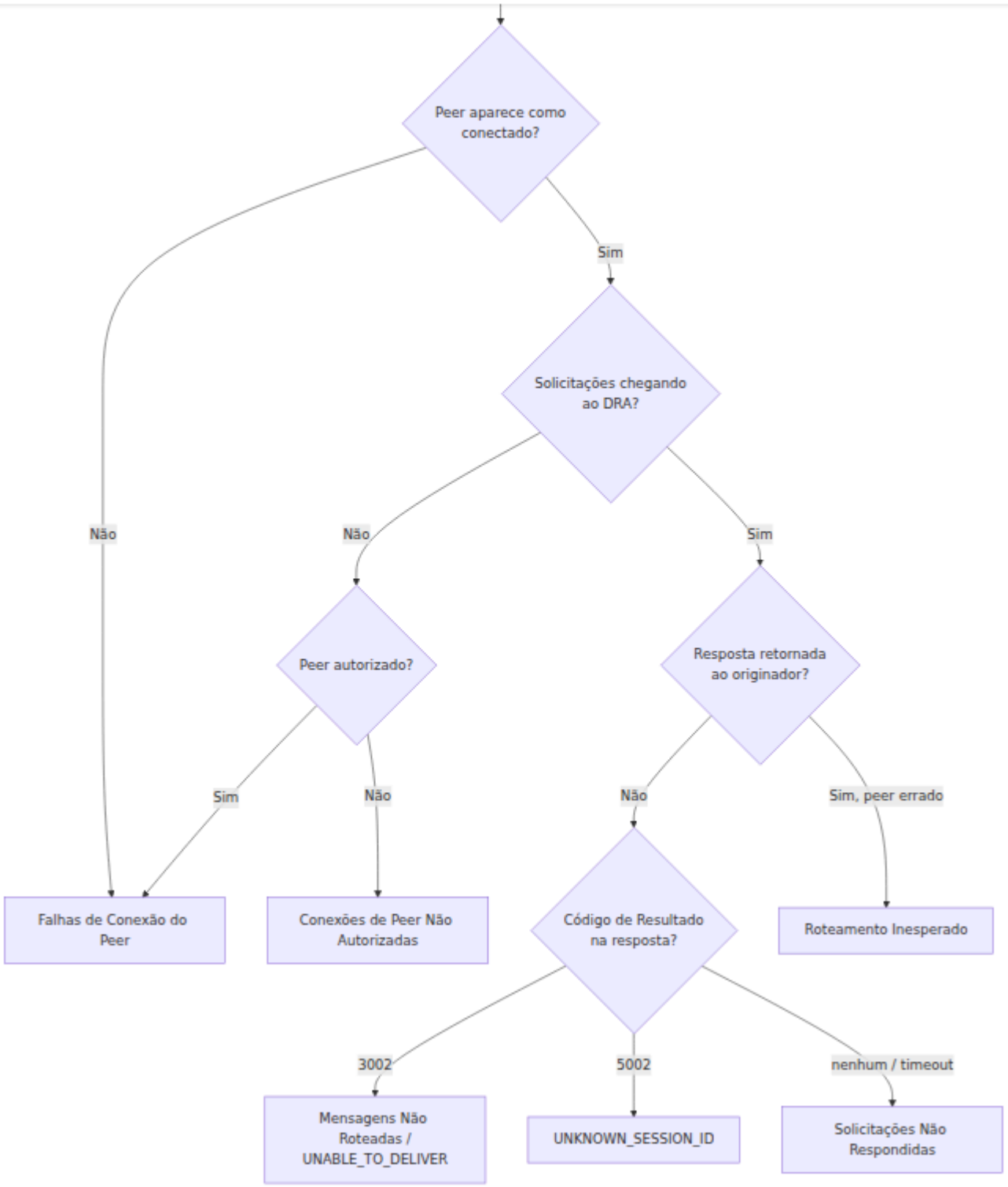
Tipo de Vínculo	Aprendido De	Usado Por	Descri►►ão
`serving_mme`	S6a ULR	SLg PLR, SLh RIR	O MME que atualmente atende o assinante
`serving_pgw`	Gx CCR-I	-	O PGW que gerencia a sessão do assinante (reservado para roteamento futuro)
`serving_cscf`	Cx SAR (Registro)	-	O S-CSCF que atende o registro IMS do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

Solução de Problemas

Este guia cobre os problemas mais comuns na operação do OmniDRA. Cada entrada lista os sintomas que um operador observa, as causas prováveis e um conjunto ordenado de etapas. Use as etapas para isolar e resolver o problema. Para os nomes das métricas neste guia, veja Métricas.

Árvore de Decisão de Diagnóstico

Use este fluxo para encontrar onde um problema se encontra antes de ir para uma seção específica.


Peer aparece como conectado?
Não
Sim
Solicitações chegando ao DRA?
Não
Sim
Peer autorizado?
Resposta retornada ao originador?
Sim
Não
Não
Sim, peer errado
Falhas de Conexão do Peer
Conexões de Peer Não Autorizadas
Código de Resultado na resposta?
Roteamento Inesperado
3002
5002
nenhum / timeout
Mensagens Não Roteadas / UNABLE_TO_DELIVER
UNKNOWN_SESSION_ID
Solicitações Não Respondidas

Regras de Roteamento

Regra Não Correspondente

Sintomas: Uma regra de roteamento avançada configurada nunca é acionada. O tráfego cai para o roteamento padrão ou para uma regra posterior em vez disso.

Causas possíveis:

- Uma ou mais condições de filtro não correspondem ao conteúdo real da mensagem.
- O código AVP no filtro não corresponde ao AVP usado pela aplicação Diameter em questão.
- Um padrão de expressão regular está incorreto ou mal ancorado.
- O tipo de mensagem está fora do escopo de `match` da regra. Por exemplo, um filtro apenas para solicitações se aplica a uma resposta.
- A regra usa o tipo de filtro errado para o módulo.

Resolução:

1. Verifique cada condição de filtro em relação a uma mensagem capturada. Todas as condições em uma regra devem corresponder para que a regra seja acionada.
2. Confirme se os códigos AVP correspondem à aplicação Diameter. Veja a referência de código AVP em Roteamento Avançado.
3. Teste padrões de expressões regulares contra valores AVP reais antes de implantá-los.
4. Confirme se o tipo de mensagem está dentro do escopo de `match` da regra.
5. Revise os tipos de filtro disponíveis em Roteamento Avançado. Confirme se você está usando o correto para o módulo.

Roteamento Inesperado

Sintomas: O DRA encaminha solicitações, mas para um peer diferente do pretendido. Alternativamente, o DRA usa roteamento padrão quando você

esperava que uma regra se aplicasse.

Causas possíveis:

- Ordem das regras. O roteamento avançado usa a regra que primeiro corresponder, portanto, uma regra anterior e mais ampla pode ofuscar uma posterior e mais específica.
- O nome do peer pretendido está incorreto ou o peer não está acessível.
- Duas regras têm filtros sobrepostos, e a errada vence.
- Nenhuma regra correspondeu, então o Roteamento Padrão assumiu.

Resolução:

1. Revise a ordem das regras. Coloque regras mais específicas antes das mais amplas. A primeira regra correspondente vence. Veja Roteamento Avançado.
2. Confirme que os nomes dos peers correspondem exatamente à configuração e que o peer alvo está conectado (`diameter_peer_status` relata `1`).
3. Verifique se há regras conflitantes com filtros sobrepostos. Reordene ou aperte-os.
4. Confirme que o comportamento de queda no Roteamento Padrão corresponde ao que você observa quando nenhuma regra se aplica.

Transformação Não Aplicada

Sintomas: Uma regra de roteamento corresponde, mas uma transformação de adição/edição/remover AVP não tem efeito na mensagem.

Causas possíveis:

- A transformação visa um código AVP que está incorreto para o caso de uso.
- Uma ação `edit` foi executada em um AVP que não existe na mensagem. `edit` modifica um AVP existente, mas nunca cria um.
- Os filtros da regra não correspondem ao fluxo de mensagem pretendido, então a mensagem nunca chega à transformação.
- O filtro de tipo de pacote (`:request` vs `:answer`) exclui a mensagem.

- A ação não suporta o tipo de pacote. `edit` se aplica apenas a solicitações.

Resolução:

1. Confirme se os códigos AVP estão corretos para o seu caso de uso. Veja a referência de código AVP em Roteamento Avançado.
2. Para uma ação `edit`, confirme se o AVP já existe na mensagem. Use `add` se o AVP pode estar ausente.
3. Confirme que os filtros da regra correspondem ao fluxo de mensagem pretendido.
4. Confirme que qualquer filtro de tipo de pacote (`:request` vs `:answer`) está correto para a direção que você transforma.
5. Confirme que a ação suporta o tipo de pacote. `edit` funciona apenas em solicitações.

Conectividade do Peer

Falhas de Conexão do Peer

Sintomas: Um peer configurado nunca atinge o estado conectado. `diameter_peer_status{origin_host="..."}` permanece `0`, e o roteamento para esse peer falha.

Causas possíveis:

- Um firewall bloqueia a `listen_port` configurada (padrão `3868`) ou a porta do peer.
- A configuração do peer tem um `ip`, `port` ou `transport` incorretos.
- Transporte incompatível. Um lado está configurado para `:diameter_tcp` e o outro para `:diameter_sctp`.
- O Origin-Host anunciado pelo peer não corresponde ao `host` configurado. As identidades devem corresponder exatamente.
- `initiate_connection` está configurado da mesma forma em ambas as extremidades, então nenhum dos lados disca, ou ambos discam.

- O peer não anuncia uma aplicação suportada durante a Troca de Capacidades (CER/CEA).
- Para um peer de saída, o serviço do peer não está em execução ou não está escutando.

Resolução:

1. Confirme se as regras do firewall permitem tráfego na `listen_port` e na porta do peer para o transporte configurado. SCTP requer o módulo SCTP do kernel e regras de firewall que permitam SCTP, não apenas TCP.
2. Confirme se os valores de `ip`, `port` e `transport` do peer correspondem exatamente ao endpoint real do peer.
3. Confirme se ambas as extremidades usam o mesmo protocolo de transporte.
4. Confirme se o Origin-Host anunciado pelo peer corresponde ao `host` configurado. As identidades Diameter devem corresponder exatamente para roteamento.
5. Verifique os papéis de `initiate_connection`. Exatamente um lado deve iniciar. Defina `initiate_connection: true` no lado que disca (tipicamente o DRA em direção a um HSS ou PCRF). Defina `false` no lado que espera (tipicamente em direção a um MME, SGSN ou P-GW).
6. Confirme que o peer anuncia pelo menos uma aplicação que o DRA atende durante CER/CEA. Um peer sem aplicação comum não é utilizável para roteamento.
7. Para um peer de saída, confirme que o serviço remoto está em execução e escutando. O DRA tenta novamente peers inatingíveis automaticamente.

Flutuação do Peer / Tráfego Enviado Muito Cedo Após Reconexão

Sintomas: Após um peer reconectar, o DRA despacha solicitações para ele, e elas falham imediatamente ou expiram enquanto o peer ainda está se estabilizando. Alternativamente, um peer que se recuperou recentemente parece utilizável, mas descarta tráfego.

Causas possíveis:

- O peer está instável na camada de rede e reconecta repetidamente (flutuação).
- O tempo de estabilização em nível de aplicação é muito curto para este peer, então o DRA roteia tráfego antes que o peer esteja pronto.
- Um tempo de estabilização excessivamente longo atrasa a recuperação legítima e estende a janela de interrupção.

Resolução:

1. Investigue a instabilidade do transporte subjacente primeiro (caminho de rede, failover de caminho SCTP ou reinicializações do lado do peer). Um peer em flutuação é um sintoma, não um problema de configuração de roteamento.

2. Ajuste o tempo de estabilização para que um peer recém-conectado precise permanecer ativo por um intervalo definido antes que o DRA roteie tráfego para ele. Um peer que desconecta dentro desta janela nunca entra no conjunto roteável. Isso mantém o tráfego longe de peers em flutuação.

```
config :dra, :gateway,
  # Mantenha um peer recém-conectado por este tempo em ms antes
de roteá-lo.
  # 0 = roteie imediatamente após a conclusão do CER/CEA.
  peer_settle_time_ms: 2_000,

  # Substituição opcional (ms) para o temporizador de watchdog
RFC 3539 Tw.
  # watchdog_timer_ms: 30_000
```

Parâmetro	Tipo	Necessário	Padrão	Descrição
`peer_settle_time_ms`	Inteiro	Não	`0`	Milissegundos que um peer recém-conectado deve permanecer ativo antes que o DRA roteie tráfego para ele. `0` roteia imediatamente após a conclusão do CER/CEA. Se peer desconectar durante a janela, o DRA nunca o marca como roteável. Isso mantém tráfego longe de peers em flutuação.
`watchdog_timer_ms`	Inteiro	Não	`30000`	Substituição opcional para o temporizador de watchdog RFC 3539, o intervalo de vivacidade do Device-

Parâmetro	Tipo	Necessário	Padrão	Descrição
				Watchdog (DWR/DWA). Deixe não definido a menos que você tenha u motivo específico pa alterar a cadência de vivacidade.

3. `diameter_peer_status` reflete a roteabilidade real. Um peer dentro de sua janela de estabilização ainda relata como não roteável nas métricas. Use isso para confirmar que o comportamento de estabilização funciona como você espera.

4. Aumente `peer_settle_time_ms` para peers que flutuam cronicamente para suprimir despachos prematuros. Mantenha-o baixo (ou `0`) para peers estáveis para minimizar o tempo de recuperação.

> O DRA promove um peer reconectado para roteável imediatamente após o tempo de estabilização em nível de aplicação expirar. Ele não espera o atraso implícito de reconexão do watchdog RFC 3539. O tempo de estabilização acima é o único controle explícito sobre quanto tempo o DRA mantém um peer em recuperação.

Conexões de Peer Não Autorizadas

Sintomas: O DRA rejeita a conexão de um peer. Alternativamente, `diameter_peer_unauthorized_connection_count_total` aumenta com um `origin_host` / `peer_ip` inesperado.

Causas possíveis:

- O peer que está se conectando não está na configuração de `peers`, e `allow_undefined_peers_to_connect` é `false`.
- O Origin-Host anunciado pelo peer não corresponde a nenhum `host` configurado.
- Um nó genuinamente não autorizado ou mal configurado tenta se conectar.

Resolução:

1. Confirme se deve permitir o peer. Se sim, adicione-o à lista de `peers` com seu exato `host`, `realm`, `ip`, `port` e `transport`.

2. Confirme se o Origin-Host anunciado pelo peer corresponde exatamente ao `host` configurado. O DRA trata uma discrepância como um peer desconhecido.

3. Mantenha `allow_undefined_peers_to_connect: false` em produção e adicione peers explicitamente. Ative-o apenas em ambientes controlados onde qualquer peer pode se conectar.

4. Mantenha `log_unauthorized_peer_connection_attempts: true` para que o DRA registre tentativas rejeitadas para revisão de segurança.

5. Investigue os rótulos `peer_ip` e `origin_host` reivindicados em `diameter_peer_unauthorized_connection_count_total`. Uma taxa sustentada de uma fonte desconhecida justifica uma investigação de segurança.

Parâmetro	Tipo	Necessário
`allow_undefined_peers_to_connect`	Booleano	Não
`log_unauthorized_peer_connection_attempts`	Booleano	Não

Problemas de Associação SCTP

Sintomas: Um peer SCTP falha ao estabelecer uma associação. Alternativamente, uma associação estabelecida cai ou falha inesperadamente. Peers TCP no mesmo host conectam normalmente.

Causas possíveis:

- O módulo SCTP do kernel não está carregado no host.
- Regras de firewall permitem TCP, mas não SCTP nos endereços configurados.
- Um ou mais endereços no conjunto de multihoming não são roteáveis a partir do peer.
- O IP primário não está na lista de `additional_ips` do peer.

- Apenas um endpoint está configurado para multihoming, então o DRA não alcança redundância total.
- Parâmetros SCTP do kernel governam o tempo de failover de caminho, e isso difere das expectativas.

Resolução:

1. Confirme se o módulo SCTP do kernel está carregado (o pacote `lksctp-tools` no Linux).
2. Confirme se as regras de firewall permitem tráfego SCTP em todos os IPs configurados, não apenas TCP.
3. Confirme se todos os endereços IP no conjunto de multihoming são roteáveis em ambas as direções entre o DRA e o peer.
4. Para um peer SCTP, inclua o IP primário na lista de `additional_ips`. Para peers TCP, o DRA usa apenas `ip`, porque TCP não suporta multihoming.
5. Configure ambos os endpoints para multihoming para redundância total de caminho.
6. Se o tempo de failover não for o esperado, revise os parâmetros SCTP do kernel. O tempo de failover de caminho depende deles, não da configuração do DRA.
7. Veja Multihoming SCTP para a configuração completa de transporte e suas limitações.

Entrega de Mensagens

Mensagens Não Roteadas / 3002 UNABLE_TO_DELIVER

Sintomas: Solicitações chegam ao DRA, mas o DRA as responde com o Código de Resultado `3002` (DIAMETER_UNABLE_TO_DELIVER). Alternativamente, o DRA não as encaminha para o destino esperado.

Causas possíveis:

- Nenhum peer conectado corresponde ao Destination-Host ou Destination-Realm da solicitação.
- O peer alvo está desconectado ou está dentro de sua janela de estabilização e ainda não é roteável.
- O peer alvo não anuncia suporte para a aplicação na solicitação.
- Uma regra de roteamento avançada selecionou um peer que não é acessível.
- Nenhum peer atende o realm solicitado. Isso resulta em `3003` (DIAMETER_REALM_NOT_SERVED), não `3002`.

Resolução:

1. Confirme que pelo menos um peer que corresponda ao Destination-Host ou Destination-Realm está conectado. `diameter_peer_status` para esse peer deve ler `1`.

2. Verifique se o peer alvo está dentro de sua janela de estabilização. Um peer que acabou de reconectar não é deliberadamente roteável até que `peer_settle_time_ms` expire. Veja Flutuação do Peer.

3. Confirme que o peer anuncia a aplicação na solicitação durante CER/CEA. O DRA exclui peers que não anunciam a aplicação da seleção.

4. Se uma regra de roteamento avançada visa um peer específico, confirme que esse peer está conectado. O DRA ignora peers desconectados. Veja Roteamento Avançado.

5. Distinga `3002` de `3003`. `3002` significa que o DRA não conseguiu selecionar um peer acessível. `3003` significa que nenhum peer atende o realm solicitado. Corrija a configuração do realm se você observar `3003`.

6. Acompanhe a taxa de respostas `3002` para identificar falhas sistemáticas de entrega:

```
rate(diameter_peer_message_result_code_count_total{result_code="3
[5m])
```

Solicitações Não Respondidas (Timeouts)

Sintomas: O DRA encaminha solicitações para um peer, mas nenhuma resposta retorna dentro do timeout. `diameter_peer_unanswered_request_count_total` aumenta, e `diameter_peer_last_response_delay` sobe.

Causas possíveis:

- O peer recebeu a solicitação, mas não respondeu dentro do `request_timeout` (padrão `5000` ms).
- O peer está sobrecarregado ou lento.
- Um caminho de rede está degradado. Isso é relevante para multihoming SCTP durante failover.
- O peer processou a solicitação, mas a resposta foi perdida em trânsito.

Resolução:

1. Identifique os peers lentos ou silenciosos:

   ```
   topk(5, sum by (routed_to) (
     rate(diameter_peer_unanswered_request_count_total[5m])
   ))
   ```

2. Compare com `diameter_peer_last_response_delay` para o mesmo peer. Isso mostra se ele está lento em vez de silencioso.

3. Revise a carga e a saúde do próprio peer. Timeouts sustentados geralmente indicam um problema a montante, não um problema do DRA.

4. Para peers SCTP, verifique eventos de failover de caminho que coincidam com os timeouts. Veja Multihoming SCTP.

5. Ajuste `request_timeout` apenas se o tempo de processamento legítimo do peer exceder o padrão. Um valor mais alto também estende o TTL de Métricas Estendidas (`request_timeout` + 5 segundos).

Parâmetro	Tipo	Necessário	Padrão	Descrição
`request_timeout`	Inteiro	Não	`5000`	Milissegundos a esperar por uma resposta antes qu o DRA trate o par solicitação/respos como expirado.

UNKNOWN_SESSION_ID em Gx / Gy com Estado

Sintomas: Sessões com estado (controle de política Gx, cobrança online Gy) falham com o Código de Resultado `5002` (DIAMETER_UNKNOWN_SESSION_ID). Normalmente, o peer rejeita uma atualização (CCR-U) ou terminação (CCR-T) mesmo que a inicial (CCR-I) tenha sido bem-sucedida.

Causas possíveis:

- O DRA roteou uma solicitação de meio de sessão para um peer diferente daquele que mantém o estado da sessão.
- O peer que mantém a sessão reiniciou e perdeu sua tabela de sessões.
- O DRA roteou para um peer que reconectou antes de restaurar seu estado. Isso é uma interação da janela de estabilização.
- O balanceamento de carga espalhou solicitações para um Session-Id entre múltiplos peers.

Resolução:

1. Confirme que todas as solicitações com o mesmo Session-Id chegam ao mesmo peer com estado. Para Gx e Gy, a afinidade de sessão é essencial. Mensagens de meio de sessão devem chegar ao peer que respondeu CCR-I. Prefira roteamento baseado em `Destination-Host`, ou uma regra avançada que vincula uma família de Session-Id a um peer, em vez de balanceamento de carga baseado em realm. Veja Roteamento Padrão e Roteamento Avançado.

2. Revise o `peer_selection_algorithm`. `:random` distribui entre peers correspondentes e pode quebrar a afinidade de sessão para aplicações com estado. `:failover` envia para o primeiro peer na lista. Para interfaces com estado, roteie com base em Destination-Host. A afinidade é então explícita e não depende de balanceamento de carga.

3. Verifique se o peer alvo reiniciou por volta do momento em que as respostas `5002` começaram. Um peer que perdeu sua tabela de sessões rejeita sessões em andamento. O cliente deve restabelecer essas sessões.

4. Se um peer reconectou recentemente, confirme que `peer_settle_time_ms` é grande o suficiente para que o DRA roteie tráfego apenas após o peer estar pronto. Veja Flutuação do Peer.

5. Acompanhe a taxa de respostas `5002` para correlacionar com eventos do peer:

```
rate(diameter_peer_message_result_code_count_total{result_code="5
[5m])
```

Referência: Códigos de Resultado

Comuns

Código	Nome	Significado	Referênci
2001	DIAMETER_SUCCESS	O DRA completou a solicitação com sucesso.	RFC 6733 §7.1.1
3002	DIAMETER_UNABLE_TO_DELIVER	O DRA não conseguiu selecionar um peer acessível para entregar a solicitação.	RFC 6733 §7.1.3
3003	DIAMETER_REALM_NOT_SERVED	Nenhum peer atende o realm solicitado.	RFC 6733 §7.1.3
3004	DIAMETER_TOO_BUSY	O peer selecionado está sobrecarregado.	RFC 6733 §7.1.3
5002	DIAMETER_UNKNOWN_SESSION_ID	O peer não reconhece o Session-Id. Esta é uma sessão com estado perdida ou mal roteada.	RFC 6733 §7.1.5
5012	DIAMETER_UNABLE_TO_COMPLY	O DRA não pode processar a solicitação	RFC 6733 §7.1.5

Código	Nome	Significado	Referênci
		conforme apresentada.	

Documentação Relacionada

- Métricas - Métricas Prometheus e consultas de exemplo referenciadas acima.
- Roteamento Padrão - Roteamento de prioridade RFC 6733, seleção de peers e roteamento de respostas.
- Roteamento Avançado - Roteamento baseado em regras, filtros, transformações e direcionamento de peers.
- Multihoming SCTP - Transporte SCTP, multihoming e failover de caminho.