



000000 DRA 0000 Diameter 000000000000000000 000000 000000000000000000
 000 ID000000 AVP 00000000000000000000000000000000

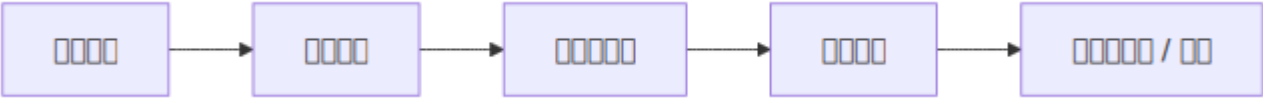
00000000000000000000 0000 0 0000000000000000 **priority** 0 **weight** 0000000000
 000000 Gx (PCRf) 0 Gy (OCS)00000000 0000000000000000000000000000000000000000

Diameter _____ DRA _____
_____ / _____

[illegible]

概要

DRA には、`match` によって、`route` によって、
DRA によって



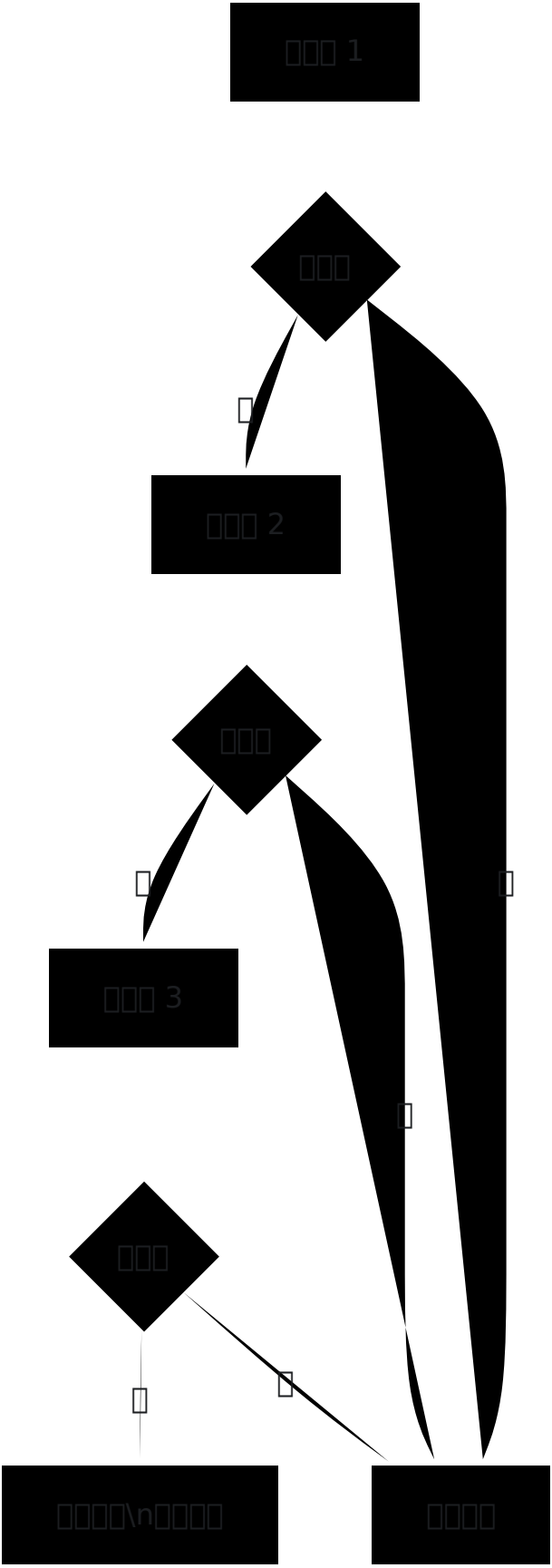
- 1. DRA によって、DRA によって
- 2. DRA によって `match` によって `:all`, `:any`, `:none` によって
- 3. DRA によって DRA によって
- 4. DRA によって DRA によって

概要

`match` によって DRA によって

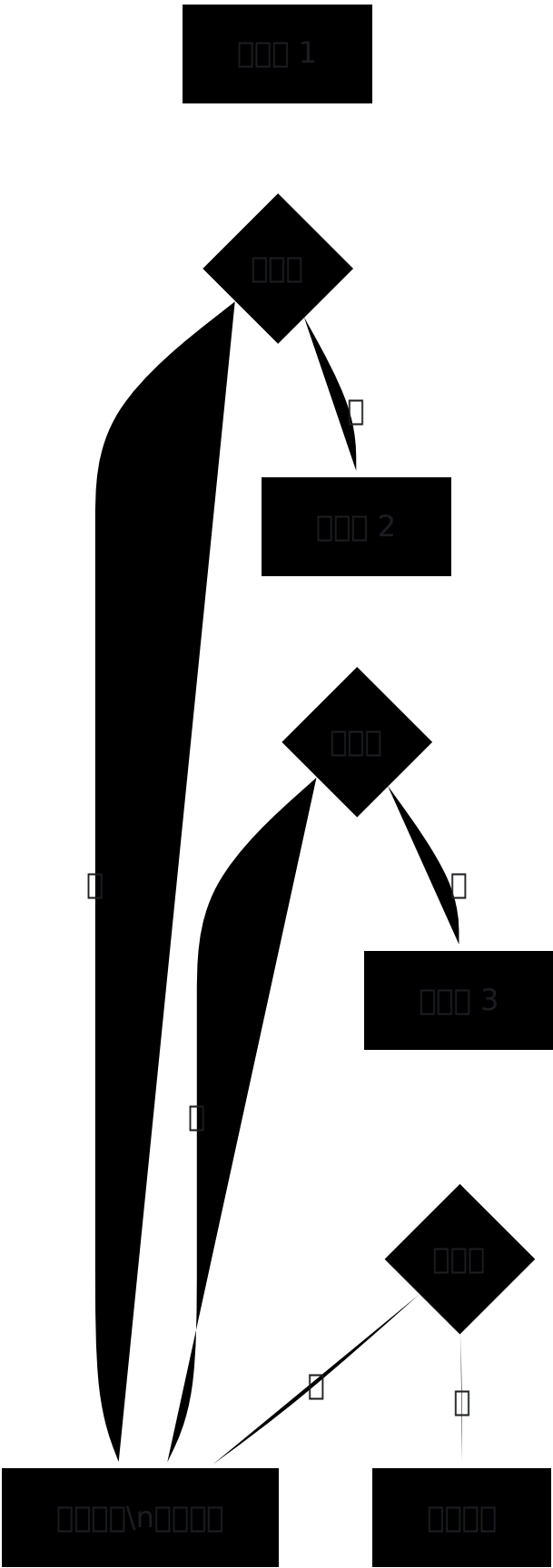
`:all` ()

filter1 AND filter2 AND filter3 によって



:any (节点)

filter1 OR filter2 OR filter3

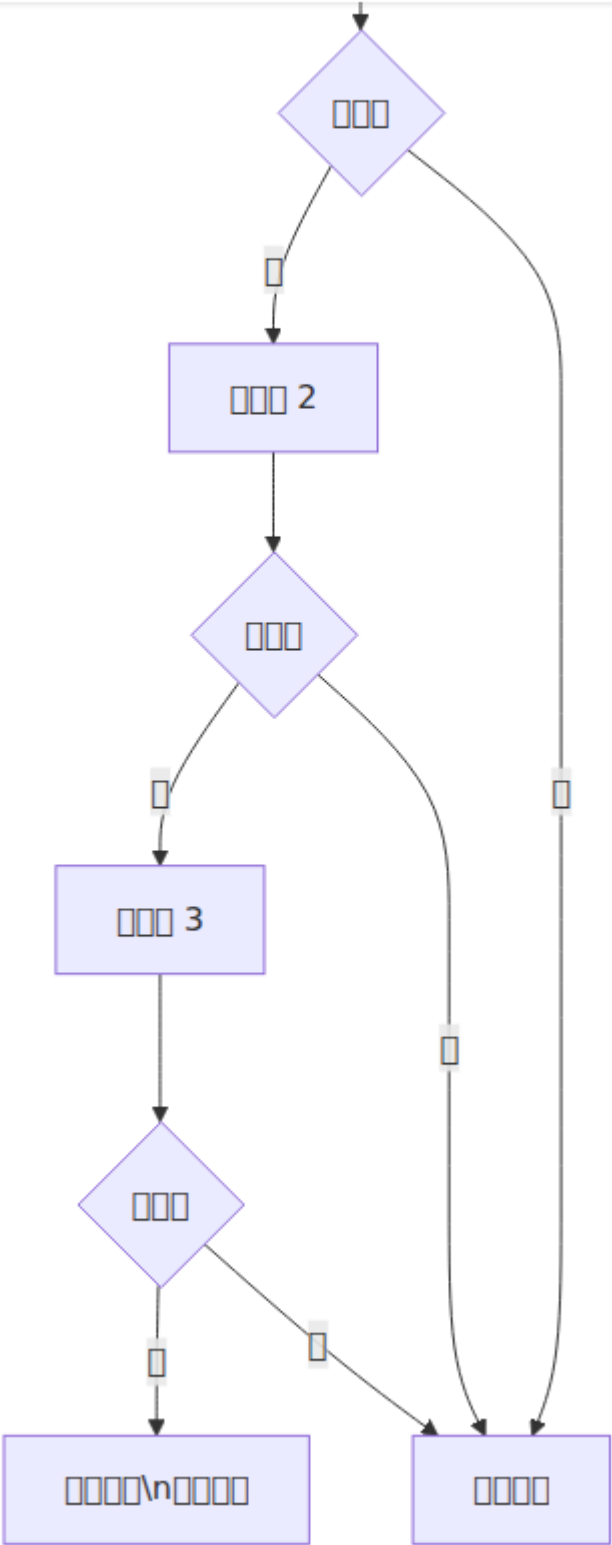


:none ()

NOT filter1 AND NOT filter2 AND NOT filter3

1

Core 5GC OmniCore OmniCall OmniRAN



概要

このドキュメントは、DRA の `peer_selection_algorithm` の設定方法と、DRA の動作について説明します。

設定方法

`diameter` の設定で `peer_selection_algorithm` を `:failover` または `:random` に設定することで、DRA の Diameter の動作を変更できます。DRA の動作は、DRA の設定で指定された `peer_selection_algorithm` に従います。

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

- `:failover` - 優先度の高いピアから接続を試みます。失敗した場合、優先度の低いピアに接続を試みます。
- `:random` - ランダムにピアを選択して接続を試みます。

動作

DRA は、`priority` と `weight` を使ってピアの優先度を決定します。

```
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}
  ]
}
```

この例では、“`pcrf-dr.example.com`”が最も優先度の高いピアです。

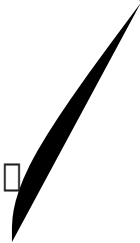
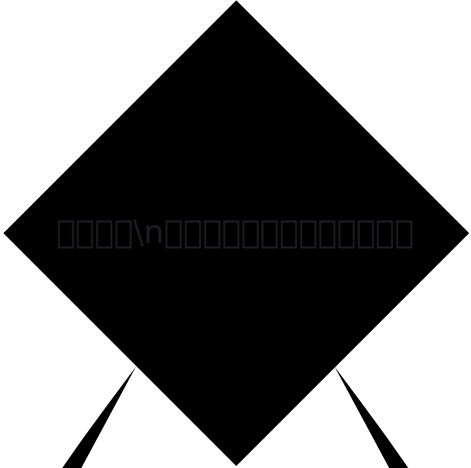
1. DRA はピアの優先度を決定します。
2. 優先度の高いピアから接続を試みます。DNS SRV の結果が異なるピアが存在する場合、DRA はランダムにピアを選択して接続を試みます。
3. 優先度の高いピアに接続が成功した場合、`weight` を使ってピアの重みを決定します。重みの高いピアは、優先度の高いピアに接続が成功した場合、優先度の低いピアに接続を試みます。

この例では、`priority` と `weight` の両方を `100` に設定しています。

□□□□□□□□

□□□□□□□□

□□□□□□□□□□



□□□□□□□□\n□□□□□□

□□□□□□□□□□

□□□□□□□□□□□□□□

項目	説明
セッション	セッションID 10, 20, 30セッション
セッション	セッションIDセッション
セッション	セッションIDセッション 70 / 30
セッション	セッション 10 セッションID 20 セッション

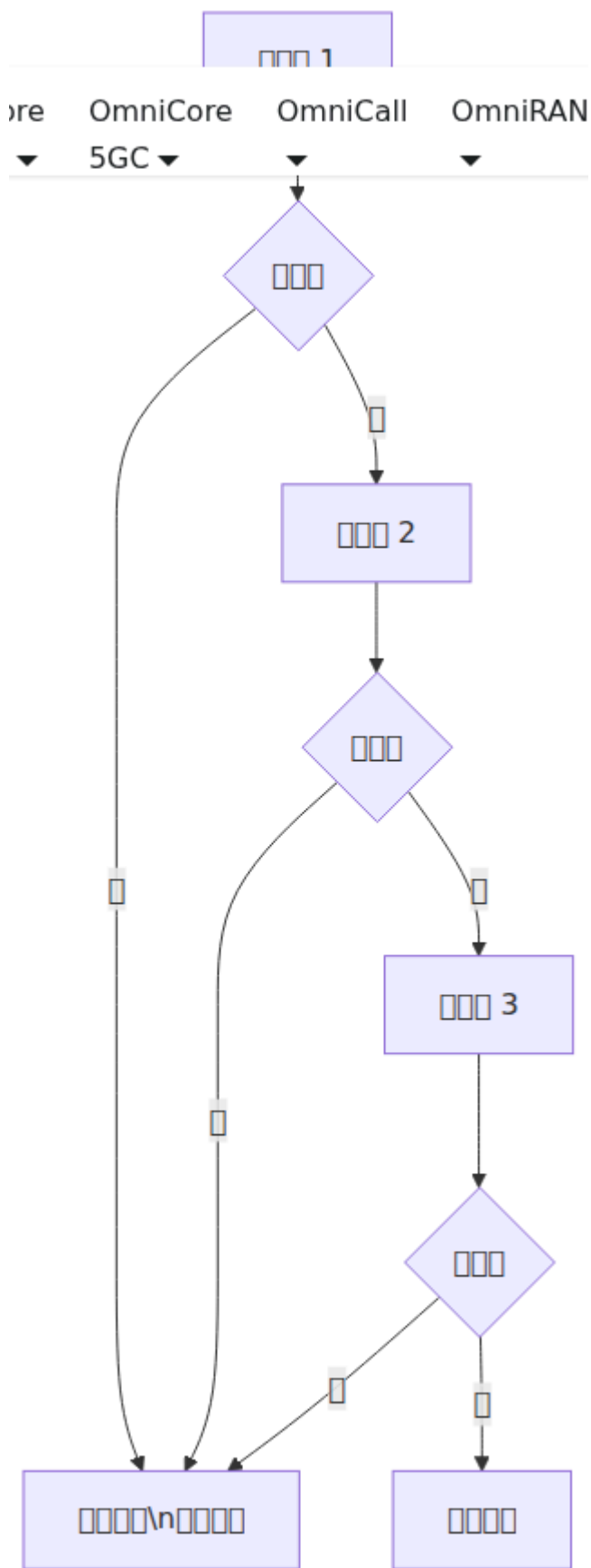
セッション (Gx / Gy)

セッションIDセッションID IDセッションIDセッションID IDセッション DRA セッションID
セッションIDセッションID **DIAMETER_UNKNOWN_SESSION_ID (5002)** セッション

セッションID Diameter セッションID **AVP (293)** セッションIDセッションID/セッションID

- セッションIDセッションIDセッションIDセッションID DRA IDセッションIDセッションIDセッションID
- セッションIDセッションID DRA セッションIDセッションID

セッションID Gx CCR-InitialセッションIDセッションID DRA セッションIDセッションIDセッションIDセッションID Origin-Host
セッションIDセッションID CCR-Update, CCR-TerminateセッションIDセッションIDセッションIDセッションIDセッションIDセッションID
Gx/Gy ID**CC-Request-Type AVP (416)** セッションID



Core\nCore/Core

配置

在 `config/runtime.exs` 中配置 `module_advanced_routing` 模块

```
module_advanced_routing: %{
  # 是否启用
  enabled: false,

  # 配置规则
  rules: [
    %{
      rule_name: "route_s6a_air_to_hss",
      match: :all,
      filters: [
        {:application_id, 16_777_251},
        {:command_code, 318}
      ],
      route: %{
        peers: ["hss01.example.com"]
      }
    }
  ]
}
```

配置项

配置项	数据类型	默认值	说明
<code>enabled</code>	布尔值	<code>false</code>	是否启用 DRA 模块
<code>rules</code>	列表	<code>[]</code>	DRA 配置规则列表

□ □ □ □

項目	単位	種類	制約	説明
rule_name	文字列	必須	-	DRA の名前。UI に表示される。
match	ブール値	必須	-	:all :any :none を指定する。
filters	文字列	必須	-	フィルタのリスト。
route	文字列	必須	-	DRA のルート。DRA の名前に付く。

111

```
0000 {filter_type, value} 0000000000000000
```

項目	形式	説明
:via_peer	<pre>{:via_peer, "mme01.example.com"}</pre>	Origin-Host 属性
:application_id	<pre>{:application_id, 16_777_251}</pre>	Diameter ID 属性 ID
:command_code	<pre>{:command_code, 318}</pre>	Diameter 属性
:avp	<pre>{:avp, {1, "9999990000000001"}}</pre>	AVP 属性 {avp_code, value}

1111

- `DRACM {command_code, [317, 318]} {via_peer, ["dra01.example.com",`

```
"dra02.example.com"]}
```

- `AVP { :avp, {1, ~r"999001.*"} }` `{ :avp, {1, ["505571234567", ~r"999001.*", ~r"505057.*"]} }`
- `:any` `:any` `{ :avp, {264, :any} }` `Origin-Host AVP { :via_peer, :any }`

AVP

AVP AVP Origin-Host AVP
{:avp, {code, value}}

--	--	--	--	--

route

項目	パラメータ名	説明
ピアリスト	%{peers: [...]}	ピアのIPアドレス/ポート番号を指定する
送信先ホスト	:destination_host	送信先のホスト名。AVP (293) を使用して DRA に通知する
ドロップフラグ	:drop	パケットをドロップするかどうかを示すフラグ
エラーコード	{:error, 3004}	Diameter プロトコルで DRA に Origin-Host、Origin-Realm、Session-Id を通知する

00 00 00000000 3xxx000000000000000000000000 3002 3003 3004 3007 00000000
5012DIAMETER_UNABLE_TO_COMPLY000000000000000000000000

配置示例/配置

名称	类型	单位	默认值	说明
host	字符串	-	-	Origin-Host 和 Diameter 的标识符
priority	整数	-	100	DRA 的优先级
weight	整数	-	100	DRA 的权重

配置

配置示例

S6a 配置示例 HSS HSS 配置示例 HSS

```
{
  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}
    ]
  }
}
```

hss01 hss01 hss-dr
DRA 3002

Gx PCRF

```
%{
  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}
    ]
  }
}
```

DRA CCR-Initial 50/50 PCRF DRA
PCRF

PCRF PCRF

```
%{
  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}
    ]
  }
}
```

PCRF 10 PCRf DRA 70/30 10 PCRf
pcrf-dr

DRA S6a HSS

```
%{
  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"]
  }
}
```

DRA dra01 dra02 Origin-Realm AVP 296 S6a
HSS

IMSI

IMSI 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"]
  }
}
```

Origin-Realm S6a 505571234567 999001
DRA

AVP

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

IMSI DRA AVP (293) AVP DRA

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

DRA IMSI 999999 S6a

□ □ □ □ □ □ □ □ □ □ □

DIAMETER_TOO_BUSY

```
%{
  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}
}
```

00000 DRA 0 **DIAMETER_TOO_BUSY (3004)** 0000000000 S6a 00000000000000
00DRA 00000000000000

□ □ □ □ □ □

[illegible]

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

DRACORE **DIAMETER_UNABLE_TO_COMPLY (5012)** S6a UE 321

DRACORE

API GET /api/rules routing_algorithm

DRACORE API

- route.selection / "priority_weight" "failover" "random"
- priority_weight API priority weight

```
{
  "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": []
  }
}
```

--

1. 00000000 000000000000000000000000000000000000
2. 00000000 000000000000000000000000
3. 0000000000 0000000000000000000000000000
4. 000000 00000000 rule_name 00000000 UI 00000000
5. 00000000000000000000/0000 00 Gx/Gy 000000/0000000000DRA 00000000000000

--

UNKNOWN_SESSION_ID

Gx Gy 5002

□□□□□

- PCEF/OCF DRA
- : random

□□□□□

1.
2. / DRA

□□□□□□ **3002**

DRG DIAMETER_UNABLE_TO_DELIVER

□□□□□

- `XXXXXXXXXXXXXXXXXXXX`
- `XXXXXXXXXXXXXXXXXXXX` Origin-Host `XXXXXXXXXX`

□□□□□

1. `UI` `Origin-Host`
2. `host` `Origin-Host`

□ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ ◆ ◆ □ □

- enabled false
-

- `enable_avp_dra` AVP `enable_avp_dra` AVP

1. `enable_avp_dra`

1. `enable_avp_dra` `true` DRA
2. `enable_avp_dra`
3. `enable_avp_dra` DRA `enable_avp_dra` AVP

2. `enable_avp_dra`

Diameter

Code	Message	Description	Reference
3002	DIAMETER_UNABLE_TO_DELIVER	Unable to deliver message	RFC 6733 §7.1.3
3003	DIAMETER_REALM_NOT_SERVED	Realm not served	RFC 6733 §7.1.3
3004	DIAMETER_TOO_BUSY	Too busy	RFC 6733 §7.1.3
5002	DIAMETER_UNKNOWN_SESSION_ID	Unknown session ID	RFC 6733 §7.1.5
5012	DIAMETER_UNABLE_TO_COMPLY	DRA unable to comply	RFC 6733 §7.1.5

AVP

AVP	Code	Meaning	Reference
Session ID	293	Session ID	RFC 6733 §6.5
Peer ID	263	Peer ID	RFC 6733 §8.8
CC-Request Type	416	CC Gx/Gy Peer INITIAL/UPDATE / TERMINATION Peer	RFC 4006 §8.3

Peer ID

ID	Code	Meaning	Reference
16777238	Gx	PCEF / PCRF Peer	3GPP TS 29.212
16777251	S6a/S6d	MME/SGSN / HSS Peer	3GPP TS 29.272
4	Gy	OCF / OCS Peer	RFC 4006

Notes

- Peer - Peer / Peer
- Peer - Peer
- Peer - Peer config/runtime.exs Peer peer_selection_algorithm

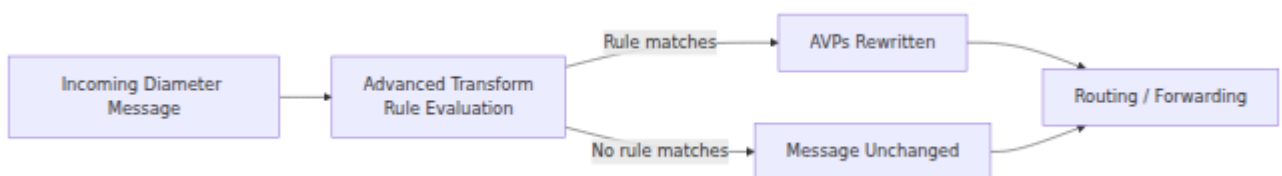
Diameter

Diameter
DRA

```
** :edit AVP :remove AVP :overwrite**
```

```
AVP AVP
```

DRAFT



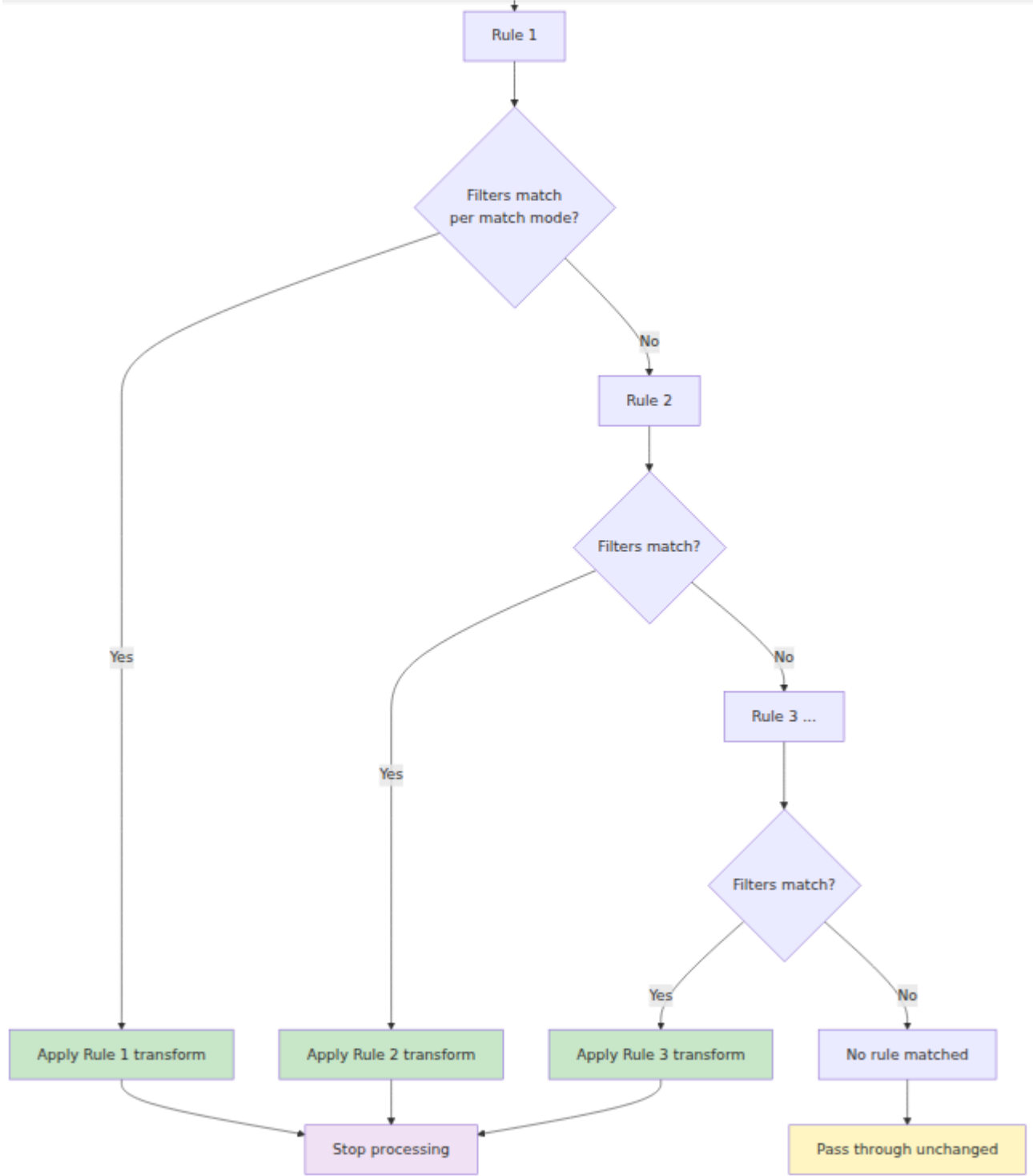
--	--	--	--	--	--

[illegible]

match

- **:all** - **AND**
- **:any** - **OR**
- **:none** - **NOR**

□ □



AAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

項目	形式	説明
:packet_type	{:packet_type, :request}	パケットの種類: request :answer パケットは黄色で示す パケット
:application_id	{:application_id, <id>}	Diameter ID S6a/S6d 16777251 ID
:command_code	{:command_code, <code>}	Diameter AIR 318
:avp	{:avp, {:<code>, <value>}}	AVP AVP
:to_peer	{:to_peer, "hss01.example.com"}	宛先 hss01.example.com
:from_peer	{:from_peer, "hss01.example.com"}	送信元 hss01.example.com

パケット

:packet_type パケットの種類

- **:request** - Diameter リクエスト
- **:answer** - Diameter レスポンス

:application_id :command_code :avp :packet_type

AVP

:avp {avp_id, avp_value} avp_id AVP avp_value

AVP User-Name Origin-Host :avp AVP

- { :avp, {1, "9999990000000001"} }
- { :avp, {1, ["50557...", "505057..."]} } AVP OR
- { :avp, {1, [~r"9999999.*"]} }

~r"pattern"

~r"999001.*"	999001 User-Name/IMSI
~r"^310[0-9]{3}.*"	MCC/MNC
~r".*test\$"	test

:to_peer :from_peer

- :to_peer DRA
- :from_peer

OR

項目	値	説明
<code>:to_peer</code>	<code>{:to_peer, "hss01.example.com"}</code>	送信元アドレス hss01.example.com
<code>:to_peer</code>	<code>{:to_peer, ["hss01.example.com", "hss02.example.com"]}</code>	送信元アドレスリスト
<code>:from_peer</code>	<code>{:from_peer, "hss01.example.com"}</code>	送信先アドレス hss01.example.com ポート

動作

`transform`は、`action`で指定された動作を実行する。

動作	説明	成功	失敗
<code>:edit</code>	指定されたAVPを編集	○	○
<code>:remove</code>	指定されたAVPを削除	○	○
<code>:overwrite</code>	指定されたAVPを上書き	○	○

`:edit`

```
transform: %{\n  action: :edit,\n  avps: [{:avp, {1, "9999990000000002"}}]\n}
```

`:edit`は、

- `Diameter` `AVP`
- `AVP` `AVP` `DRA`
- `avps` `{:avp, {<code>, <new_value>}}` `AVP`
- `:edit` `AVP` `AVP` `:overwrite`

`:remove`

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

`:remove`

- `avps` `AVP`
- `AVP` `avps` `:any`
-

`:overwrite`

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

`:overwrite`

- `AVP` `dictionary`
- `:overwrite` `edit` `AVP`
- `dictionary` `Diameter` `:diameter_gen_3gpp_s6a`
-

字典表

名称	类型	值	备注	说明
action	字符串		-	字典项的 :edit 字典项 AVP - 字典项 :remove 字典项 AVP - 字典项 :overwrite 字典项 AVP 字典项 - 字典项
avps	字符串		-	AVP 字典项 :edit/:overwri 字典项 { :avp, {<code>, <new_value>}} 字典项 AVP 字典项 <new_value> 字典项 :remove 字典项 <code> 字典项 :any 字典项
dictionary	字符串	字典项 :overwrite	:not_used	字典项 Diameter 字典项 D 字典项 AVP 字典项 :diameter_gen_3gpp_s6a 字典项 :remove 字典项

配置

在 config/runtime.exs 中 module_advanced_transform 字典项 字典项 enabled 字典项 true 字典项

```
module_advanced_transform: %{\n  enabled: true,\n  rules: [\n    %{\n      rule_name: "transform_user_name",\n      match: :all,\n      filters: [\n        {:application_id, 16_777_251},\n        {:command_code, 318},\n        {:avp, {1, "9999990000000001"}}\n      ],\n      transform: %{\n        action: :edit,\n        avps: [{:avp, {1, "9999990000000002"}}]\n      }\n    }\n  ]\n}
```

Table

Field	Type	Value	Default	Description
enabled	Boolean	false	false	Whether the transform is enabled. If false, the transform will not be applied.
rules	Array of Rule Objects	-	-	Array of rules to apply. Each rule is an object with the following fields: rule_name, match, filters, and transform. The transform field is optional.

Table

rules: [rule1, rule2, ...]

match: :all S6a 16777251 318
User-Name AVP 1 9999990000000001 :edit User-Name
9999990000000002 User-Name AVP DRA

DRA IMSI

2 IMSI

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

Origin-Realm AVP 296 User-Name/IMSI AVP 1
50557 Destination-Realm AVP 283
IMSI

MVNO DRA IMSI

3 AVP

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

packet_type from_peer
hss01.partner.example.com command_code AIA 318
:remove Origin-Host AVP 264 AVP

AVP AVP AVP

4S6aAVP

```
module_advanced_transform: %{\n  enabled: true,\n  rules: [\n    %{\n      rule_name: "overwrite_user_name_s6a",\n      match: :all,\n      filters: [\n        { :packet_type, :request },\n        { :application_id, 16_777_251 },\n        { :command_code, 318 }\n      ],\n      transform: %{\n        action: :overwrite,\n        dictionary: :diameter_gen_3gpp_s6a,\n        avps: [{ :avp, { 1, "9999990000000002" } }]\n      }\n    }\n  ]\n}
```

16777251AIR318:overwrite
:diameter_gen_3gpp_s6aAVP:overwrite
edit/AVP
AVPAVP:editAVP

ID

ID			
16777251	S6a/S6d	MME/SGSN ↔ HSS	3GPP TS 29.272

□ □ □ □

項目	名称	規格	規格
318	無線LAN/無線LAN/AIR/AIA	S6a/S6d	3GPP TS 29.272 §7.2.5
316	無線LAN/無線LAN/ULR/ULA	S6a/S6d	3GPP TS 29.272 §7.2.3

AVP

AVP	AVP	AVP	AVP
1	User-Name	3GPP TS 29.272 / RFC 6733 §8.14	
264	Origin-Host	RFC 6733 §6.3	
283	Destination-Realm	RFC 6733 §6.6	
296	Origin-Realm	RFC 6733 §6.4	

--

1. 名前 - 名前空間のパス
2. 名前 - 名前空間のパス
3. 名前 - 名前空間のパス
4. 名前 - 名前空間のパス `rule_name` ???
5. 名前 - 名前空間のパス

□□□□

- □□□□ - □□□□□□□□□□□□□□□□

DRA 簡介

OmniDRA 是一個 **Diameter** 伺服器 (**DRA**) 實現 Diameter 協議的軟體。MME/SGSN 透過 HSS/PCRF 與 OCS 互動，OmniDRA 透過 Elixir/Erlang 的 OTP `:diameter` 模組實現 **Diameter** 協議。OmniDRA 是一個

目錄

1. DRA 簡介
2. 安裝
3. 配置
4. 測試
5. 部署
6. 參考
7. 附錄

DRA 簡介

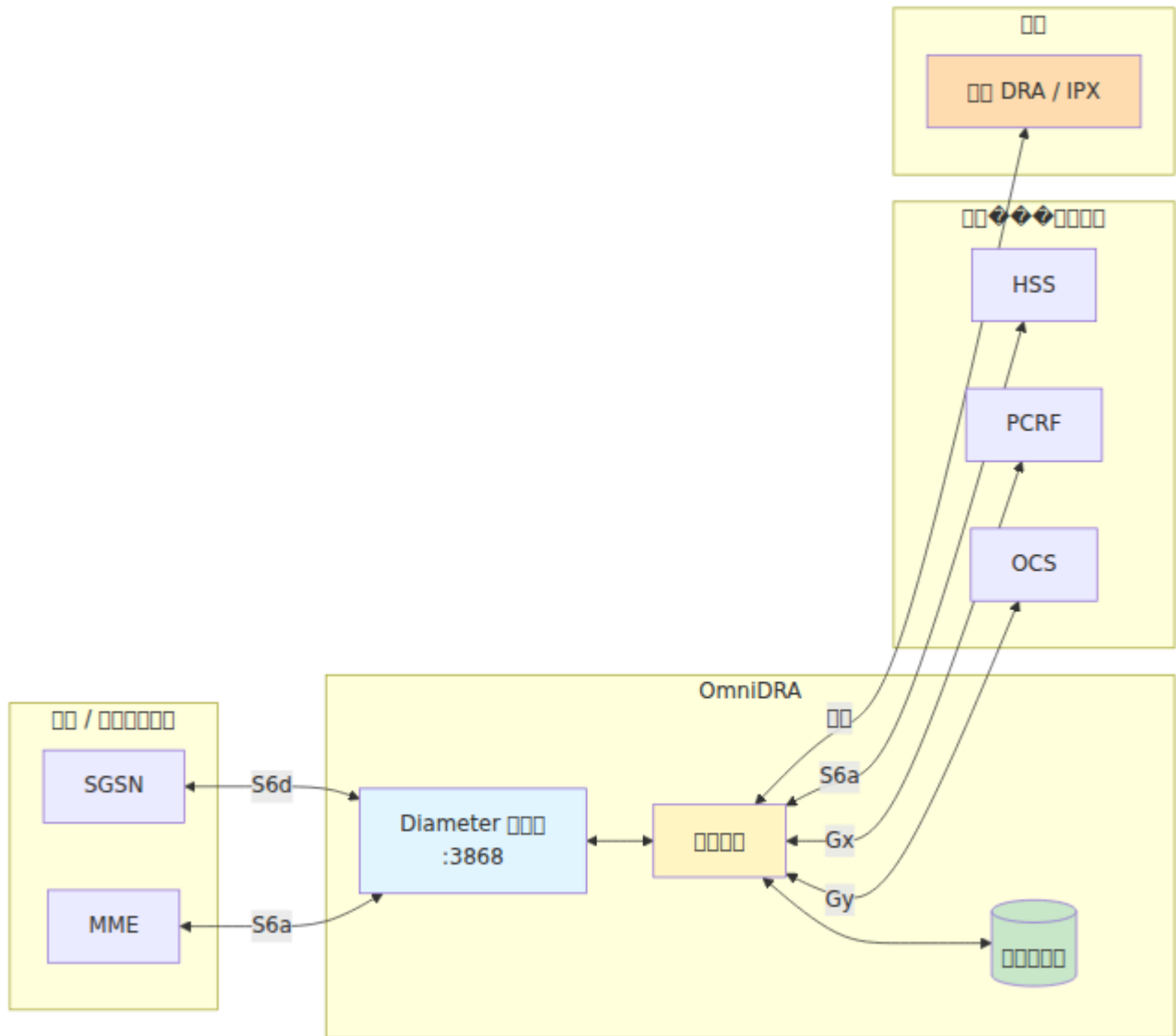
一個 **Diameter** 協議的實現。Diameter 協議是 IETF 定義的一種基於 IP 的協議，用於在網絡設備之間交換用戶數據。OmniDRA 是一個 Diameter 協議的實現，它支持 S6a、Gx 和 Gy 等接口。Diameter 協議的實現需要遵循 IETF 的 RFC 6733 標準。

特點

- 實現了 Diameter 協議 (RFC 6733) 的所有功能，包括 **Destination-Host**、**Destination-Realm** 和 **Application-Id** 等。
- 支持多種 Diameter 協議的實現，包括 DRA 的 **Hop-by-Hop Identifier** 等。
- 支持多種 Diameter 協議的實現，包括 DRA 的 **Hop-by-Hop Identifier** 等。
- 支持多種 Diameter 協議的實現，包括 DRA 的 **Hop-by-Hop Identifier** 等。

概要

この図は、MME、SGSN、HSS、PCRF、OCS、DRAの構成と、DRAがDiameterプロトコルを用いてDRAと接続する様子を示しています。



DRAはTCP/SCTPポート3868でDiameterプロトコル(RFC 6733)を実行し、HSS、PCRF、OCSと接続します。DRAはDiameterプロトコルを用いてDRAと接続します。

詳細

この図は、DRAがDiameterプロトコルを用いてDRAと接続する様子を示しています。

☐☐ Diameter ☐☐

☐☐☐☐☐
☐☐☐☐☐☐☐

Diameter ☐☐☐
GSMA FS.19 ☐☐

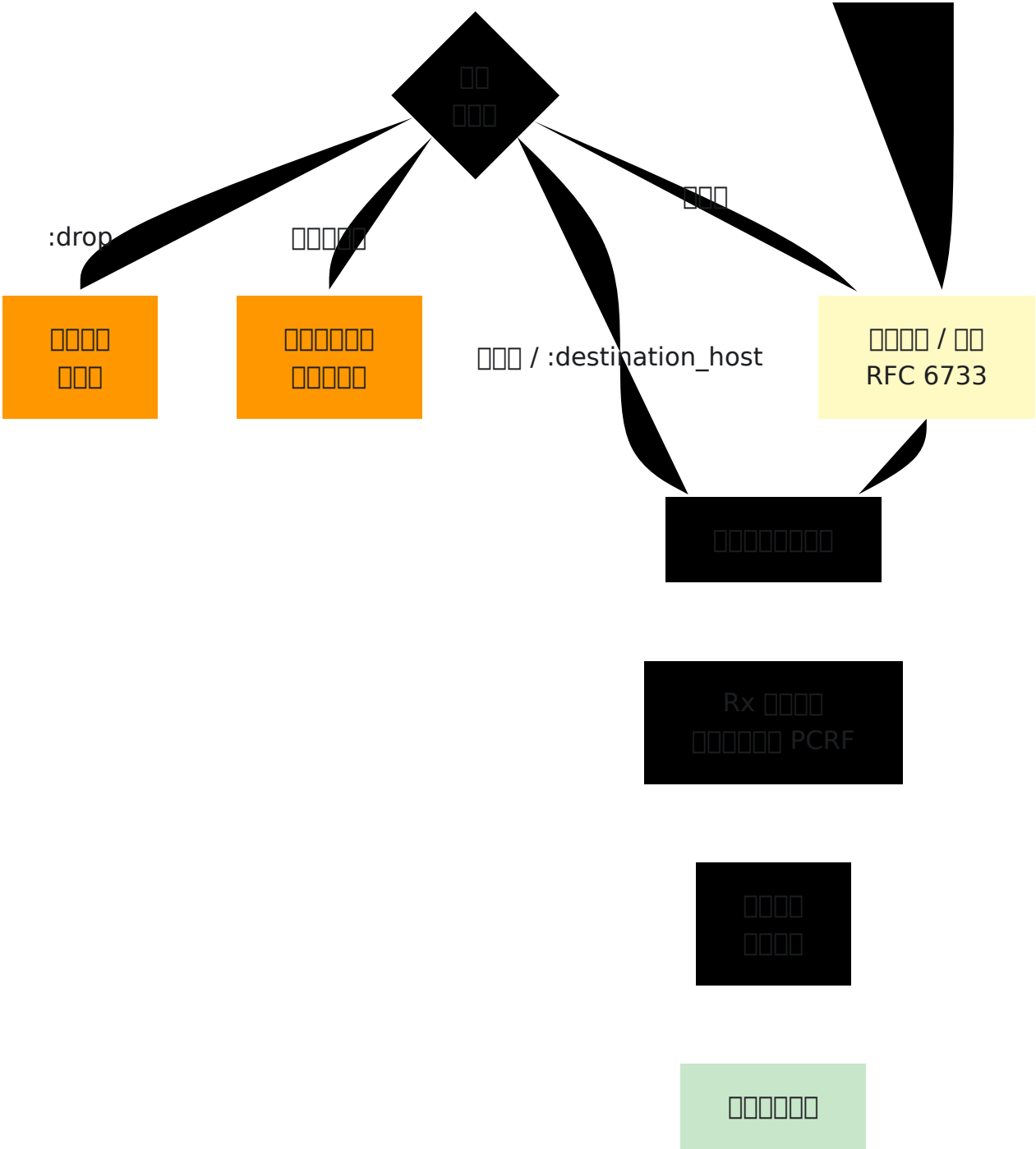
AVP ☐☐☐
☐☐☐☐☐ / ☐☐

☐☐☐☐

☐☐☐☐☐

☐☐☐☐
☐☐☐

☐☐☐☐☐☐☐



UE

UE	UE
UE	UE
Diameter UE	UE GSMA FS.19 UE 1/2/3 UE UE

項目	説明
AVP 項目	AVP 項目 AVP 項目
項目	項目
項目	項目
項目	項目
項目	項目
項目 / 項目	項目 RFC 6733 項目
Rx 項目	Rx 項目 PCRF 項目 Rx Session-Termination-Request 項目 Rx 項目
項目	項目 AVP Route-Record Origin-Host/Realm 項目 項目

00000000 :drop 0000DRA 000000000000error 000000000000 **Result-Code** 0
Diameter 000000DRA 000000000000000000 **Destination-Host** 0000000000000000
000

```
Rx [REDACTED] Rx[REDACTED]
:drop [REDACTED]error [REDACTED] Destination-Host[REDACTED]
```

--	--	--	--	--	--	--

□□□□□□□□□□ DRA □□ RFC 6733 §6.1 □□□□□□□□□□□□

1. **Destination-Host AVP (293)** - 32 ビット AVP。DRA が宛ホストの IP アドレスを格納する。
2. **Destination-Realm AVP (283)** - 32 ビット Destination-Host。DRA が宛ホストの IP アドレスを格納する。
3. **Application-Id** - DRA がアプリケーション ID を格納する。

5/5

[illegible]

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

- **Diameter** Origin-Host / FQDN
- **Application-Ids**
- **Application-Data**

DRA CER/CEA

DRA

□□□□

DRA □□□□□□□□□□□□□□□□□□□□

□□

□□□□□□ □□□□ DRA □□□□□□□□□□□□□□ AVP□□□□□□ DRA □□□□□□□□□□□□□□
□□□□□

□□

□□□□□□□□□□□□□□ DRA □□□□□□□□□□□□□□ DRA □□□□□□□□□□

□□ Diameter □□

□□□□□
□□□□□□□

Diameter □□□
GSMA FS.19 □□

AVP □□□
□□□□□ / □□

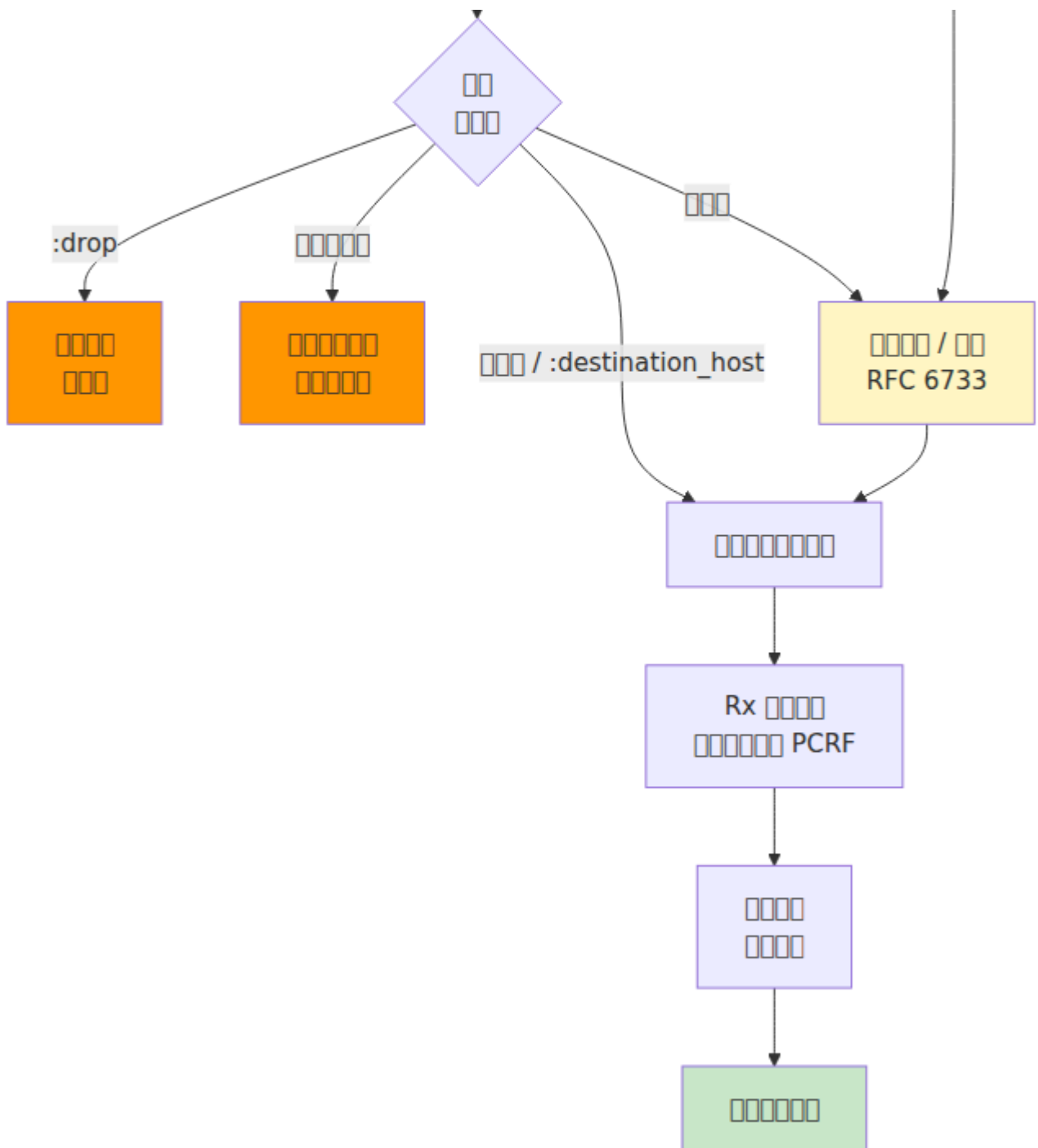
ore OmniCore OmniCall OmniRAN OmniCharge Platform 5A □□□□ ▼
▼ 5GC ▼ ▼ ▼ ▼ ▼

□□□□

□□□□□

□□□□
□□□

□□□□□□□



Destination host

- Destination host - Diameter message received
- ID ID - Diameter message received
- ID - Diameter message received DRA Diameter message received
- ID - DRA Diameter message received AVP ID
- ID - Diameter message received DRA Diameter message received

Diameter 协议规范在 3GPP TS 29.272 中定义，详细描述了 MME/SGSN 与 HSS 之间的接口。
 该接口遵循 RFC 6733 §6.2 的定义。

接口

3GPP 接口 ID

ID	接口	描述	规范
16777251	S6a/S6d	MME/SGSN ↔ HSS 接口	3GPP TS 29.272
16777238	Gx	PCEF ↔ PCRF 接口	3GPP TS 29.212
4	Gy	计费系统 ↔ OCS	3GPP TS 32.299

□□□□ **AVP**

項目	項目名	定義	参照
293	Destination-Host	宛先ホスト名	RFC 6733 §6.5
283	Destination-Realm	宛先ドメイン	RFC 6733 §6.4
282	Route-Record	経路レコード	RFC 6733 §6.7.1

--	--	--	--	--

- **3GPP** - RFC 6733 **IPsec** **VPN**
- **3GPP** - **IPsec**/**VPN**
- **3GPP** - **DRA** **VPN**
- **3GPP** - **GSMA FS.19 Diameter** **AVP** **VPN**

□ □ DRA □ □

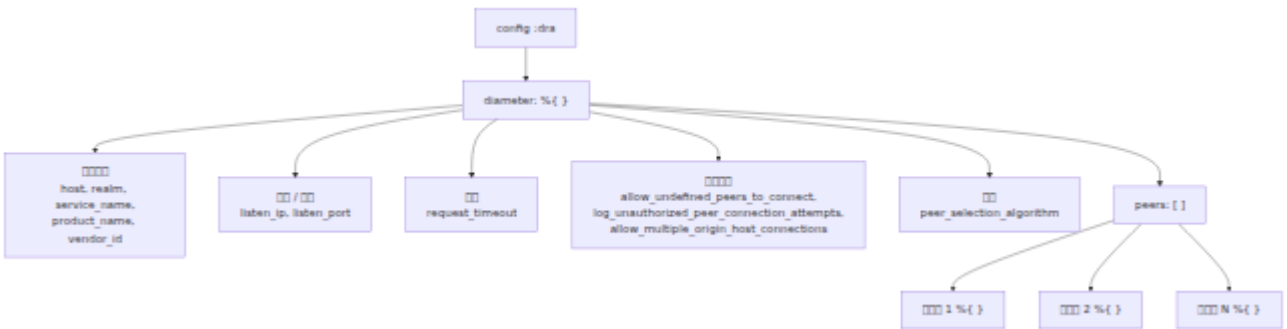
```
DRA  Omnidra  Diameter  
config :dra, diameter: % {...}  
config/runtime.exs
```

11

```

0000000000000000 diameter 000000000000 DRA 000000000000 peers 000000
0000000000 Diameter 0000

```



配置

```
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: [  
      # 配置 peers...  
    ]  
  }  
}
```

部署

部署 DRA 需要配置 `diameter` 配置项

項目	項目名	単位	初期値	説明
	service_name	文字列	-	Diameter サーバのサービス名を指定します。 例: cisco
	listen_ip	IP アドレス	-	DRA が受信する IP アドレスを指定します。 指定しない場合は、すべての IP アドレスを受信します。 Multihome 環境では、複数の IP アドレスを指定できます。
	listen_port	ポート番号	-	DRA が受信するポート番号を指定します。 例: 3868
	host	ホスト名	-	DRA が接続するホスト名を指定します。 例: realm.cisco.com Host 名は、ドメイン名で指定する必要があります。 例: dra01.prod.cisco.com CER/CEA には適用されません。
	realm	文字列	-	DRA が接続するrealmを指定します。 例: Origin-Realm
	product_name	文字列	-	CER/CEA の製品名を指定します。
	vendor_id	文字列	-	RFC 6724 で定義された Vendor ID を指定します。 例: VendorID
	request_timeout	秒	5000	リクエストのタイムアウト時間（秒）を指定します。 DRA がリクエストを送信するときに適用されます。


```
config :dra, diameter_socket_buffer_size: 4_194_304

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

項目	単位	値	説明
diameter_socket_buffer_size	byte	4194304 (4 MB)	SCTP 送信バッファサイズ :dra config :dra, diameter_socket_buffer_size diameter: ... :diameter_sctp ... Multihoming

peer_selection_algorithm

peer_selection_algorithm 設定は DRA 設定ファイルの peer_selection_algorithm 項目で指定します。

設定値	説明
:random	DRA 宛の接続をランダムに選択します。
:failover	DRA 宛の接続をフェイルオーバーで選択します。

peers

peers 設定は Diameter 設定ファイルの peers 項目で指定します。
DRA 設定ファイルの peers 項目で指定します。

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

配置参数

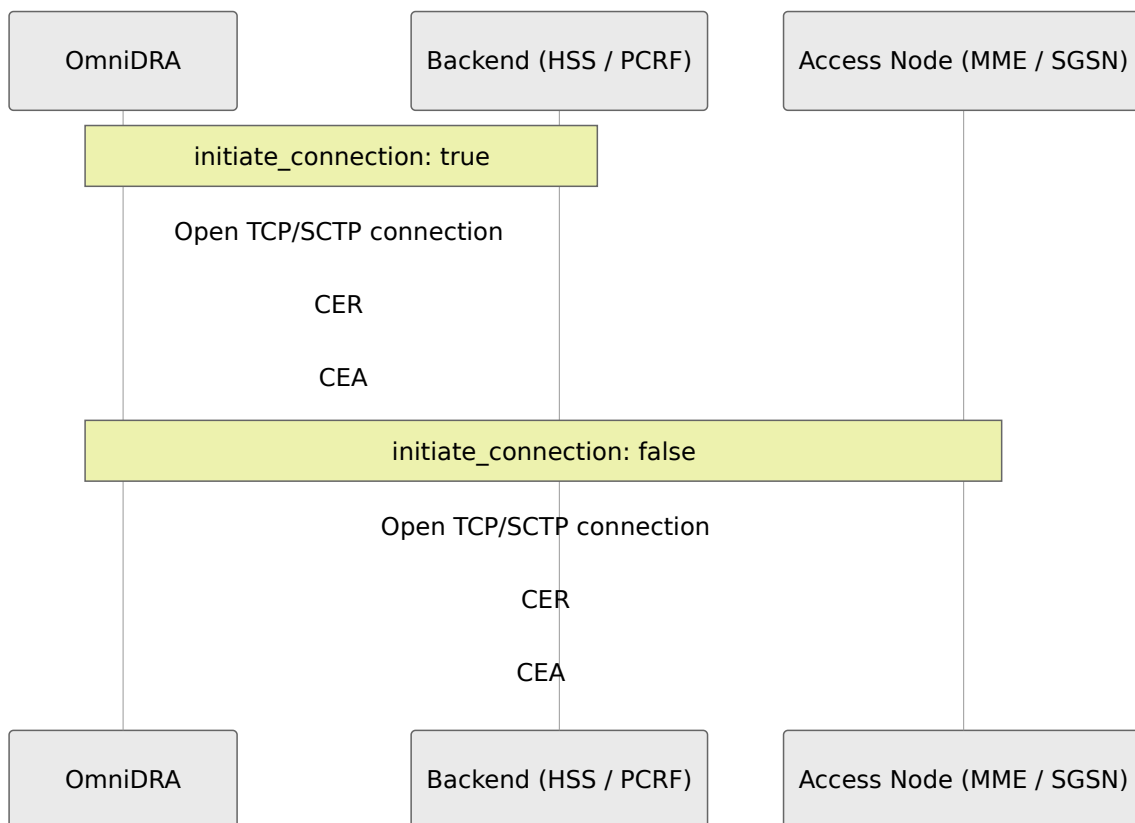
参数	数据类型	是否必填	默认值	说明
host	字符串	否	-	指定 Diameter 服务器的 FQDN 或 IP 地址。如果指定 FQDN，则必须指定 realm。
realm	字符串	否	-	指定 Diameter 服务器的 realm。
ip	字符串	否	-	指定 Diameter 服务器的 IP 地址。
additional_ips	字符串	否	-	指定 SCTP 服务器的 IP 地址。如果指定 IP，则必须指定 transport。:diameter_sctp 指定 SCTP Multihoming。
port	整数	否	-	指定 Diameter 服务器的端口。默认值为 3868。
transport	字符串	否	-	指定传输协议。可以是 diameter_tcp、TCP、diameter_sctp 或 SCTP。如果指定 SCTP，则必须指定 additional_ips。
source_ip	字符串	否	listen_ip	DRA 的源 IP 地址。如果指定 listen_ip，则必须指定 source_ip。
source_port	整数	否	-	DRA 的源端口。如果指定 source_port，则必须指定 source_ip。

initiate_connection: true

- DRA 通过 Diameter 与 Backend 建立 TCP/SCTP 连接
- 与 Backend 建立 HSS 与 PCRF 连接
- 与 Backend 建立 DRA 连接

initiate_connection: false

- DRA 通过 Diameter 与 Backend 建立连接
- 与 Backend 建立 MME/SGSN 与 P-GW 连接
- 配置 peers 配置 allow_undefined_peers_to_connect 为 true
- DRA 通过 CER 与 Backend 建立 Origin-Host 与 Diameter 连接 IP 与 DRA 建立 host 连接



配置

DRA 配置 host 为 realm 为 Origin-Host
dra01.production.omnitouch realm 为 Origin-Realm 与 DRA 建立
连接

원래 DRA 설정은 아래와 같이 DRA 설정을 하게 됩니다

`origin_host` 와 `origin_realm` 값을 DRA 설정합니다

원래 DRA 설정은 아래와 같이 `host` 와 `realm` 값을 설정하고 DRA 설정을 하게 됩니다

```
peers: [  
  {%  
    # 원래 DRA 설정  
    host: "hss.partner.example",  
    realm: "partner.example",  
    ip: "10.0.0.9",  
    port: 3868,  
    transport: :diameter_sctp,  
    initiate_connection: true,  
    # DRA 설정  
    origin_host: "dra-partner.epc.mnc001.mcc001.3gppnetwork.org",  
    origin_realm: "epc.mnc001.mcc001.3gppnetwork.org"  
  }  
]
```

원래 설정은 `origin_host` 와 `origin_realm` 값을

원래 DRA 설정은 아래와 같이



```

origin_realms=DRA
Origin-Host=

```

DRA CER DRA

□ □ □ □

□□□□□□□□ DRA□DRA □□□□ MME □ SCTP □□□□□□ TCP □□ HSS □ PCRF□

```

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 ☐☐ MME ☐☐
      %{
        host: "mme01.operator.example.com",
        realm: "operator.example.com",
        ip: "10.100.2.15",
        port: 3868,
        transport: :diameter_sctp,
        initiate_connection: false
      },
      # HSS - DRA ☐☐☐☐☐☐☐☐☐☐
      %{
        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 ☐☐☐☐
      %{
        host: "pcrf01.mvno.example.com",
        realm: "mvno.example.com",
        ip: "10.100.3.22",
        port: 3868,
        transport: :diameter_tcp,
        initiate_connection: true
      }
    ]
  }

```

--	--	--	--

- **Host** - DRA **Host** **Host** **Host**
- **Host** - **DRA** **CER/CEA** **DRA** **Host** **realm** **product_name** **vendor_id**
- **ID 10415** - **3GPP Vendor-Id** **3GPP Diameter**
- **peer_selection_algorithm** **DRA**

--	--	--	--

- `allow_undefined_peers_to_connect` `false`
- `log_unauthorized_peer_connection_attempts: true`
- `listen_ip` `listen_port`

--	--	--	--

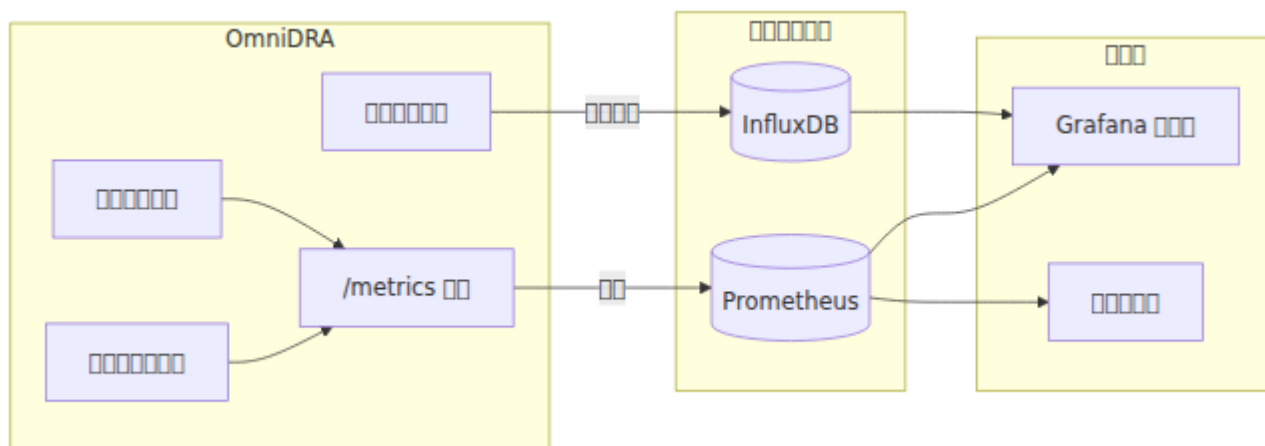
- **SCTP Multihoming** - 多个 SCTP 端点同时接收数据
- **配置** - DRA 配置 `peer_selection_algorithm` 参数
- **配置** - 配置 `host` 参数

Overview

OmniDRA uses **Prometheus** to monitor the system and generate reports. Prometheus is a monitoring and alerting toolkit written in Go. It is designed to be scalable, reliable, and easy to use. Prometheus uses a Pull model where the client (Prometheus) connects to the server (PromQL) to retrieve data.

Architecture

OmniDRA uses `/metrics` to expose Prometheus metrics. Prometheus is a monitoring and alerting toolkit written in Go. It is designed to be scalable, reliable, and easy to use. Prometheus uses a Pull model where the client (Prometheus) connects to the server (PromQL) to retrieve data.



Prometheus uses `/metrics` to expose Prometheus metrics. Prometheus is a monitoring and alerting toolkit written in Go. It is designed to be scalable, reliable, and easy to use. Prometheus uses a Pull model where the client (Prometheus) connects to the server (PromQL) to retrieve data. **InfluxDB** is a time series database that stores metrics. Prometheus uses **InfluxDB** to store metrics.

OmniDRA uses `/metrics` to expose Prometheus metrics. Prometheus is a monitoring and alerting toolkit written in Go. It is designed to be scalable, reliable, and easy to use. Prometheus uses a Pull model where the client (Prometheus) connects to the server (PromQL) to retrieve data. `rate()` is a function that calculates the rate of change of a metric over time.

Configuration

OmniDRA uses DRA to configure the system.

Configuration

Configuration

- `diameter_peer_status`

- : Gauge
- : □□□□□□□1□□□□□□0□□

11

名前	名前
origin_host	ホスト名 (FQDN)
ip	IP アドレス

□□□□:

```
# 000000000000
diameter_peer_status{origin_host="hss01.example.com"}

# 000000000000
count(diameter_peer_status == 0)
```

□ □ □ □ □ □ □

- `type: diameter_peer_supported_applications`
- `type: Gauge`
- `type: CER/CEA`

11

項目	説明
origin_host	接続元ホスト名 (FQDN)
supported_applications	サポートされているアプリケーション

0000:

```
# 0000000000000000
diameter_peer_supported_applications{origin_host="hss01.example.com"}
```

0000

0000

- 00: diameter_peer_message_count_total
- 00: Counter
- 00: 0000000000000000

00:

00	00
origin_host	0000000000
received_from	0000000000
application_id	000000 ID0000 0000 ID
cmd_code	0000000000 000000
application_name	000000000000 3GPP_S6a00
cmd_name	0000000000 AIR00
direction	request 0 response0

0000:

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

#
sum by (application_name)
(rate(diameter_peer_message_count_total[5m]))
```

Filtering

Labels

- `diameter_peer_message_result_code_count_total`
- Counter
- Labels

Table

Label	Description
<code>origin_host</code>	Origin host
<code>routed_to</code>	Routed to
<code>application_id</code>	Application ID
<code>cmd_code</code>	Command code
<code>application_name</code>	Application name
<code>cmd_name</code>	Command name
<code>result_code</code>	Result code

#####:

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

# #####
sum by (result_code) (
  rate(diameter_peer_message_result_code_count_total{
    result_code!="2001"
  }[5m])
)
```

#####:

##	##	##
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

##

####

- 名称: `diameter_peer_last_response_delay`
- 类型: Gauge
- 描述: 记录DRA → HSS → DRA的延迟

属性:

属性	类型
<code>origin_host</code>	字符串
<code>routed_to</code>	字符串
<code>application_name</code>	字符串
<code>cmd_name</code>	字符串

示例:

```
# 记录 HSS 的延迟
diameter_peer_last_response_delay{routed_to="hss01.example.com"}

# 记录 S6a 的延迟
max(diameter_peer_last_response_delay{application_name="3GPP_S6a"})
```

配置: `Gauge{histogram_quantile 0.99}`

计数器

描述:

- 名称: `diameter_peer_unanswered_request_count_total`
- 类型: Counter
- 描述: DRA 未响应的请求总数

属性:

欄名	説明
origin_host	送信元ホスト
routed_to	宛先ホスト
application_id	アプリケーション ID
cmd_code	コマンドコード
application_name	アプリケーション名
cmd_name	コマンド名

例:

```
# 送信元ホスト
rate(diameter_peer_unanswered_request_count_total[5m])

# 宛先ホスト
topk(5, sum by (routed_to) (
  rate(diameter_peer_unanswered_request_count_total[5m])
))
```

監視項目

監視項目

- 項目: `diameter_peer_unauthorized_connection_count_total`
- 単位: Counter
- 単位: 1000000000000000000

例:

項目	説明
origin_host	接続元ホスト名
supported_applications	サポートされているアプリケーション
peer_ip	接続元 IP アドレス

例:

```
# 接続元ホスト名
rate(diameter_peer_unauthorized_connection_count_total[5m])

# 接続元 IP アドレス
diameter_peer_unauthorized_connection_count_total > 0
```

接続元ホスト名と接続元 IP アドレスの AVP を取得する

接続元ホスト名

接続元ホスト名は、接続元ホスト名フィールドに格納されている

例:

例:

- 項目: diameter_advanced_routing_drop_count_total
- 項目: Counter
- 項目: :drop DRA の接続元

例:

項目	説明
application_id	アプリケーション ID
cmd_code	コマンドコード
application_name	アプリケーション名
cmd_name	コマンド名

例:

```
# 集計対象を指定
sum by (cmd_name) (
  rate(diameter_advanced_routing_drop_count_total[5m])
)

# 集計結果を出力
sum(rate(diameter_advanced_routing_drop_count_total[5m]))
```

④ **DRA** 監視:

- 監視対象の DRA を指定
- DRA の監視
- DRA の監視結果の取得

監視対象

項目

- 項目: diameter_advanced_routing_error_count_total
- 項目: Counter
- 項目: 監視対象の DRA の名前

例:

項目	説明
result_code	結果コード
application_id	アプリケーション ID
cmd_code	コマンドコード
application_name	アプリケーション名
cmd_name	コマンド名

例:

```
# 結果コードを絞り込む
sum by (result_code) (
  rate(diameter_advanced_routing_error_count_total[5m])
)

# DIAMETER_T00_BUSY の場合
rate(diameter_advanced_routing_error_count_total{
  result_code="3004"
}[5m])

# 結果コードを絞り込む
diameter_advanced_routing_error_count_total{
  result_code="3004"
} > 100
```

❗ **DRA** の注意事項:

- 結果コードを絞り込む際は、必ず「3004」を指定する
- DRA の結果コードは、必ず「3004」である
- DRA の結果コードは、必ず「3004」である

例:

項目	項目	項目	項目
enabled	項目 項目 項目	項目	項目 true 項目 項目 false 項目 DRA 項目
attach_attempt_reporting_enabled	項目 項目 項目	項目	項目 LTE 項目S6a AIR/AIA項目

InfluxDB 項目

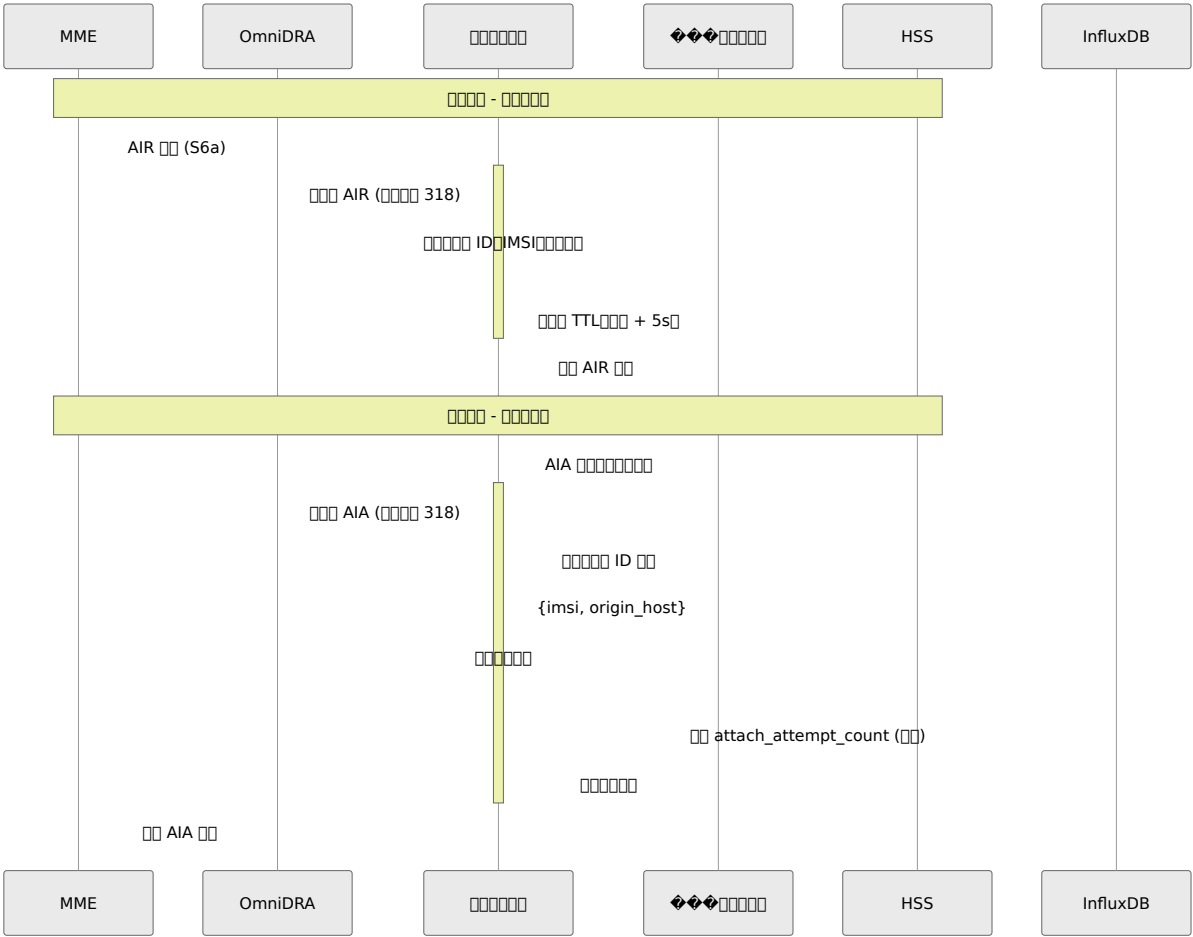
項目項目項目 Instream 項目項目項目 InfluxDB項目Instream項目項目 config :dra,
DRA.Metrics.InfluxDB 項目項目項目

```
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
```

項目	項目名	項目値	項目単位	項目説明
auth	auth	-	-	InfluxDB 接続用 [username: ..., password: ...]
database	database	-	-	InfluxDB データベース名
host	host	-	-	InfluxDB 接続用 IP アドレス
port_udp	port_udp	8089	-	InfluxDB 接続用 UDP ポート InfluxDB 接続用 UDP ポート 8089 接続用 HTTP API ポート 8086
writer	writer	Instream.Writer.Line	-	Instream 接続用 Instream.Writer.Line 接続用 接続用

接続用

S6a AIR/AIA 接続用 ID 接続用
HSS 接続用



attach_attempt_count

imsi	IMSI AVP

origin_host	
result_code	HSS

코드	설명
2001	DIAMETER_SUCCESS
5001	DIAMETER_AUTHENTICATION_REJECTED
5004	Diameter AVP unsupported

이러한 RFC 6733 에러는 다음과 같습니다

예시

- IMSI - IMSI
- IMSI - IMSI
- IMSI - IMSI MME/SGSN
- IMSI - IMSI

참고

- S6a AIR/AIA ID 16777251 318
 - 5
 -
 -
-

ID	Interface	Function	Reference
16777251	S6a/S6d	MME/SGSN ↔ HSS Authentication	3GPP TS 29.272
16777238	Gx	PCEF ↔ PCRF Policy Control	3GPP TS 29.212
16777216	Cx	I-CSCF/S-CSCF ↔ HSS IMS	3GPP TS 29.229

Interface	Function	Reference	Protocol
318	AIR/AIA (Authentication)	S6a	3GPP TS 29.272

- Prometheus /metrics endpoint for monitoring
- InfluxDB for time-series data storage
- OmniDRA uses BEAM/Erlang for high performance
- rate() function for monitoring request rates
- IMSI (International Mobile Subscriber Identity) for user identification

- - Authentication and authorization
- - AVP (Authentication Vector Parameters) for signaling

OmniDRA

OmniDRA is a Diameter peer configuration tool for OpenWrt. It is designed to be easy to use and to be integrated with the OpenWrt system.

Usage

OmniDRA can be used to configure a Diameter peer. The configuration is done by editing the `peer.conf` file. The file is located in the `/etc/omnidra/` directory. The file contains the configuration for the peer, including the peer name, the peer IP address, the peer port, the peer type, the peer timeout, the peer watchdog timer, and the peer peer settle time.

OmniDRA can also be used to configure a Diameter peer. The configuration is done by editing the `peer.conf` file. The file is located in the `/etc/omnidra/` directory. The file contains the configuration for the peer, including the peer name, the peer IP address, the peer port, the peer type, the peer timeout, the peer watchdog timer, and the peer peer settle time.

Configuration

The configuration file is located in the `/etc/omnidra/` directory. The file is named `peer.conf`. The file contains the configuration for the peer, including the peer name, the peer IP address, the peer port, the peer type, the peer timeout, the peer watchdog timer, and the peer peer settle time.

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

項目	単位	初期値	説明
peer_settle_time_ms	秒	0	DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。0は、接続確立後に、宛先ピアに到達するまでに許容される最大時間。
watchdog_timer_ms	秒	30000	RFC 3539で定義されたTw（Watchdog Timer）の値。DWR/DWA（Downstream Watchdog/Timer）は、DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。

DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。

設定

peer_connected, settling before routable
peer disconnected during settle window; never became routable
destination peer is settling; request not yet routable...
no suitable peer to route request...

DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。

状態	説明
peer connected, settling before routable	DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。
peer disconnected during settle window; never became routable	DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。
destination peer is settling; request not yet routable...	DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。 DIAMETER_UNABLE_TO_DELIVER (3002)
no suitable peer to route request...	DRAGRPC接続確立後に、宛先ピアに到達するまでに許容される最大時間。

□□□□□□□□□□□□□□□□□□□□

□□

- RFC 3539 - Diameter□□□□□□□□□□Tw□□
- RFC 6733 - Diameter□□□□CER/CEA□DWR/DWA□□

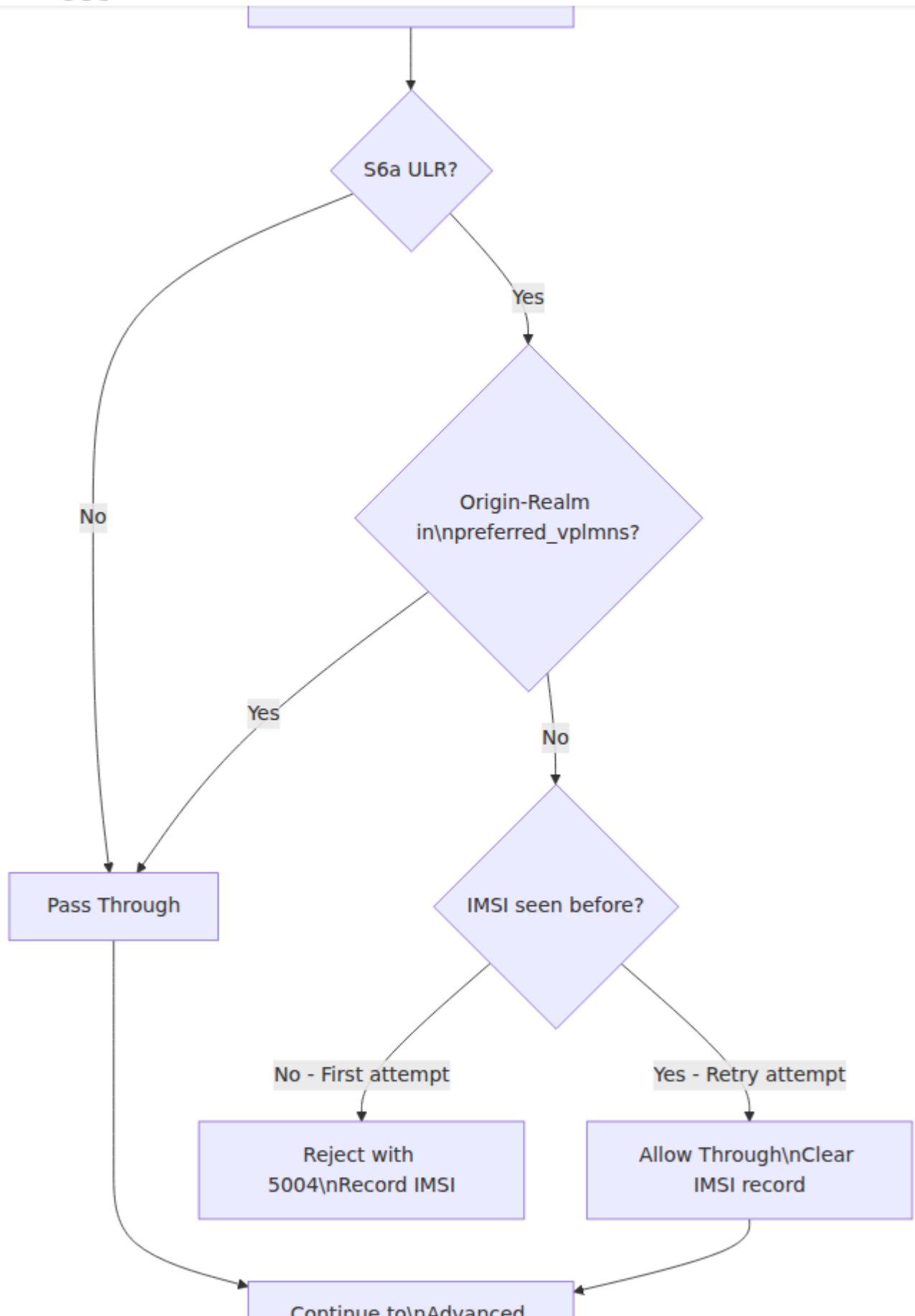
SoR (SoR)

SoR (SoR) HPLMN VPLMN VPLMN DRA S6a (ULR) **DIAMETER_ERROR_ROAMING_NOT_ALLOWED (5004)** ULR UE VPLMN UE

VPLMN

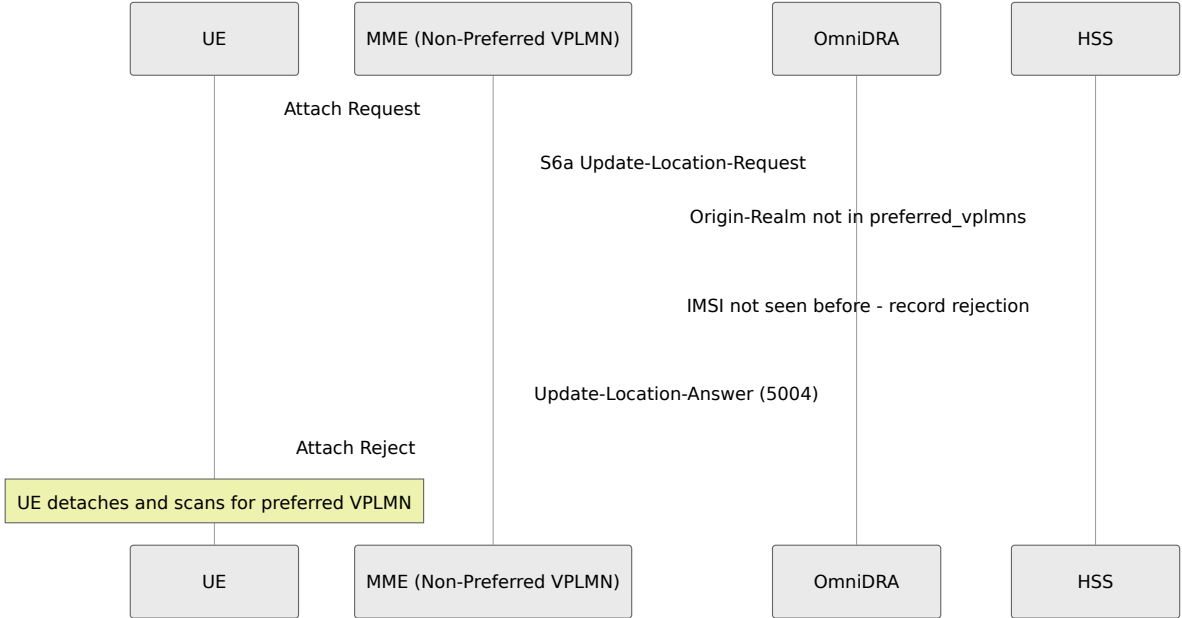
3GPP TS 29.272 5.2.1.1 3GPP TS 23.122 SoR

DRA S6a (ID 16777251 316) Diameter

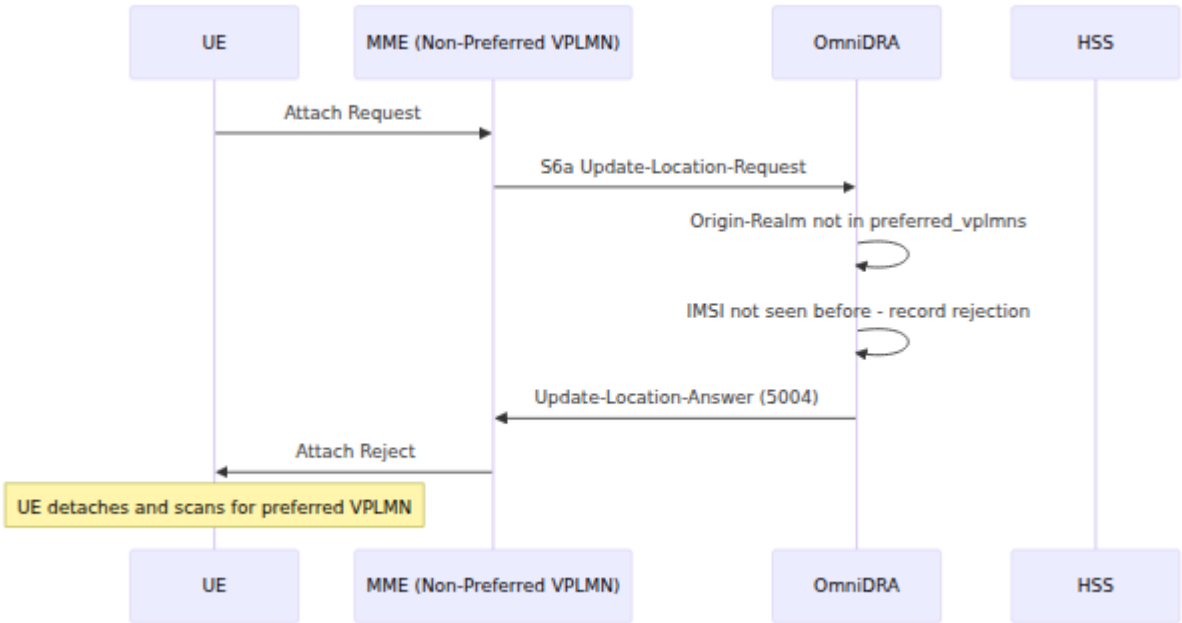


Continue to Advanced Routing

□□□□□□□□□□□□□□



□□□□□□□□□□□□□□



□□□□

□□□□□□□□□□□□□□□□□□ SoR □□□□□□□□□□□□□□□□□□

11

項目	型	デフォルト値	説明
enabled	boolean	false	roaming 機能を有効にするかどうか。 false なら無効。
children	list	[]	DRA の roaming 機能のモジュール。 [DRA.Router.Modules.RoamingSt IMSI 番号で指定。
max_rejections	int	1	roaming 機能の最大拒否回数。 1 以上。
rejection_ttl_seconds	int	300	roaming 機能の IMSI 番号の拒否有効時間。 0 以上。
rejection_result_code	int	5004	roaming 機能の Diameter 拒否コード。 5004 は DIAMETER_ERROR_ROAMING_NO 3GPP TS 29.272 の 7.4.4 節。
preferred_vplmns	list	[]	roaming 機能の VPLMN の Origin-Realm 番号。 Origin-Realm 番号は ULR の

□ □ □ □

□ □ □ □ □ □ □ □

□□□□□□□□□□ VPLMN □□□□□□□□□□□□□□□□□□ ULR □□□ 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"
  ]
}
```

UE が epc.mnc003.mcc310.3gppnetwork.org の ULR を検索する際に、
5004 のエラーコードを返す UE は、300 秒の間 VPLMN の ULR を

検索しない。

例

UE が VPLMN を検索する。

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

UE は、600 秒の間 VPLMN を検索しない。

UE が VPLMN を検索する。

□□

□□

□□: `diameter.roaming_steering.reject.count` □□: □□□ □□: □□ SoR □□□□ ULR □□
DRA □□□□□□ □□:

- `origin_realm` - □□□ VPLMN □ Origin-Realm
- `result_code` - □□□□□□ Diameter □□□□
- `imsi` - □□□□□□ IMSI

□□□□□ **VPLMN**□

□□: `diameter.roaming_steering.allow.count` □□: □□□ □□: □□□□□ ULR □□□□DRA
□□□□□□ □□:

- `origin_realm` - □ VPLMN □ Origin-Realm
- `reason` - □□□□□□□□□□ `preferred_vplmn` □ `max_rejections_reached`
- `imsi` - □□□ IMSI□□□□□□ `max_rejections_reached` □□□

□□□□

□□: `diameter.roaming_steering.error.count` □□: □□□ □□: □□□□□□ SoR □□□□□□□
□□□□DRA □□□□□□ □□:

- `result_code` - Diameter □□□□
- `application_id` - Diameter □□□□ ID□□□□
- `cmd_code` - Diameter □□□□□□□□
- `application_name` - □□□□□□□□□□
- `cmd_name` - □□□□□□□□

□□□□

□□□□□□□□□□□□□□

□□: □□□□□□□□ VPLMN □□□□□□□□□□□□

□□□□:

- rejection_ttl_seconds □□□□□□□□□□□□□□
- max_rejections □□ UE □□□□□□□□□□□□

□□□□:

1. □□ rejection_ttl_seconds □□□ UE □□□□□□
2. □□ max_rejections □ UE □□□□□□□□□□□□

□□ **VPLMN** □□□□□□□

□□: □□□ 5004 □□□□□□□□□□ ULR□

□□□□:

- □□ VPLMN □ Origin-Realm □□ preferred_vplmns □□
- Origin-Realm □□□□□□□□□□□□□□□□

□□□□:

1. □ DRA □□□□□□□□ Origin-Realm □□□□□□□□
2. □□□□□ Origin-Realm □□□□ preferred_vplmns □□□□

□□□□□□

□□: □□□□□□□□□□□□ VPLMN □ ULR□

□□□□:

- enabled □□□ false□
- □□□□□□□□□□□□

□□□□:

1. `enabled: true`
2. `DRA`

11

Diameter

错误码	错误名称	错误描述	参考
5004	DIAMETER_ERROR_ROAMING_NOT_ALLOWED	未知未知错误码 未知 VPLMN 未知	3GPP TS 29.272 7.4.4

S6a ☐ ☐

번호	구분	내용	참고
316	구분/구분 (ULR/ULA)	MME 구분구분구분구분구분 구분구분	3GPP TS 29.272 7.2.3

□□□□ ID

ID	IP	IP	IP
16777251	S6a/S6d	MME/SGSN ↔ HSS 10.10.10.10	3GPP TS 29.272

Rx [illegible]

11

3GPP TS 29.214 §5.6.5

Destination-Host Rx Session-Termination-Request Subscription-Id Framed-IP-Address AVP AF

STR Destination-Host AF AF AVP

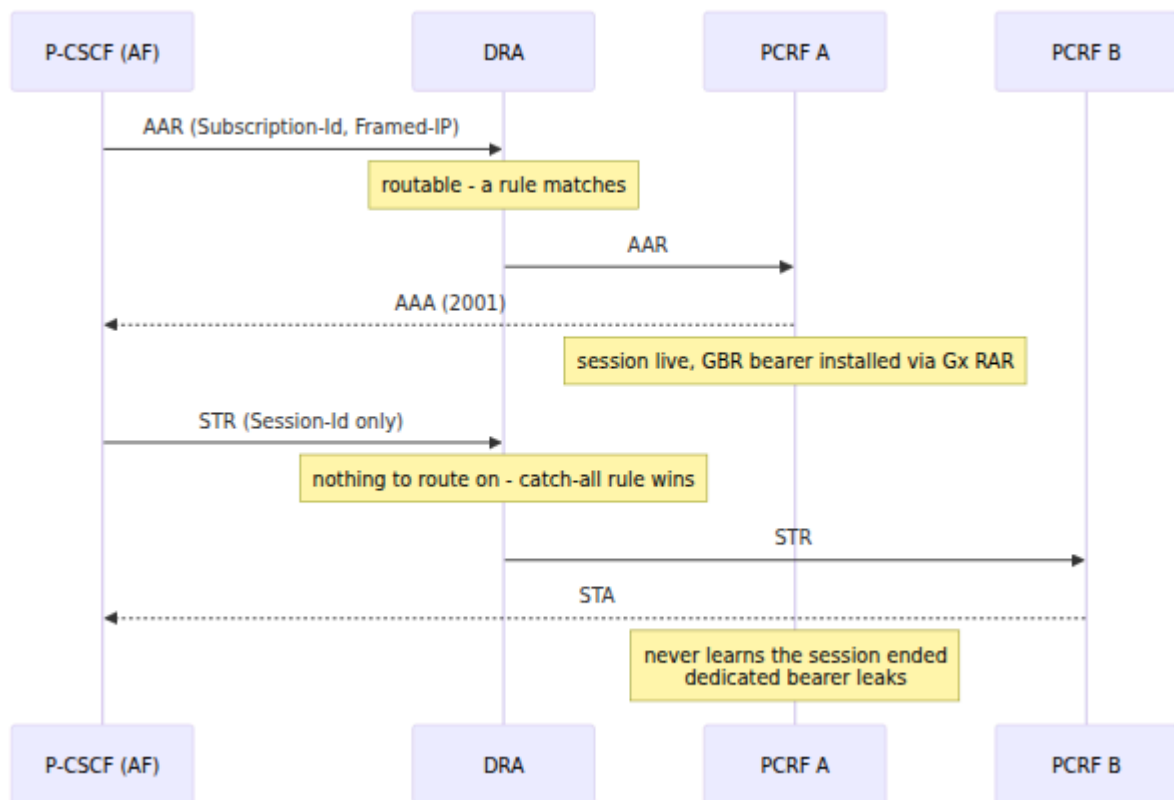
```
Session-Id, Origin-Host, Origin-Realm, Destination-Realm, Auth-Application-Id, Vendor-Specific-Application-Id, AF-Application-Identifier, Termination-Cause
```

Session-Id AF RFC 6733 §8.8

AAA-Request Subscription-Id Framed-IP-Address
 Gx CC-Request ABNF (TS 29.212 §5.6.2) [Destination-Host] *[Subscription-Id] [Framed-IP-Address]
 Gx

TS 29.213 §7.3.5 00000000 DRA 00000“0000000000000000 DRA”000000“00000”0

PCRf DRA PCRf



RFC 6733 §8.4.2 STR §8.1 STR “ STA “ STA “ 2001 PCRF (5002 DIAMETER_UNKNOWN_SESSION_ID ASA §8.5.2 STA)

peers: [a, b] AAR STR

1. PCRF

- PCRF PCRF
- Rx

PCRF

--	--	--	--	--	--	--

UE	Network
AA-Answer (265, Result-Code 2001)	UE Session-Id → PCRF
AA-Request (265)	PCRF
Session-Termination-Request (275)	PCRF
Session-Termination-Answer (275)	PCRF UE Result-Code (2xxx ~ 5xxx)

[illegible]

DRA □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ DRA // □

- AAR AAR
- DRA STA STR RFC 6733 §5.5.4
- PCRF

STA 2xxx 5xxx 3xxx/4xxx AF

STR PCRF PCRF
2001 DRA
insert_new PCRF

--	--	--	--	--	--	--	--

[illegible]

項目	説明
enabled	module_subscriber_lookupが有効かどうかを示すフラグ
ttl_seconds	STA が登録されている有効期限 (秒)

ttl_seconds は AF の Authorization-Lifetime より DRA の Authorization-Lifetime: 36000 秒 P-CSCF の 10 秒12 秒のいずれか
 の ttl_seconds / 2 の値と 60s の DRA の 1.5 × TTL の値の大きい方を用いる

4.2.2

session_id

操作	対象
[:diameter, :rx_session_binding, :bind]	serving_host
[:diameter, :rx_session_binding, :route]	serving_host
[:diameter, :rx_session_binding, :release]	-

[:diameter, :rx_session_binding, :reap] 操作は count DRA の TTL が満了した STA を削除する

DRA は {session_id, origin_host, inserted_at}

4.2.3

- DRA は :one_for_all を用いて、すべての DRA に対して、DRAs の AF の DRA
- **DRA** は DRA の Route-Record Destination-Host (RFC 6733 §6.1.9) DRA の PCRF

- 發送 **AA-Answer** 訊息給 MME 的 DRA 向 STA 發送 AAR 訊息時 `insert_new` 訊息中會包含 MME 的 ID 號碼 (**RFC 6733 §8.8**)
`ttn_seconds` 訊息中會包含 `:reap` 訊息
- **DRA** 會向 MME 的 Rx 訊息中包含 `{:answer_message, 3002}` 訊息
`route: {:error, 3002}` 訊息中會包含 MME 的 PCRF 訊息中會包含 MME 的 ID
- 訊息中 `children` 訊息 DRA.Supervisor 訊息中 `children` 訊息中會包含 `enabled: true` 訊息 `children` 訊息 GenServer 訊息中會包含 "MME" 訊息
- **DRA** 會向 MME 的 AF → PCRF 訊息 PCRF 訊息 RAR 訊息 ASR 訊息中 `Destination-Host` (**RFC 6733 §8.3.1, §8.5.1**) 訊息中會包含 MME 的 ID
- **DRA** 會向 MME 的 Rel-7+ Rx 訊息 ID (**16777236**) 訊息中會包含 Rel-6 Rx/Gq ID (16777229) 訊息
- **DRA** 會向 MME 的 Telemetry.Metrics 訊息中會包含 MME 的 Telemetry.Metrics 訊息 Prometheus 訊息中會包含 `subscriber_lookup` 訊息中會包含 MME 的 shell 訊息

SCTP

SCTP 是 Diameter 的可靠传输层，基于 IP 的多播传输。SCTP 是 Diameter 的可靠传输层。

部署

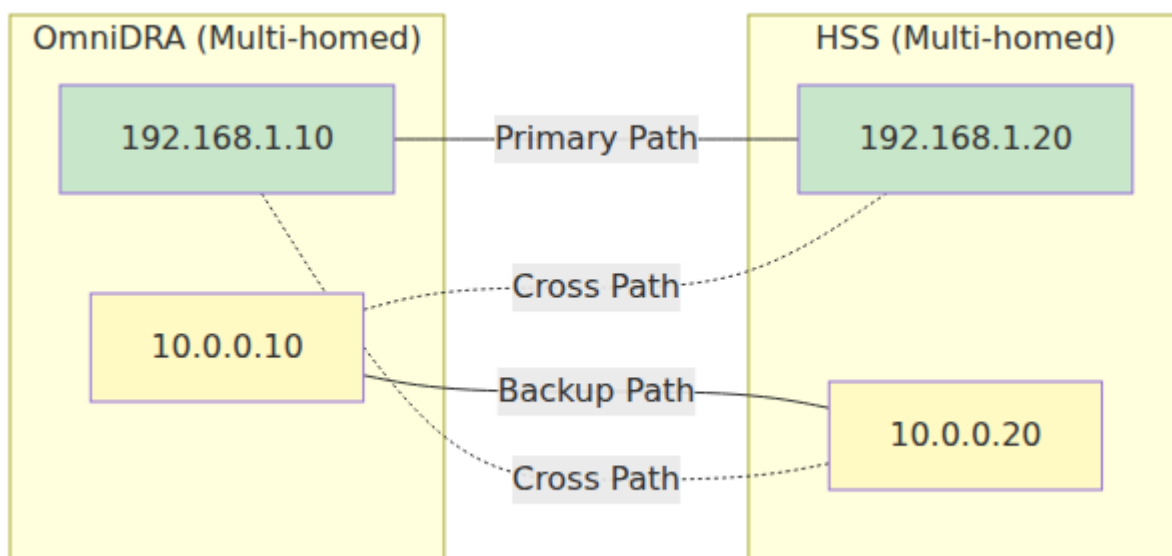
RFC 6733 定义了 SCTP 与 Diameter 的部署。SCTP 部署遵循 **RFC 4960** 定义。TCP 部署遵循 /IP 部署。SCTP 部署遵循 IP 部署。OmniDRA 部署遵循 NIC 部署。Diameter 部署遵循 Diameter 部署。

部署

- SCTP 部署遵循 RFC 4960 定义。
- 部署遵循 /IP 部署。
- 部署遵循 Diameter 部署。
- 部署遵循 Diameter 部署。

部署

OmniDRA 部署遵循 IP 部署。IP 部署遵循 IP 部署。IP 部署遵循 IP 部署。



OmniDRA SCTP IP

IP

config/runtime.exs OmniDRA IP

DRA

listen_ip OmniDRA IP IP IP

```
%{
  # IP
  listen_ip: "192.168.1.10",

  # IP SCTP
  listen_ip: ["192.168.1.10", "10.0.0.10"],

  listen_port: 3868
}
```

listen_ip			-	OmniDRA IP SCTP TCP OmniDRA IP SCTP OmniDRA IP
listen_port			-	Diameter 3868 RFC 6733

IP

配置

配置文件中定义了 `ip` 和 `additional_ips` 两个属性，用于指定对等体的 IP 地址。

```
peers: [  
  {%  
    host: "hss01.example.com",  
    realm: "example.com",  
    ip: "192.168.1.20", # 主 IP 地址  
    additional_ips: ["192.168.1.20", "10.0.0.20"], # 备用 IP 地址  
    port: 3868,  
    transport: :diameter_sctp,  
    initiate_connection: true  
  }  
]
```

項目	項目名	項目型	必須	説明
ホスト名	host	文字列	必須	Diameter サーバの FQDN を指定します。
ドメイン名	realm	文字列	必須	Diameter サーバの DRA を指定します。
IP アドレス	ip	文字列	必須	IP アドレスを指定します。
追加の IP アドレス	additional_ips	文字列	任意	SCTP を使用する IP アドレスを指定します。OmniDRA の ip または SCTP の IP または TCP の OmniDRA の TCP を指定します。
ポート番号	port	整数	必須	Diameter サーバのポート番号を 3868 または RFC 6733 で指定します。
トランスポートプロトコル	transport	文字列	任意	transport プロトコルを指定します。:diameter_sctp、:diameter_tcp、:diameter_sctp を指定します。
接続の開始	initiate_connection	ブール値	任意	true を指定すると OmniDRA に接続を開始します。false を指定すると OmniDRA に接続しません。

ip アドレスを指定します。additional_ips は追加の IP アドレスを指定します。

配置

配置 OmniDRA 与 HSS 和 MME 的连接

```
config :dra,
  diameter: %{
    service_name: :omnitouch_dra,
    listen_ip: ["192.168.1.10", "10.0.0.10"], # 配置 DRA
    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: [
      # 配置 HSS 连接
      %{
        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
      },
      # 配置 MME 连接
      %{
        host: "mme01.example.com",
        realm: "example.com",
        ip: "192.168.1.30",
        port: 3868,
        transport: :diameter_sctp,
        initiate_connection: false
      }
    ]
  }
}
```

配置 listen_ip 和 listen_port 为 OmniDRA 与 Diameter 连接

SCTP 設定

OmniDRA 上で SCTP を有効にするには、以下の設定が必要です。

`diameter_socket_buffer_size` を 4 MB に、`listen_ip` と `peers` を設定し、`:dra` を `diameter:` に設定します。

SCTP を有効にするには、`SO_RCVBUF` を SCTP の MTU に設定し、Linux 上で `SO_RCVBUF` を `net.core.rmem_max` と `net.core.wmem_max` に設定し、`sysctl` を実行します。

```
config :dra,
  # ... - ... diameter: ... SCTP ...
  diameter_socket_buffer_size: 4 * 1024 * 1024, # 4 MB
  ...
  diameter: %{
    # ...
  }
```

項目	値	単位	説明
<code>diameter_socket_buffer_size</code>	4194304	(4 MB)	SCTP の <code>recbuf</code> と <code>sndbuf</code> のサイズを TCP の <code>net.core.rmem_max/net.core.wmem_max</code> に設定し、 <code>sysctl</code> を実行します。

確認

SCTP が有効になっていることを確認します。

- Linux 上で `lksctp-tools` をインストールします。
- IP アドレスを `listen_ip` に設定します。
- IP アドレスを `peers` に設定し、SCTP を有効にします。
- OmniDRA の `dra` を `diameter:` に設定し、OmniDRA を再起動します。

11

- **TCP** :diameter_tcp OmniDRA IP listen_ip
IP OmniDRA additional_ips
- Sctp OmniDRA

--	--	--	--

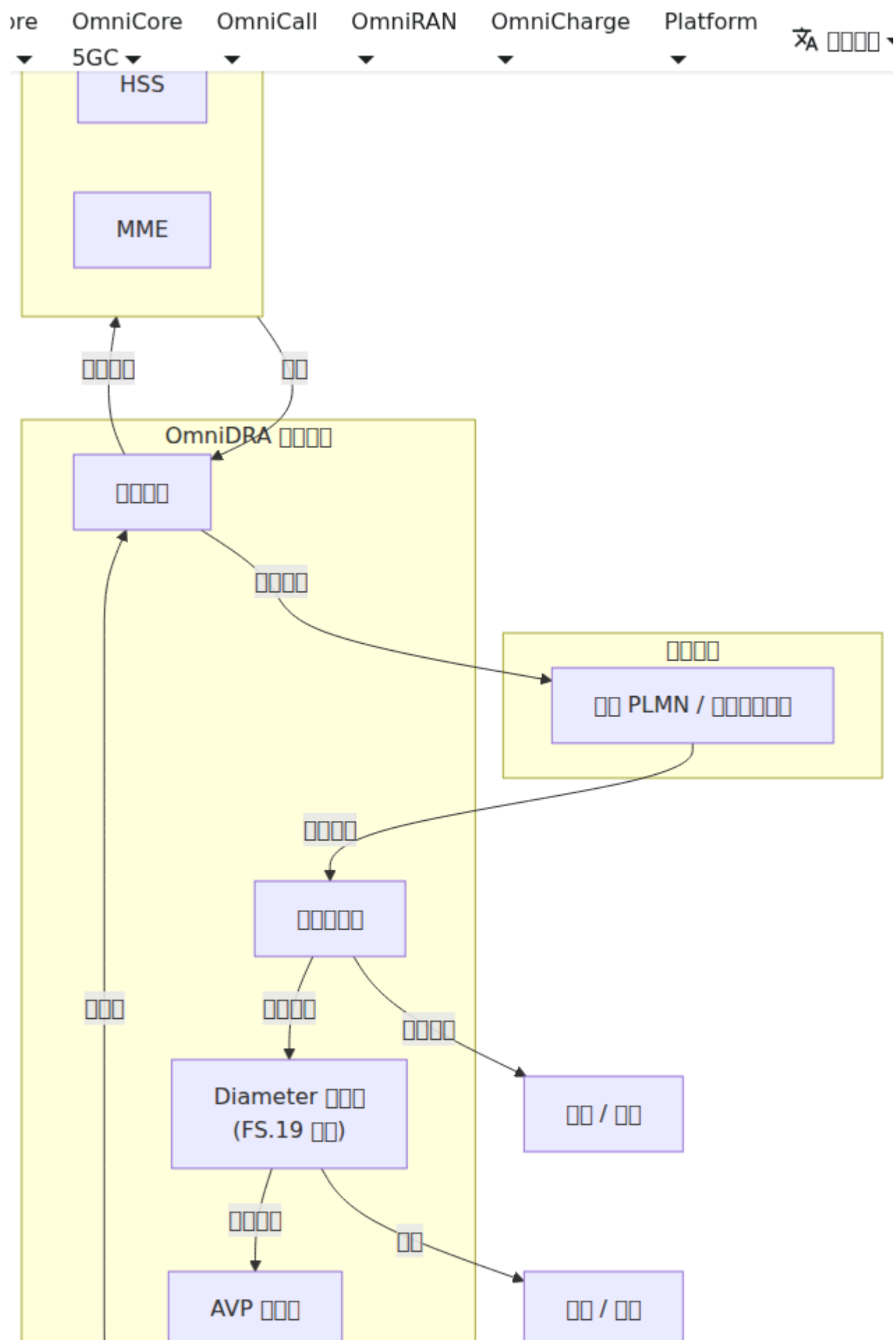
- RFC 4960 - SSH (SCTP)
- RFC 6733 - Diameter RADIUS
- RADIUS

Diameter

OmniDRA  Diameter  **GSMA FS.19**  Diameter  v10.0 
 **GSMA FS.21**  v12.0  Diameter 





消息	消息	Diameter 消息
<code>{:error, 3002}</code>	向 Diameter 服务器 发送	DIAMETER_UNABLE_TO_DELIVER
<code>{:error, 3004}</code>	向 Diameter 服务器 发送	DIAMETER_TOO_BUSY
<code>{:error, 3007}</code>	向 Diameter 服务器 发送	DIAMETER_APPLICATION_UNSUPPORTED
<code>:drop</code>	丢弃	丢弃
<code>:log_only</code>	只记录日志	只记录

Diameter 消息

Diameter 消息规范 **GSMA FS.19** 中 **3.3** 中详细描述了消息规范 **FS.21** 中 **7** 中详细描述了消息规范



消息规范

GSMA 消息 FS.19 中 3.3.4 中 FS.21 中 7.3.1 中

消息规范详细描述了消息规范详细描述了消息规范详细描述了消息规范详细描述了消息规范详细描述了消息规范

配置

項目	FS.19 項目	説明
AVP 項目	3.3.4 4.8.1	Origin-Host, Origin-Realm AVP 項目
Session-Id 項目	3.3.4	Session-Id AVP 263 RFC 6733 8.8
Destination-Host	3.3.4	Destination-Host AVP 項目
AVP 項目	3.3.5	AVP 項目 ULR 項目 User-Name Visited-PLMN-Id 3GPP TS 29.272 7.2.3

```
low_layer: %{\n  enabled: true,\n  action: {:error, 3002},\n  # AVP 項目FS.19 3.3.4 項目\n  # 264 = Origin-Host, 296 = Origin-Realm, 283 = Destination-Realm, 293 = Destination-Host\n  single_instance_avps: [264, 296, 283, 293],\n  # Session-Id 項目 AVPRFC 6733 8.8 項目\n  enforce_session_id_first: true,\n  # 項目 Destination-HostFS.19 3.3.4 項目\n  reject_destination_host_in_answers: true,\n  # 項目 AVP3GPP TS 29.272\n  # 1 = User-Name (IMSI), 1407 = Visited-PLMN-Id, 264 = Origin-Host, 296 = Origin-Realm\n  mandatory_avps: %{\n    316 => [1, 1407, 264, 296],\n    318 => [1, 1407, 264, 296]\n  }\n}
```

00 **1** - 0000000000

GSMA □□□ FS.19 □ 3.3.5 □□FS.21 □ 7.2.1 □

00 1 Diameter ID
S6a Sh

FS.19 **Diameter**

[illegible]

```
category_1: %{
  enabled: true,
  action: {:error, 3007},
  whitelists: %{
    # 中国移动 - 中国移动 ULR 3 AIR FS.19 3.3.5
    "restricted-partner.roaming.com" => %{
      16_777_251 => [316, 318]
    },
    # 中国移动 - 中国移动 S6a
    all: %{
      # S6a/S6d (FS.19 3.3.5 2)
      16_777_251 => [316, 317, 318, 319, 320, 321, 323],
      # S13 - ME 中国移动 (FS.19 3.3.5)
      16_777_252 => [324],
      # S6c - 中国移动 HSS 3 SMS (FS.19 3.3.5.1)
      16_777_312 => [8388647, 8388648],
      # SGd - 中国移动 MME 3 SMS (FS.19 3.3.5.2)
      16_777_313 => [8388645, 8388646]
    }
  }
}
```

□□ **S6a/S6d** □□□□

Sequence	Interface	Direction	Reference
316	Uu/3GPP	MME → HSS	3GPP TS 29.272 §7.2.3
317	Uu/3GPP	HSS → MME	3GPP TS 29.272 §7.2.7
318	Uu/3GPP	MME → HSS	3GPP TS 29.272 §7.2.5
319	Uu/3GPP	HSS → MME	3GPP TS 29.272 §7.2.9
320	Uu/3GPP	HSS → MME	3GPP TS 29.272 §7.2.11
321	Uu/3GPP UE	MME → HSS	3GPP TS 29.272 §7.2.13
323	Uu/3GPP	MME → HSS	3GPP TS 29.272 §7.2.15

Sequence ID FS.19 3.3.5

Sequence ID	Interface	Reference
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

2 - Sequence

GSMA FS.19 3.3.6 FS.21 7.2.2

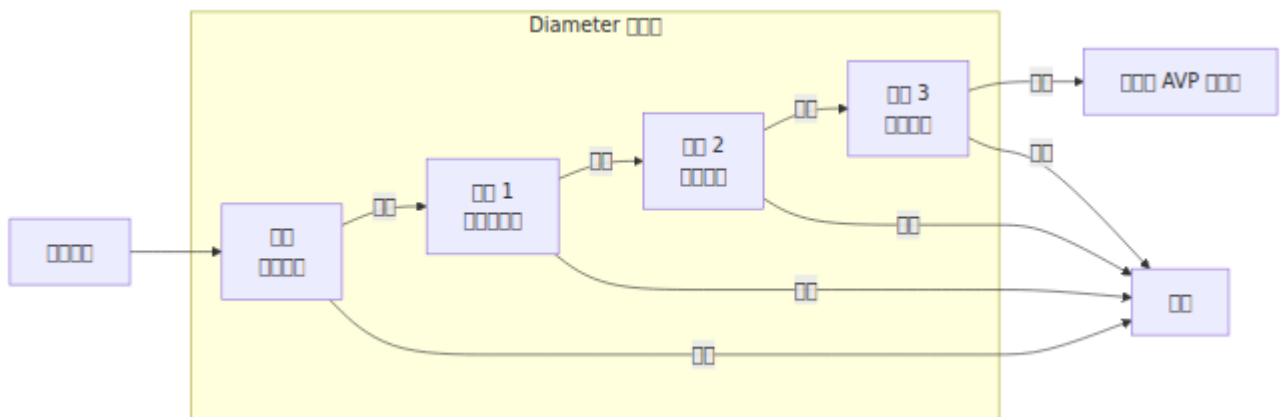
2 IMSI User-Name
 AVP MSISDN AVP 701

```

#
home_imsi_prefixes: ["31338", "31339"],
home_msisdn_prefixes: ["+1313"],

category_2: %{
  enabled: true,
  action: {:error, 3002},
  # S6a FS.19 3.3.6 12
  protected_command_codes: [316, 317, 318, 319, 320, 321, 323]
}

```



3 -

GSMA FS.19 3.3.7 FS.21 7.2.3

3 IMSI

ETS IMSI 24

項目	FS.19 項目	説明
設定項目	3.3.7.1 項目	ULR の Origin-Realm 及び IMSI の Origin-Realm
速度/更新 間隔	3.3.7.2 項目	

```
category_3: %{
  enabled: true,
  # log_only 項目FS.19 3.3.7 項目
  action: :log_only,
  # 速度/更新間隔FS.19 3.3.7.2 項目
  max_velocity_kmh: 1200,
  # IMSI 項目
  min_time_between_updates_seconds: 2
}
```

項目

項目 Diameter 項目
module_diameter_firewall 項目

```
config :dra,
  module_diameter_firewall: %{
    enabled: true,
    home_imsi_prefixes: ["31338", "31339"],
    home_msisdn_prefixes: ["+1313"],
    low_layer: %{ ... }, # 項目
    category_1: %{ ... }, # 項目 1
    category_2: %{ ... }, # 項目 2
    category_3: %{ ... } # 項目 3
  }
```

配置

配置	是否	是否	配置	配置
<code>enabled</code>	是否	是否	<code>false</code>	是否启用 Diameter 配置 <code>false</code>
<code>children</code>	是否	是否	<code>[]</code>	DRA 配置项 <code>[DRA.Router.Modules.DiameterF</code> 配置 3 个配置项
<code>home_imsi_prefixes</code>	是否	是否	<code>[]</code>	配置 IMSI 配置 2 个配置
<code>home_msisdn_prefixes</code>	是否	是否	<code>[]</code>	配置 MSISDN 配置 2 个配置

配置

配置	是否必填	是否只读	默认值	说明
<code>enabled</code>	是	否	<code>true</code>	是否启用该策略
<code>action</code>	是	否	<code>{:error, 3002}</code>	策略执行失败时的错误码
<code>single_instance_avps</code>	是	否	<code>[264, 296, 283, 293]</code>	策略生效的 AVP 列表 Origin-Host (264) Origin-Realm (296) Destination-Realm (283) Destination-Host (293)
<code>enforce_session_id_first</code>	是	否	<code>true</code>	是否 Session-Id 在 AVP 列表第一位
<code>reject_destination_host_in_answers</code>	是	否	<code>true</code>	是否拒绝 Destination-Host AVP 在答案中
<code>mandatory_avps</code>	是	否	<code>%{}</code>	策略生效的 AVP 列表 格式为 <code>%{316 => [1, 1407, 264, 296]}</code>

URI	URI Scheme	URI Host	URI Path	URI Query
				URI ULR URI URI User- Name URI Visited- PLMN-Id URI Origin-Host URI Origin- Realm

URI 1 URI

URI	URI Scheme	URI Host	URI Path	URI Query
enabled	URI	URI	true	URI 1 URI
action	URI	URI	{:error, 3007}	URI 1 URI DIAMETER_APPLICATION_UNSUPPORTED
whitelists	URI	URI	-	URI :all URI ID / URI URI :all URI

Table 2

Item	Item	Item	Item
enabled	Item	Item	Item 2 Item
action	Item	Item	Item Item
protected_command_codes	Item	Item	Item Item S6a Item Item

3

enabled			false	LocationTracker
action			:log_only	:log_only
max_velocity_kmh			1200	
min_time_between_updates_seconds			2	IMSI

配置

配置	说明
<code>[:diameter, :firewall, :low_layer, :block]</code>	低层防火墙配置
<code>[:diameter, :firewall, :category_1, :block]</code>	第一类防火墙配置
<code>[:diameter, :firewall, :category_2, :block]</code>	第二类防火墙配置
<code>[:diameter, :firewall, :category_3, :block]</code>	第三类防火墙配置
<code>[:diameter, :firewall, :pass, :count]</code>	防火墙通过次数

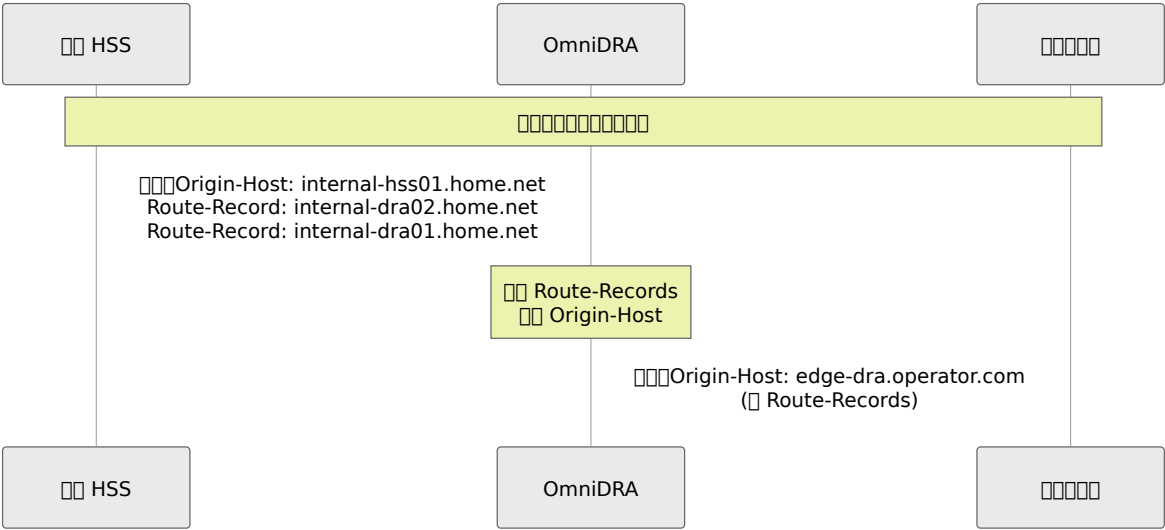
配置项包括：`origin_host`、`application_id`、`command_code`、`application_name`、`command_name`、`reason`

配置

GSMA 配置 FS.19 2.4 3.4 FS.21 3.6

Diameter 配置 Route-Record AVP Origin-Host/Origin-Realm 配置

DRA 配置



Table

Item	FS.19 Item	Item
Route-Record Item	Item 2.4	Route-Record AVP Item 282
Origin-Host Item	Item 3.4	Origin-Host AVP Item DRA Item
Origin-Realm Item	Item 3.4	Origin-Realm AVP Item
Item	-	Item

```
module_topology_hiding: %{
  enabled: true,
  # 删除 Route-Record AVPFS.19 2.4
  strip_route_records: true,
  # 删除 Origin-Host 属性FS.19 3.4
  rewrite_origin_host: %{enabled: true, replacement: :self},
  # 删除 Origin-Realm
  rewrite_origin_realm: %{enabled: false, replacement: :self},
  # 删除所有外部对等体
  external_peers: :all
}
```

配置

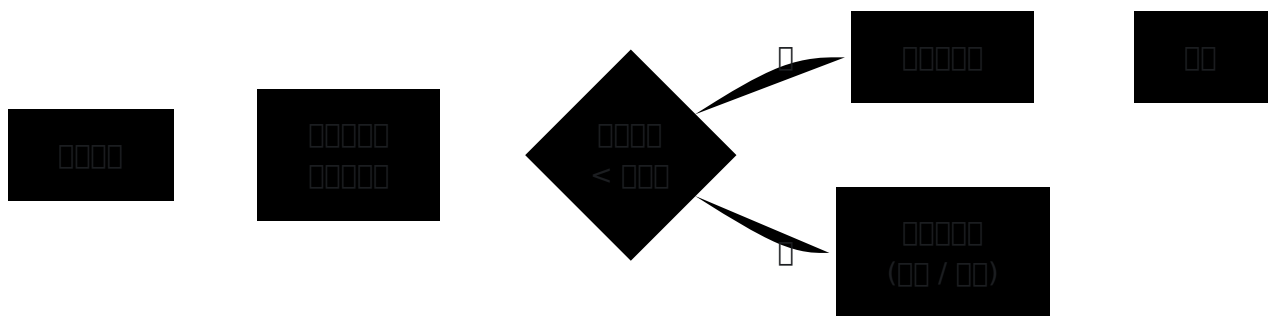
配置项	配置值	是否生效	配置值	配置说明
enabled	是	否	false	是否启用
strip_route_records	是	否	true	是否删除 DRA 属性 Route-Record AVP 282
rewrite_origin_host	是	否	是	是否重写 Origin-Host 属性
rewrite_origin_realm	是	否	是	是否重写 Origin-Realm 属性
external_peers	:all 所有	否	:all	是否删除所有外部对等体 :all 删除所有外部对等体

配置项 rewrite_origin_host 和 rewrite_origin_realm


```

module_rate_limiter: %{
  enabled: true,
  # 配置FS.19 3.4
  default_max_requests_per_second: 1000,
  default_action: {:error, 3004},
  # 配置
  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}
  }
}

```



enabled			false	
children			[]	DRA [DRA.Router.Modu]
default_max_requests_per_second			1000	
default_action			{:error, 3004}	DIAMETER_TOC
peer_limits			%{}	

max_requests_per_second				
action				

□ □ □ □

入参	出参
[:diameter, :rate_limiter, :throttled]	返回当前速率限制器 <code>current_rate</code> 和 <code>limit</code>
[:diameter, :rate_limiter, :allowed]	返回是否允许

AVP ☐ ☐ ☐

GSMA □□□ FS.19 □ 3.3.4 □□4.8.1□4.8.2□FS.21 □ 3.6 □

AVP 000000000000 Diameter AVP00000 FS.21 3.6 000000000000 FS.19 00000
00000000000000000000

11

項目	GSMA 項目	備考
AAA AVP の 値	FS.21 の 3.6 値	AAA の値は AVP の値と一致する AVP
AAA AVP の値	FS.21 の 3.6 値	AAA AVP の値は AVP の値と一致 する

```
module_avp_sanitizer: %{
  enabled: true,
  action: {:error, 3002},
  # AVP FS.21 3.6
  # 0 = IETF, 10415 = 3GPP, 13019 = ETSI, 5535 = 3GPP2
  allowed_vendor_ids: [0, 10415, 13019, 5535],
  strip_unknown_vendor_avps: true,
  # AVP FS.21 3.6
  max_avp_nesting_depth: 10
}
```

11

項目	型	デフォルト	説明
enabled	bool	false	AVP を有効にするかどうか
action	enum	{:error, 3002}	AVP のエラー発生時の動作
allowed_vendor_ids	list	[0, 10415, 13019, 5535]	ID が許可されている AVP
strip_unknown_vendor_avps	bool	true	許可されていない AVP を削除するかどうか
max_avp_nesting_depth	int	10	AVP のネストの最大深さ

□□□□ ID□

Vendor ID	Vendor	Notes
0	IETF	RFC 4006 Diameter AVP
10415	3GPP	3GPP TS 29.261 AVP S6a Gx Rx
13019	ETSI	ETSI TS 102 229 AVP
5535	3GPP2	3GPP2 TS 29.261 AVP CDMA

AVP

AVP Name	Notes
[:diameter, :avp_sanitizer, :unknown_vendor, :stripped]	AVP
[:diameter, :avp_sanitizer, :nesting_depth, :violation]	AVP
[:diameter, :avp_sanitizer, :pass, :count]	

AVP

AVP

1. AVP -
2. **AVP** -
3. **Diameter** - ID /
4. **Diameter** -
5. **Diameter** - IMSI/MSISDN

6. 配置 - 配置 Route-Record 配置 Origin-Host

7. **Diameter** 配置 3 - 配置 :log_only 配置

配置

配置 FS.19 3.4 配置 配置 配置

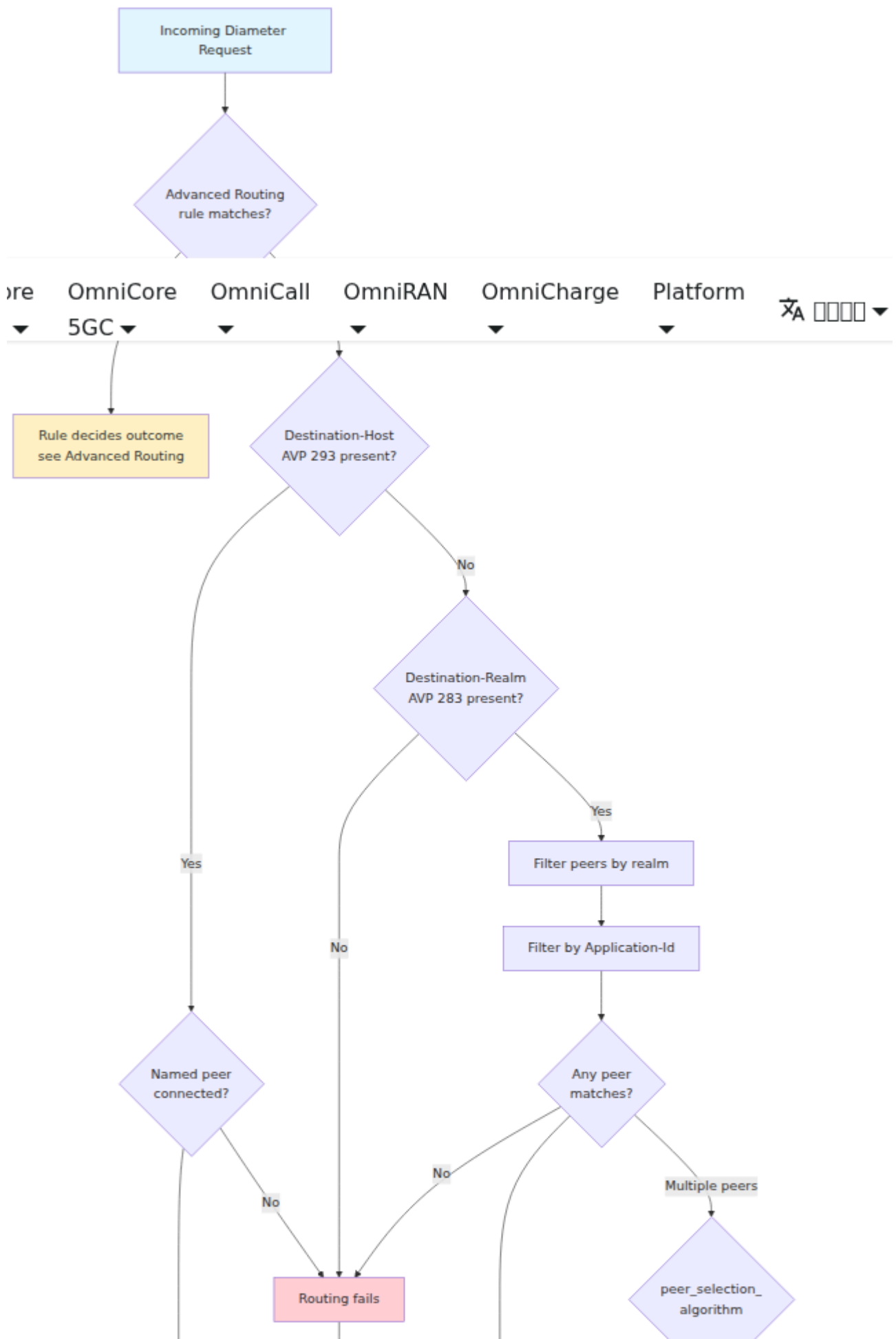
配置	配置	配置
配置 DoS	配置	Diameter 配置
配置	配置 1	AVP 配置
配置	配置 2	配置 1配置
配置	配置 3	配置 2配置
配置	配置	-
AVP 配置	AVP 配置	配置
配置AVP 配置	配置	AVP 配置

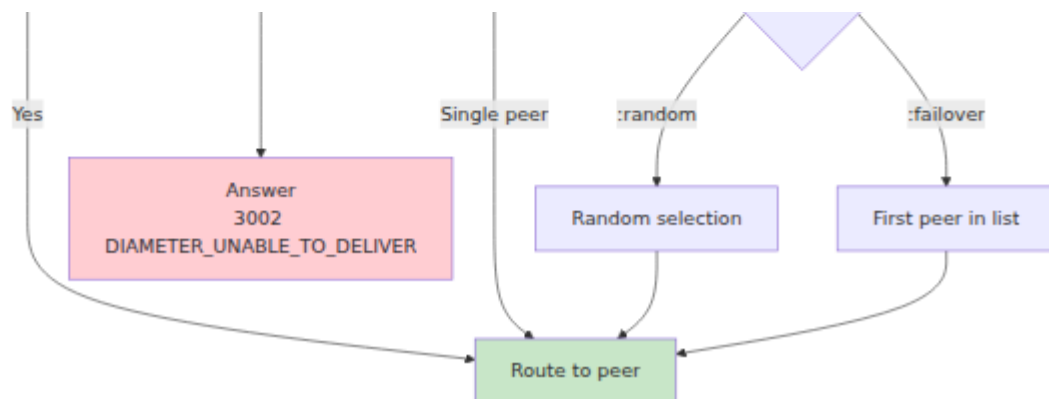
GSMA 000000

00 / 00	FS.19 v10.0	FS.21 v12.0
000000	0 3.3.4 0	0 7.3.1 0
00 1 00	0 3.3.5 0000 B.3.3	0 7.2.1 0
00 2 00	0 3.3.6 0000 B.3.4	0 7.2.2 000 16 0
00 3 0???	0 3.3.7 0000 B.3.5	0 7.2.3 0
0000	0 2.4 000 3.4 0	0 3.6 0
0000	0 3.4 0	-
AVP 00	0 4.8.1 004.8.2 0	0 3.6 0
0000	0 3.4 0	0 3.15 0
0000000000	00 A	0 7 0

(RFC 6733)







1. DRA가 Diameter_UNable_to_Deliver 메시지를 수신하면
 Destination-Host

2.

3. Application-Id에 peer_selection_algorithm이 설정되어 있는지 확인
 4.

```

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

항목	타입	기본값	필수	설명
peer_selection_algorithm	Atom	:random	필수	DRA가 Diameter_Unable_to_Deliver 메시지를 수신할 때 사용할 peer_selection_algorithm을 지정합니다. :random 또는 :failover로 설정할 수 있습니다.

- **:random** - DRA가 Diameter_Unable_to_Deliver 메시지를 수신할 때
 임의의 peer를 선택합니다.
- **:failover** - DRA가 Diameter_Unable_to_Deliver 메시지를 수신할 때
 첫 번째 peer를 선택합니다.

5. DRA가 Diameter_Unable_to_Deliver 메시지를 수신하면

--	--	--	--	--	--	--	--	--

Destination-Host
Application-Id AVP DRA

DRACARD Result-Code 3002 DIAMETER_UNABLE_TO_DELIVER RFC 6733

\$7.1.5 DRACARD

11

□□□□

番号	名前	説明	参照
2001	DIAMETER_SUCCESS	DRA正常終了	RFC 6733 §7.1.1
3002	DIAMETER_UNABLE_TO_DELIVER	DRAメッセージを宛先に 送れない	RFC 6733 §7.1.3

AVP

	AVP			
1	User-Name	UTF8String	(IMSI in 3GPP)	RFC 6733 §8.14
264	Origin-Host	DiameterIdentity		RFC 6733 §6.3
268	Result-Code	Unsigned32		RFC 6733 §7.1
283	Destination-Realm	DiameterIdentity		RFC 6733 §6.6
293	Destination-Host	DiameterIdentity	()	RFC 6733 §6.5
296	Origin-Realm	DiameterIdentity		RFC 6733 §6.4
297	Experimental-Result	Grouped		RFC 6733 §7.6

AVP

번호	코드	설명	참고
257	CER/CEA	인증서/인증서	RFC 6733 §5.3
258	RAR/RAA	인증서/인증서	RFC 6733 §8.3
274	ASR/ASA	인증서/인증서	RFC 6733 §8.5
275	STR/STA	인증서/인증서	RFC 6733 §8.4
280	DWR/DWA	인증서/인증서	RFC 6733 §5.5
282	DPR/DPA	인증서/인증서	RFC 6733 §5.4
316	ULR/ULA	인증서/인증서 (S6a)	3GPP TS 29.272
317	CLR/CLA	인증서/인증서 (S6a)	3GPP TS 29.272
318	AIR/AIA	인증서/인증서 (S6a)	3GPP TS 29.272
321	PUR/PUA	인증서/인증서 (S6a)	3GPP TS 29.272

3GPP ID

Application-IdsApplication-Id

Application-Id	接口	接口描述	参考
16777251	S6a/S6d	MME/SGSN ↔ HSS 鉴权/注册	3GPP TS 29.272
16777252	S13/S13'	MME ↔ EIR 鉴权/注册	3GPP TS 29.272
16777238	Gx	PCEF ↔ PCRF 策略控制	3GPP TS 29.212
16777267	S9	PCRF ↔ PCRF 策略控制	3GPP TS 29.215
16777272	S6b	PDN-GW ↔ 3GPP AAA 鉴权/注册 3GPP网络	3GPP TS 29.273
16777216	Cx	I-CSCF/S-CSCF ↔ HSS IMS 鉴权/注册	3GPP TS 29.229
16777217	Sh	AS ↔ HSS IMS 鉴权/注册	3GPP TS 29.329
16777255	SLg	MME/SGSN ↔ GMLC 鉴权/注册	3GPP TS 29.172
16777291	SLh	GMLC ↔ HSS 鉴权/注册	3GPP TS 29.173
16777302	Sy	PCRF ↔ OCS 策略控制	3GPP TS 29.219
16777310	S6m	MTC-IWF ↔ HSS/HLR 鉴权/注册 M2M网络	3GPP TS 29.336

Application-Id	API	API Description	3GPP TS
16777312	S6c	SMS-SC/IP-SM-GW ↔ HSS SMS API	3GPP TS 29.338
16777345	S6t	SCEF ↔ HSS API	3GPP TS 29.336
16777236	Rx	AF ↔ PCRF API	3GPP TS 29.214

API (Gy/Ro) API 3GPP TS 32.299

API

- API - API
- API - API Destination-Host API

SLF (SLF)

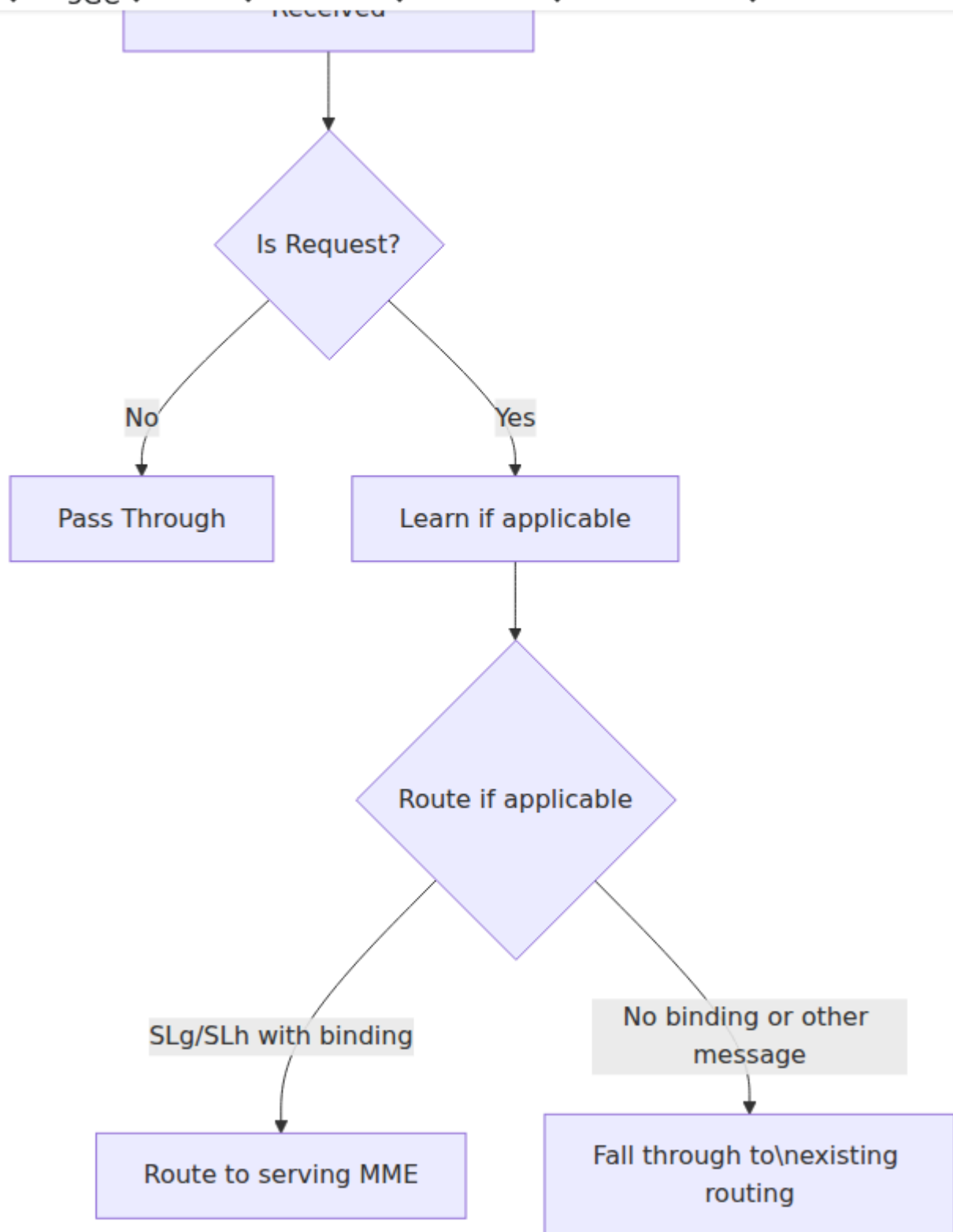
SLF (SLF) SLF DRA Diameter SLF

SLg/SLh GMLC MME MME

3GPP TS 29.172 3GPP TS 29.173 SLF 3GPP TS 29.272

DRA S6a Update-Location-Request

SLg Provide-Location-Request



□□□□□□



Subscriber unknown

NoBinding

Registration message
observed

Deregistration message
observed

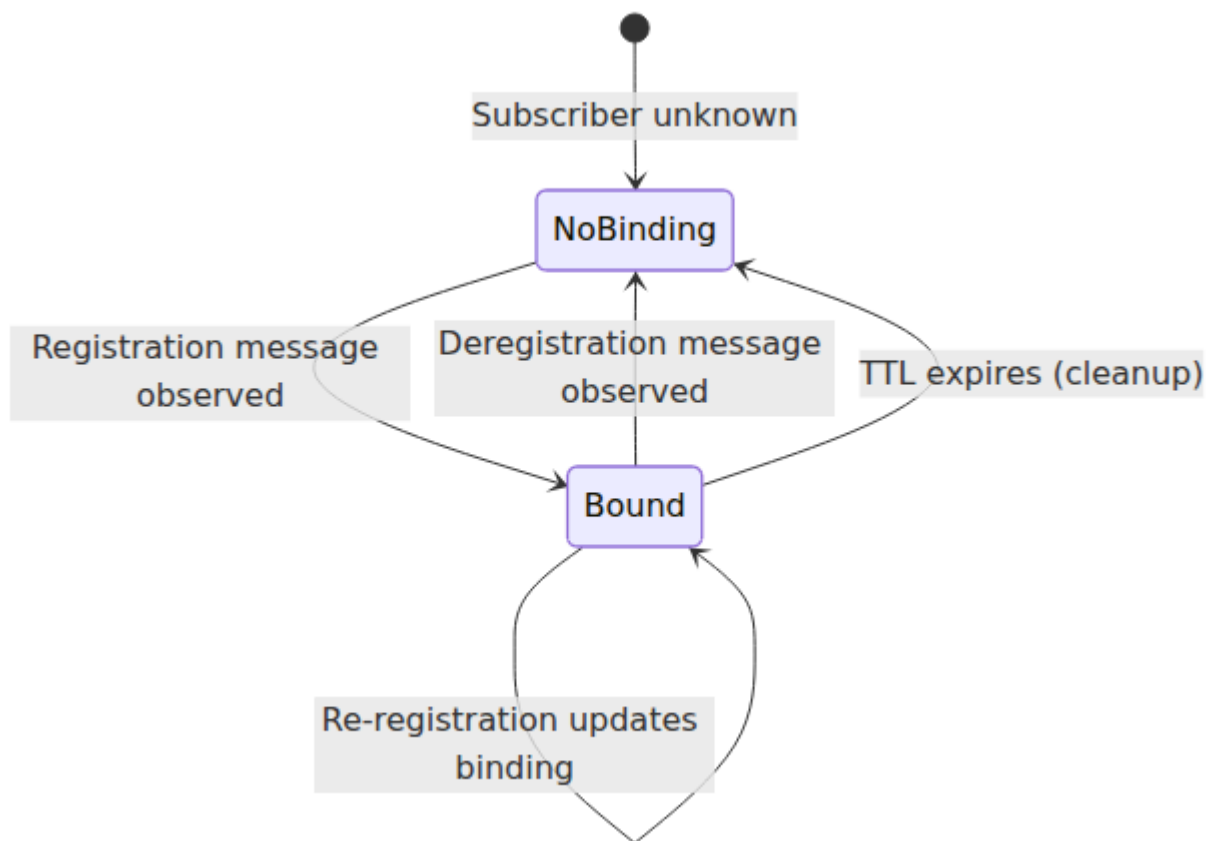
TTL expires (cleanup)

Bound

Re-registration updates
binding

□□□□□□□□

□□□□ Diameter □□□□□□□□□□□□□□□□



消息	消息	消息	消息	IMSI 消息
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)

Table 1: Diameter Messages

Interface	Message Name	Message ID	Originator
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

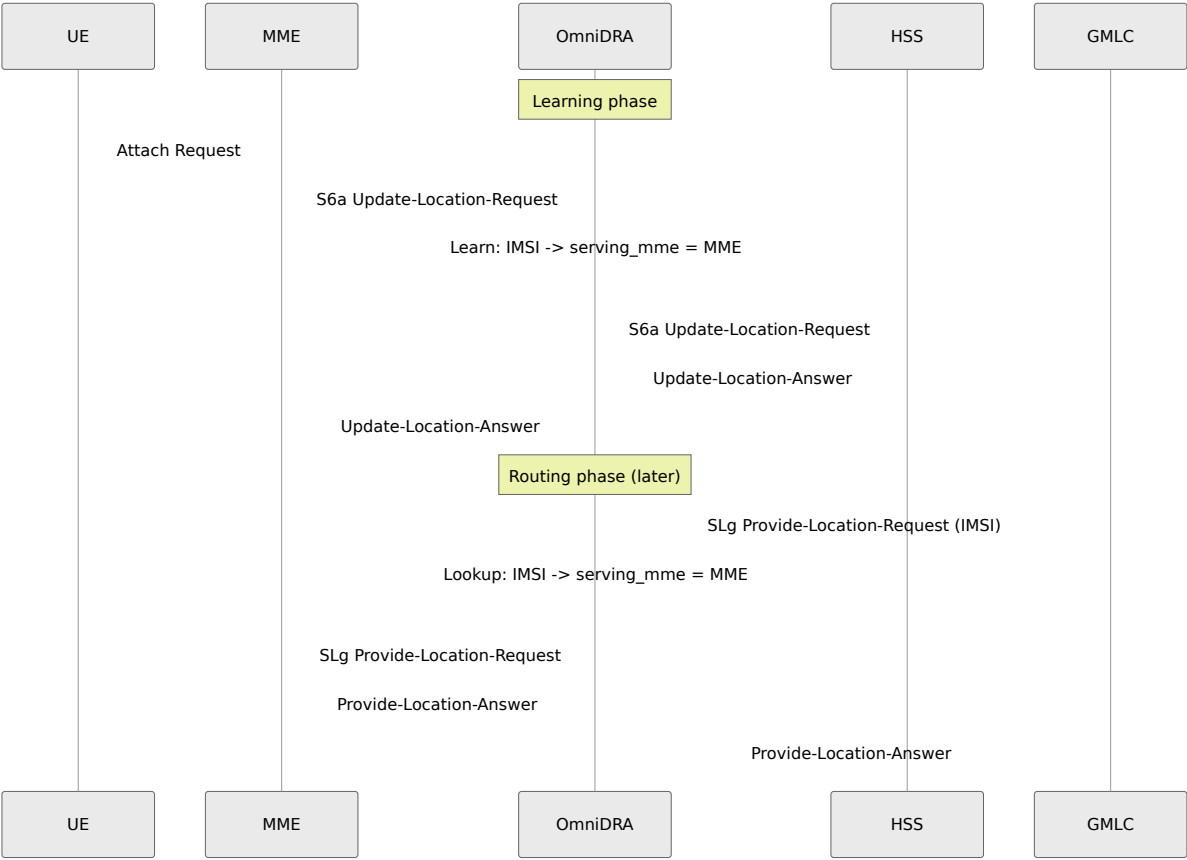
Table 2: Diameter Messages

Table 3: Diameter Messages

Interface	Message Name	Message ID	Originator
SLg	Provide-Location-Request (PLR)	8388620	serving_mme
SLh	LCS-Routing-Info-Request (RIR)	8388622	serving_mme

DRA configuration parameters: `diameter.request_timeout` (default: 30s)
Peer Connectivity Configuration: `Peer Connectivity Configuration`

□□□□□□□□□□



□□□□

□□□□□□□□□□□□□□□□□□ SLF □□□□□□□□□□□□□□□□□□ SLF □□□□□□□□

Incoming Request

Subscriber Lookup

Advanced Routing

Peer Selection / Relay

□□

□ config/runtime.exs □□ module_subscriber_lookup □□□□□□□

```
module_subscriber_lookup: %{\n  # \n\n  enabled: false,\n\n  # \n\n  binding_ttl_seconds: 86400\n}
```

Table

Field	Type	Value	Description
enabled	bool	false	Whether the subscriber lookup is enabled. Default is false.
children	list	[]	List of children DRA modules. Default is [].
binding_ttl_seconds	int	86400	Binding TTL in seconds. Default is 86400 (24 hours).

Example

Example

Example configuration for 24 hours binding TTL

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

이러한 구성을 통해 MME/PGW/CSCF는 24시간 DRA를 SLg와 SLh를 통해
MME를 관리할 수 있습니다.

이러한 구성 (LCS)은 GMLC를 통해 MME를 관리할 수 있습니다.

이러한

이러한 구성을 통해 MME를 관리할 수 있습니다.

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

이러한 구성 1은 MMEs를 관리할 수 있습니다. 30초
이러한

이러한 구성을 통해 MME를 관리할 수 있습니다.

이러한

이러한

이러한: `diameter.subscriber_lookup.binding.update`

이러한: `update`

이러한: DRA를 관리할 수 있습니다.

이러한:

- `imsi` - IMSI
- `binding_type` - `serving_mme`, `serving_pgw`, `serving_cscf`
- `serving_host` - Origin-Host

이러한

이러한: `diameter.subscriber_lookup.binding.delete`

이러한: `delete`

DR: DRA 00000000000000000000000000000000

DR:

- `imsi` - 0000 IMSI
- `binding_type` - 00000000

0000

DR: `diameter.subscriber_lookup.route.count`

DR: 000

DR: DRA 000000000000 SLg/SLh 000000000000

DR:

- `imsi` - 0000 IMSI
- `binding_type` - DRA 0000000000000000 `serving_mme`
- `serving_host` - DRA 0000000000 Origin-Host

0000

0000000000 **MME**

DR: DRA 000 SLg Provide-Location-Requests 0 SLh LCS-Routing-Info-Requests 00
0000 MME000000000000

00000:

- 00000000 DRA 000000000000
- 000000TTL 000
- 00 MME 00000000 DRA
- `enabled` 000 `false`

0000:

1. 000000000000
2. 000000 S6a ULR 0000000 DRA00000000000000000000
3. 00 `binding_ttl_seconds` 0000000000000000000000

4. 如何 MME 如何配置

如何配置

如何: 如何 S6a/Gx/Cx 如何 DRA如何配置

如何:

- 如何
- 如何如何如何如何如何
- 如何如何 AVP 如何如何 IMSI

如何:

1. 如何 `enabled: true`
2. 如何 DRA 如何如何如何如何
3. 如何如何如何 DRA 如何如何 `SubscriberLookup: Learned` 如何如何如何如何如何如何如何如何

如何如何如何如何如何如何如何如何

如何: DRA 如何如何如何如何如何如何如何如何 MME

如何:

- Cancel-Location-Request (CLR) 如何 Purge-UE-Request (PUR) 如何 DRA
- `binding_ttl_seconds` 如何如何如何如何如何如何如何如何

如何:

1. 如何 `binding_ttl_seconds` 如何如何如何如何如何如何如何如何如何如何
2. 如何 S6a 如何如何如何 DRA

□□

□□□□ ID

ID	□□	□□	□□
16777251	S6a/S6d	MME/SGSN □ HSS □□□□□□□□	3GPP TS 29.272
16777238	Gx	PCEF □ PCRF □□□□□□□□	3GPP TS 29.212
16777216	Cx	I-CSCF/S-CSCF □ HSS □ IMS □□	3GPP TS 29.229
16777255	SLg	GMLC □ MME □□□□□	3GPP TS 29.172
16777291	SLh	GMLC □ HSS/DRA □ LCS □□□□	3GPP TS 29.173

□□□

□□	□□	□□	□□
272	Credit-Control-Request/Answer (CCR/CCA)	Gx	□□□□□□□□□□
301	Server-Assignment-Request/Answer (SAR/SAA)	Cx	IMS □□□□□
316	Update-Location-Request/Answer (ULR/ULA)	S6a	□ MME □□□□□□ □□
317	Cancel-Location-Request/Answer (CLR/CLA)	S6a	HSS □□□□□□□
321	Purge-UE-Request/Answer (PUR/PUA)	S6a	MME □□□ UE □ □
8388620	Provide-Location-Request/Answer (PLR/PLA)	SLg	□□□ MME □□□□□ □□□
8388622	LCS-Routing-Info-Request/Answer (RIR/RIA)	SLh	LCS □□□□□□

网络

网络	接口	接口	接口
serving_mme	S6a ULR	SLg PLR, SLh RIR	网络 MME
serving_pgw	Gx CCR-I	-	网络 PGW网络
serving_cscf	Cx SAR (Registration)	-	网络 IMS 网络 S-CSCF网络



OmniDRA Metrics



□ □

目標

前提知識

ネットワークの基礎知識、OSの基礎知識、Diameterの基礎知識

学習目標

- ネットワークの基礎知識を復習する
- ネットワーク AVP と Diameter ネットワーク AVP の関係
- ネットワークの基礎知識/OSの基礎知識
- ネットワークの基礎知識 `match` ネットワークの基礎知識
- ネットワークの基礎知識

学習内容

1. ネットワークの基礎知識を復習する
2. ネットワーク AVP と Diameter ネットワーク AVP の関係 **Advanced Routing** ネットワーク AVP の関係
3. ネットワークの基礎知識 AVP ネットワークの基礎知識
4. ネットワークの基礎知識 `match` ネットワークの基礎知識
5. ネットワーク **Advanced Routing** ネットワークの基礎知識

学習方法

ネットワークDRA ネットワークの基礎知識ネットワークの基礎知識DRA ネットワークの基礎知識

学習成果

- ネットワークの基礎知識を復習する
- ネットワークの基礎知識を復習する
- ネットワーク **???** ネットワークの基礎知識
- ネットワークの基礎知識 **Standard Routing** ネットワークの基礎知識

学習評価

1. 配置 Advanced Routing
2. 配置 `diameter_peer_status` 为 1
3. 配置
4. 配置 Standard Routing

配置

配置 AVP 的 / 配置

配置

- 配置 AVP
- `edit` 配置 AVP `edit` 配置 AVP
- 配置
- 配置 `:request` `:answer`
- `edit` 配置

配置

1. 配置 AVP Advanced Routing 配置 AVP
2. 配置 `edit` AVP `add`
3. 配置
4. 配置 `:request` `:answer`
5. `edit` 配置

配置

配置

配置 `diameter_peer_status{origin_host="..."}` 为 0

配置

- 配置 `listen_port` 为 3868

- `ip` `port` `transport` `listen_port`
- `:diameter_tcp` `:diameter_sctp`
- `Origin-Host` `host`
- `initiate_connection`
- CER/CEA
-

1.

1. `listen_port` SCTP SCTP SCTP SCTP
2. `ip` `port` `transport`
- 3.
4. `Origin-Host` `host` Diameter
5. `initiate_connection` `initiate_connection: true` DRA HSS PCRF `false` MME SGSN P-GW
6. CER/CEA DRA
7. DRA

2. /

DRA

3.

-
- DRA
-

4.

1. SCTP
2. DRA

```

config :dra, :gateway,
  # 0 = 0 CER/CEA
  peer_settle_time_ms: 2_000,

  # RFC 3539
  # watchdog_timer_ms: 30_000

```

項目	単位	値	説明
peer_settle_time_ms	ms	0	DRA 0 CER/CEA DRA 0
watchdog_timer_ms	ms	30000	RFC 3539 Tw DWR/DWA

- diameter_peer_status
- peer_settle_time_ms 0

DRA RFC 3539 DRA

diameter_peer_unauthorized_connection_count_total
origin_host / peer_ip

- peers allow_undefined_peers_to_connect false

- 配置 Origin-Host 参数 host
- 配置 peers 参数

配置

1. 配置 peers 参数 host realm ip port transport
2. 配置 Origin-Host 参数 host 配置 DRA 参数
3. 配置 allow_undefined_peers_to_connect: false 参数
4. 配置 log_unauthorized_peer_connection_attempts: true 参数 DRA 参数
5. 配置 diameter_peer_unauthorized_connection_count_total 参数 peer_ip 参数 origin_host 参数

配置	配置	配置	配置
allow_undefined_peers_to_connect	配置	配置	配置 false
log_unauthorized_peer_connection_attempts	配置	配置	配置 true

SCTP 配置

このSCTP 設定は、ネットワークの信頼性を向上させるために TCP と比較して

使用します。

- ネットワークに SCTP を追加
- ネットワークに TCP と比較して SCTP を
- ネットワークの信頼性を向上させるために
- IP アドレスを `additional_ips` に追加
- ネットワークの信頼性を向上させるために DRA を追加
- この SCTP 設定は、ネットワークの信頼性を向上させるために

手順

1. この SCTP 設定は、Linux に `lksctp-tools` を追加
2. ネットワークの信頼性を向上させるために IP アドレスを SCTP と比較して TCP を
3. ネットワークの信頼性を向上させるために IP アドレスを DRA を追加
4. この SCTP 設定は、IP アドレスを `additional_ips` に追加して TCP と比較して DRA を `ip` に追加
5. ネットワークの信頼性を向上させるために
6. ネットワークの信頼性を向上させるために SCTP を追加して DRA を追加
7. この **SCTP Multihoming** 設定は、ネットワークの信頼性を向上させるために

エラー

エラー / 3002 エラー

ネットワークの信頼性を向上させるために DRA を追加して 3002 **DIAMETER_UNABLE_TO_DELIVER** エラーを発生させる

エラー

- ネットワークの信頼性を向上させるために Destination-Host と Destination-Realm を追加
- ネットワークの信頼性を向上させるために

- 配置 Diameter 代理
- 配置 Diameter 客户端
- 配置 Diameter 服务器 3003 DIAMETER_REALM_NOT_SERVED 3002

配置

1. 配置 Destination-Host 和 Destination-Realm 配置 `diameter_peer_status` 为 1
2. 配置 `peer_settle_time_ms` 配置 **Peer Flapping**
3. 配置 CER/CEA 配置 DRA 配置
4. 配置 DRA 配置 **Advanced Routing**
5. 配置 3002 和 3003 3002 配置 DRA 配置 3003 配置 3003 配置
6. 配置 3002 配置

```
rate(diameter_peer_message_result_code_count_total{result_code="3003"}[5m])
```

配置

配置 DRA 配置

`diameter_peer_unanswered_request_count_total` 配置
`diameter_peer_last_response_delay` 配置

配置

- 配置 `request_timeout` 配置 5000 配置
- 配置
- 配置 SCTP 配置
- 配置

配置

1.

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

2. `diameter_peer_last_response_delay`

3. DRA

4. SCTP Sctp Multihosting Sctp Multihoming

5. request_timeout TTL
request_timeout + 5

項目	単位	初期値	範囲	説明
request_timeout	秒	1	5000	DRA 読み込み/書き込みのタイムアウト時間

Gx / Gy UNKNOWN_SESSION_ID

00000000Gx 000000Gy 0000000000 5002DIAMETER_UNKNOWN_SESSION_ID00
00000000000000CCR-U000000CCR-T000000CCR-I000000

□ □ □ □ □ □

- DRA
-
- DRA
- Session-Id

□ □ □ □ □

1. Session-Id Gx Gy
 CCR-I Destination-Host Session-Id
 Standard Routing Advanced Routing

2. 設定 `peer_selection_algorithm` を `:random` に変更すると、ランダムにピアを選択します。
設定 `:failover` は、ピアが失敗した場合、Destination-Host のピアを選択します。
3. 設定 `5002` は、ピアが失敗した場合、ピアを選択します。
4. 設定 `peer_settle_time_ms` は、ピアが失敗した場合、DRA のピアを選択します。
Peer Flapping
5. 設定 `5002` は、ピアが失敗した場合、ピアを選択します。

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

Table 1: Diameter Error Codes

Error Code	Error Name	Description	Reference
2001	DIAMETER_SUCCESS	DRA successfully processed the request	RFC 6733 §7.1.1
3002	DIAMETER_UNABLE_TO_DELIVER	DRA unable to deliver the request to the destination	RFC 6733 §7.1.3
3003	DIAMETER_REALM_NOT_SERVED	DRA does not serve the realm specified in the request	RFC 6733 §7.1.3
3004	DIAMETER_TOO_BUSY	DRA is too busy to process the request	RFC 6733 §7.1.3
5002	DIAMETER_UNKNOWN_SESSION_ID	DRA does not know the Session-Id specified in the request	RFC 6733 §7.1.5
5012	DIAMETER_UNABLE_TO_COMPLY	DRA unable to comply with the request	RFC 6733 §7.1.5

Table 2: Diameter Error Codes

- Metrics - Prometheus metrics
- Standard Routing - RFC 6733
- Advanced Routing -
- SCTP Multihoming - SCTP

