Enrutamiento Avanzado

El Enrutamiento Avanzado permite que el DRA tome decisiones de enrutamiento basadas en el contenido de cada solicitud Diameter. No utiliza **Destination-Realm** por sí solo. Las **Reglas** coinciden con propiedades del mensaje, como el par de origen, Application-Id, Command-Code y valores de AVP. Cuando una regla coincide, dirige la solicitud a un conjunto elegido de pares o a un destino especial.

Una regla también puede distribuir el tráfico entre varios pares para **failover** y **balanceo de carga**. Utiliza la `prioridad` y `peso` por par. Para aplicaciones con estado como Gx (PCRF) y Gy (OCS), el Enrutamiento Avanzado preserva la **afinidad de sesión**. Los mensajes a mitad de sesión siempre regresan al par que posee la sesión.

El Enrutamiento Avanzado evalúa **solo paquetes de solicitud Diameter entrantes**. Los paquetes de respuesta siguen el enrutamiento de sesión establecido de regreso al par de origen. El DRA nunca los vuelve a evaluar contra estas reglas. Para reescribir el contenido del mensaje, consulte Transformación Avanzada. Para el enrutamiento basado en realm/host que se ejecuta cuando ninguna regla coincide, consulte Enrutamiento Estándar.

Cómo Funciona

El DRA evalúa cada solicitud entrante contra las reglas configuradas **en orden, de arriba hacia abajo**. La **primera** regla cuyos filtros coincidan gana y decide la ruta. El DRA no evalúa las reglas posteriores. Si ninguna regla coincide, la solicitud continúa con el enrutamiento estándar de **Destination-Host** / **Destination-Realm**.



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

Una regla coincidente solo puede resolver a pares que **no están actualmente conectados**. En ese caso, el DRA responde con **DIAMETER_UNABLE_TO_DELIVER (3002)**. El DRA nunca devuelve la solicitud al enrutamiento estándar. El enrutamiento estándar la entregaría al nodo incorrecto y devolvería un resultado engañoso.

Posición en la Canalización

El Enrutamiento Avanzado se ejecuta después de la Dirección de Roaming y la Búsqueda de Suscriptores. Su decisión anula el par que el enrutamiento estándar seleccionaría de otro modo.



Procesamiento de Reglas

El DRA evalúa las reglas de arriba hacia abajo. La **primera regla coincidente gana**. Cada regla combina sus filtros de acuerdo con su modo de `match`. Aplica su acción de `route` tan pronto como coincide. Si ninguna regla coincide, el DRA recurre al enrutamiento estándar.



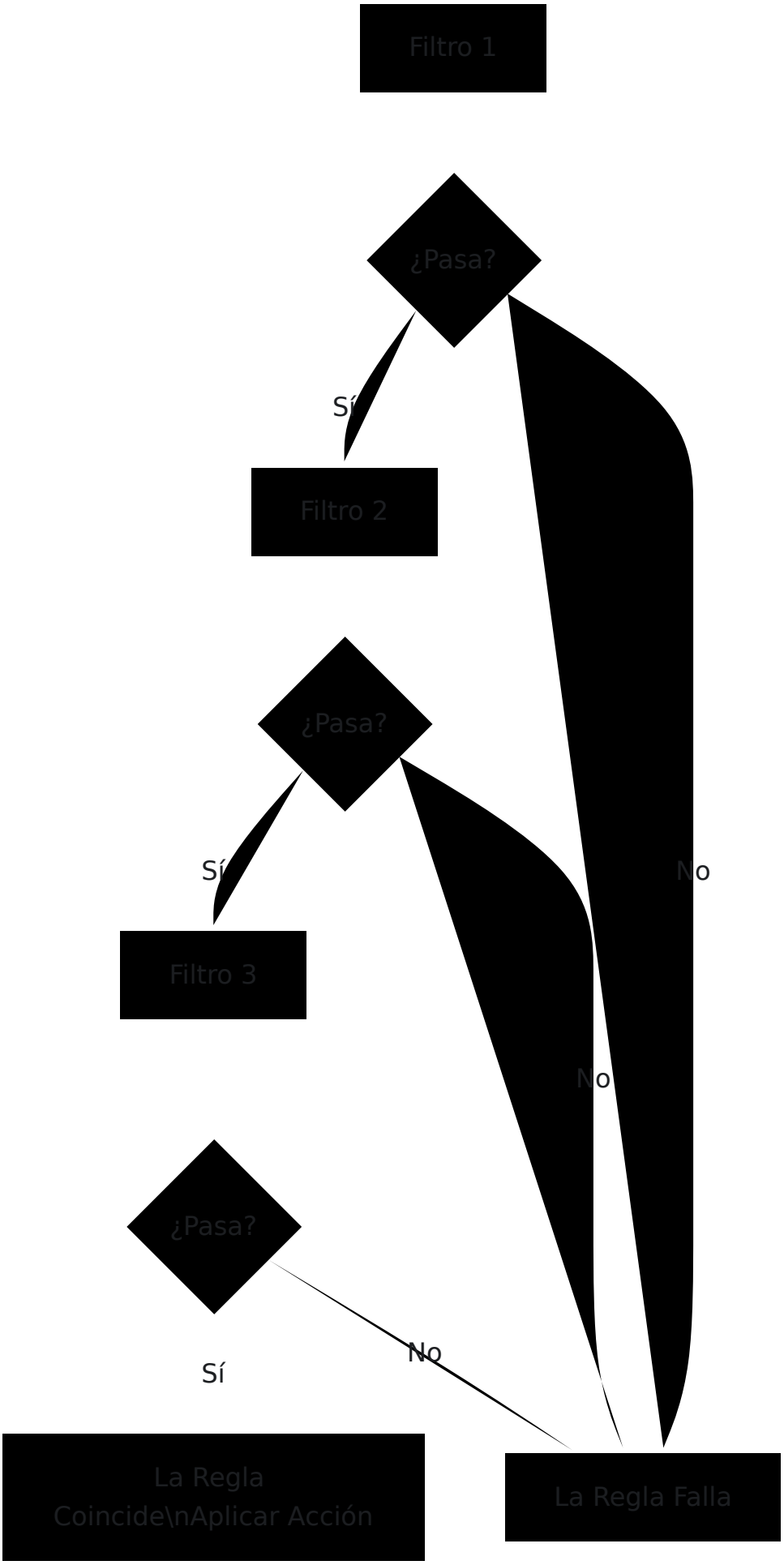
1. El DRA evalúa las reglas **en orden de arriba hacia abajo**, como se define en la configuración.
2. El DRA combina los filtros dentro de una regla de acuerdo con el modo de `match` de la regla (`:all`, `:any`, o `:none`).
3. La **primera regla coincidente gana**. El DRA no evalúa las reglas posteriores.
4. Si ninguna regla coincide, el DRA utiliza el enrutamiento estándar.

Modos de Coincidencia

El parámetro `match` determina cómo el DRA combina los filtros de una regla.

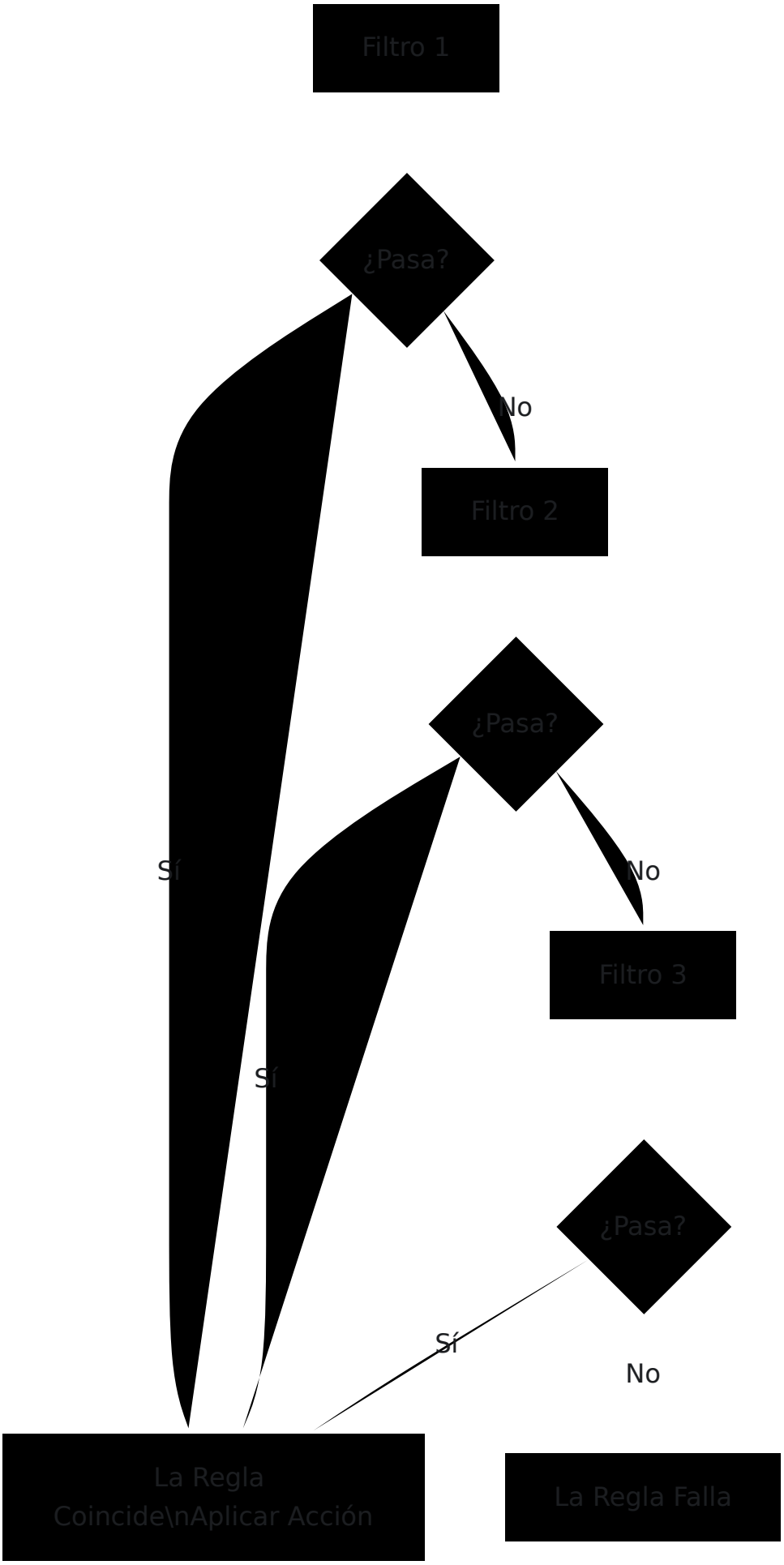
`:all` (lógica AND)

Cada filtro debe coincidir para que la regla tenga éxito. Con tres filtros, `filter1 AND filter2 AND filter3` deben ser todos verdaderos.


Filtro 1
¿Pasa?
Sí
Filtro 2
¿Pasa?
Sí
Filtro 3
¿Pasa?
Sí
No
No
No
La Regla Coincide\nAplicar Acción
La Regla Falla

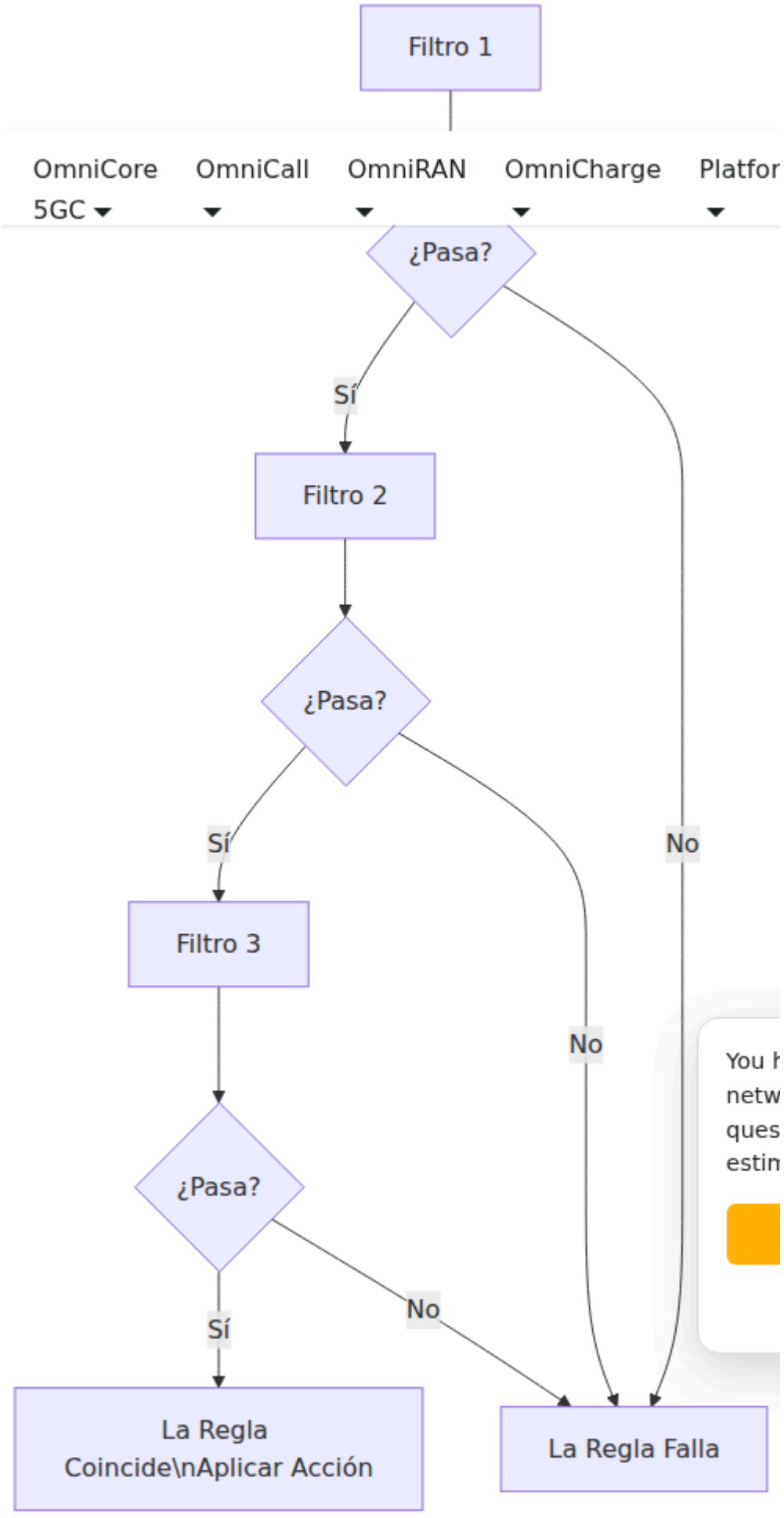
`:any` (lógica OR)

Al menos un filtro debe coincidir. Con tres filtros, `filter1 OR filter2 OR filter3` tiene éxito si alguno pasa.


Filtro 1
¿Pasa?
No
Filtro 2
¿Pasa?
Sí
No
Filtro 3
Sí
¿Pasa?
Sí
No
La Regla Coincide\nAplicar Acción
La Regla Falla

`:none` (lógica NOR)

Ningún filtro puede coincidir. Esta es una coincidencia inversa. Con tres filtros, `NOT filter1 AND NOT filter2 AND NOT filter3` deben ser todos verdaderos para que la regla tenga éxito. Cada filtro debe fallar.


Filtro 1
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platfor
¿Pasa?
Sí
Filtro 2
¿Pasa?
Sí
Filtro 3
¿Pasa?
Sí
La Regla Coincide\nAplicar Acción
No
No
No
La Regla Falla

Selección de Pares

Cuando una regla enruta a una lista de pares, el DRA selecciona solo entre los pares que están **actualmente conectados**. La forma en que escriba los pares determina cómo el DRA elige entre ellos.

Nombres de host simples

Una lista de nombres de host simples sigue el `peer_selection_algorithm` global (`:failover` o `:random`) configurado bajo `diameter`. Este es el comportamiento original y no cambia. Debe definir cada par en la configuración de pares Diameter del DRA. Nombre cada par por su **nombre de host exacto**, que es sensible a mayúsculas y minúsculas. Cada par debe estar actualmente conectado. El DRA omite los pares desconectados.

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

- `:failover` - siempre el primer par conectado en la lista. Esto da un primario determinista, luego un secundario.
- `:random` - un par conectado aleatorio. Esto distribuye la carga de manera uniforme.

Prioridad y peso

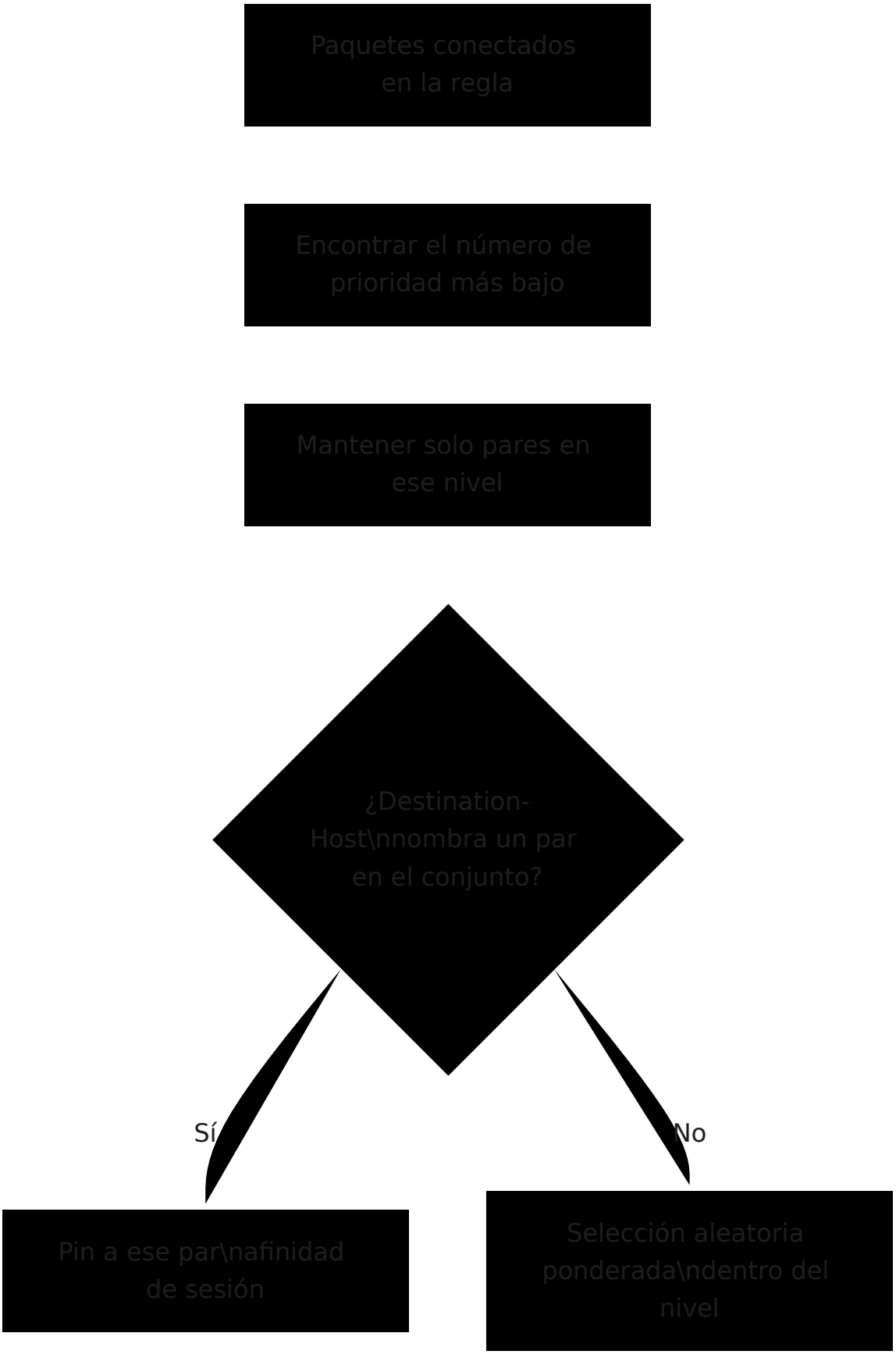
Para el failover y balanceo de carga por regla, escriba los pares como mapas con `priority` y `weight`. Esto es independiente del algoritmo global.

```
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}
  ]
}
```

La selección es **"la prioridad elige el nivel, el peso divide la carga"**:

1. El DRA considera solo pares conectados.
2. El **número de prioridad más bajo** presente gana el nivel. Se prefiere un número más bajo, como con los registros DNS SRV. Los niveles de números más altos actúan como copias de seguridad. El DRA utiliza un nivel de copia de seguridad solo después de que todos los pares conectados en un nivel inferior hayan desaparecido. Esto proporciona **failover**.
3. Dentro del nivel ganador, el DRA elige un par al azar, **ponderado por `weight`**. Los pesos iguales distribuyen la carga de manera uniforme. Los pesos desiguales sesgan la división. Esto proporciona **balanceo de carga**.

Si omite `priority` o `weight`, cada uno se establece por defecto en `100`.



Este modelo único expresa cada topología común:

Objetivo	Cómo expresarlo
Failover puro	Prioridades distintas en orden descendente (por ejemplo, 10, 20, 30); peso irrelevante
Balanceo de carga puro	Una prioridad para todos; pesos iguales
Balanceo de carga ponderado	Una prioridad para todos; pesos en la proporción deseada (por ejemplo, 70 / 30)
Balanceo de carga con copia de seguridad	Dos pares en prioridad 10, un tercero en prioridad 20

Afinidad de Sesión (Gx / Gy)

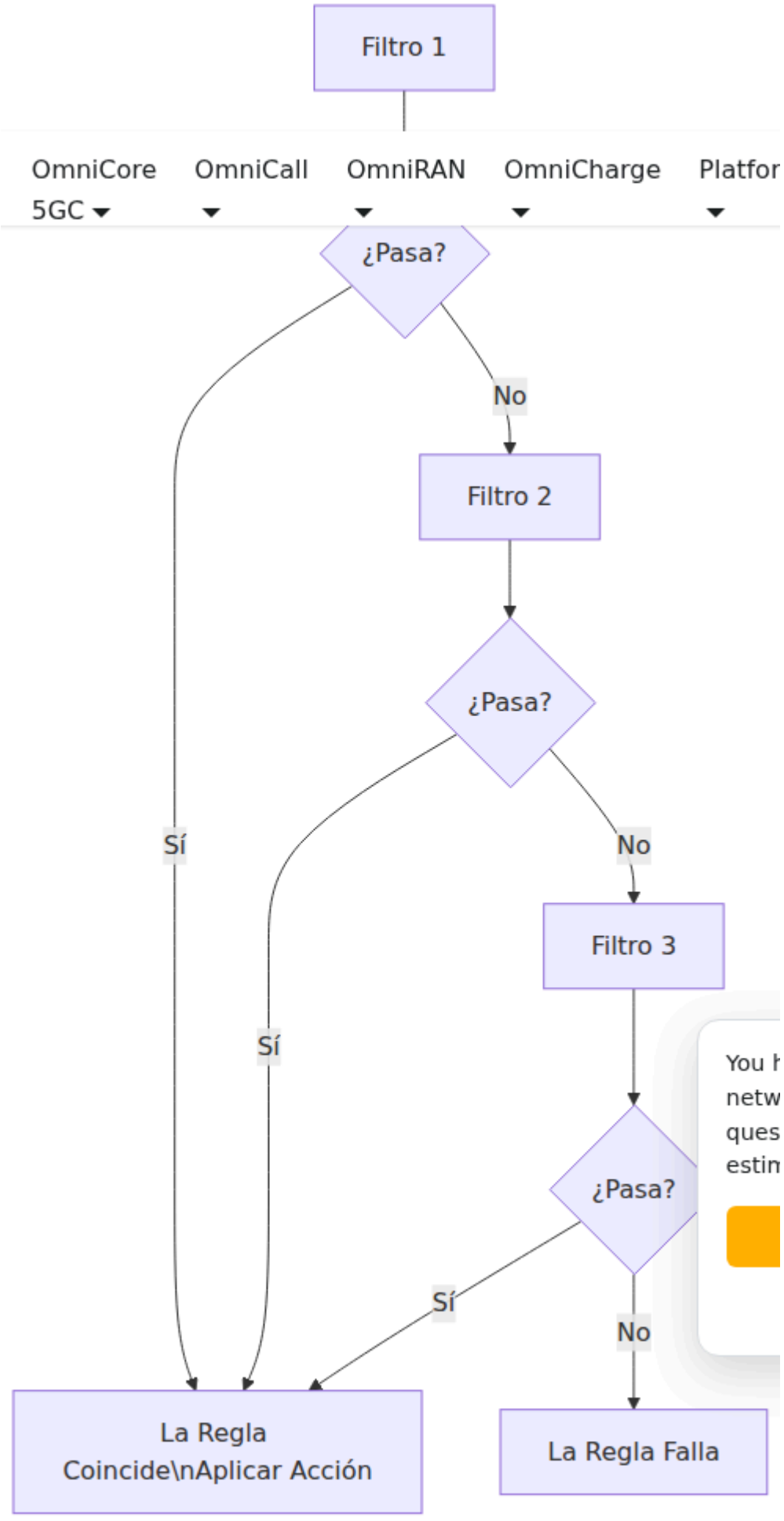
Las aplicaciones con estado llevan un **Session-Id** durante la vida de una sesión. Los mensajes a mitad de sesión deben llegar al **mismo** servidor que manejó la solicitud que inició la sesión. Si el DRA reequilibra un mensaje a mitad de sesión a un par diferente, ese par lo rechaza con **DIAMETER_UNKNOWN_SESSION_ID (5002)**.

El Enrutamiento Avanzado preserva la afinidad con el mecanismo estándar de Diameter, el **Destination-Host AVP (293)**. Cuando una regla utiliza pares de prioridad/peso:

- Si la solicitud lleva un Destination-Host que nombra uno de los pares conectados de la regla, el DRA **fija** la solicitud a ese par. Esto anula la prioridad y el peso.
- Si no hay tal Destination-Host, el DRA balancea la carga de la solicitud normalmente.

La solicitud que inicia la sesión (por ejemplo, Gx CCR-Initial) no lleva Destination-Host. El DRA la balancea a través del nivel. La respuesta devuelve el Origin-Host del servidor elegido. El cliente luego coloca ese valor en Destination-Host en todas las solicitudes posteriores (CCR-Update, CCR-

Terminate). Esto fija esas solicitudes a ese servidor automáticamente. En Gx/Gy, el **CC-Request-Type AVP (416)** distingue el tipo de solicitud.


Filtro 1
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platfor
¿Pasa?
No
Filtro 2
¿Pasa?
Sí
No
Filtro 3
Sí
¿Pasa?
Sí
No
La Regla Coincide\nAplicar Acción
La Regla Falla

Esta afinidad se aplica a listas de pares de prioridad/peso. Las listas de nombres de host simples siguen el algoritmo global. Este comportamiento no los afecta.

Configuración

Configure el Enrutamiento Avanzado bajo `module_advanced_routing` en `config/runtime.exs`.

```
module_advanced_routing: %{
  # Habilitar o deshabilitar el módulo
  enabled: false,

  # Las reglas se evalúan en orden; la primera coincidencia gana
  rules: [
    %{
      rule_name: "route_s6a_air_to_hss",
      match: :all,
      filters: [
        {:application_id, 16_777_251},
        {:command_code, 318}
      ],
      route: %{
        peers: ["hss01.example.com"]
      }
    }
  ]
}
```

Parámetros del Módulo

Parámetro	Tipo	Requerido	Por Defecto	Descripción
`enabled`	Booleano	No	`false`	Interruptor maestro para el módulo. Cuando `false`, el DRA no evalúa reglas y todo el tráfico utiliza enrutamiento estándar.
`rules`	Lista	Sí	`[]`	Lista ordenada de reglas de enrutamiento. El DRA las evalúa de arriba hacia abajo. La primera coincidencia gana. Consulte Parámetros de Regla.

Parámetros de Regla

Parámetro	Tipo	Requerido	Por Defecto	Descripción
`rule_name`	Cadena	Sí	-	Un nombre descriptivo y único para la regla. El DRA lo muestra en los registros y en la interfaz de gestión.
`match`	Átomo	Sí	-	Cómo se combinan los filtros: `:all` (AND lógico), `:any` (OR lógico), o `:none` (NOR lógico). Consulte Modos de Coincidencia.
`filters`	Lista	Sí	-	Las condiciones que un mensaje debe cumplir. Consulte Filtros.
`route`	Mapa o Átomo	Sí	-	A dónde envía el DRA el tráfico coincidente. Esta es una lista de pares o un destino especial. Consulte Destinos de Ruta.

Filtros

Un filtro es una tupla de `{filter_type, value}`. El Enrutamiento Avanzado admite estos filtros:

Filtro	Ejemplo	Descripción
`:via_peer`	`{:via_peer, "mme01.example.com"}`	El Origin-Host del par que envió la solicitud (el salto más reciente).
`:application_id`	`{:application_id, 16_777_251}`	El Diameter Application-Id. Consulte Identificadores de Aplicación.
`:command_code`	`{:command_code, 318}`	El Diameter Command-Code.
`:avp`	`{:avp, {1, "999999000000001"}}`	Un valor de AVP, escrito como `{avp_code, value}`.

Valores de Filtro

- **Valores de lista (OR).** Cualquier valor de filtro puede ser una lista. El DRA trata la lista como un OR lógico sobre sus miembros. Por ejemplo, `{:command_code, [317, 318]}` coincide con el código de comando 317 o 318. `{:via_peer, ["dra01.example.com", "dra02.example.com"]}` coincide con cualquiera de los pares.
- **Valores Regex.** Los valores de filtro de AVP y los miembros de la lista admiten expresiones regulares a través de un sigilo regex, por ejemplo `{:avp, {1, ~r"999001.*"}}`. Puede mezclar miembros, por ejemplo `{:avp, {1, ["505571234567", ~r"999001.*", ~r"505057.*"]}}`.
- **Verificación de presencia (`:any`).** Un valor de `:any` coincide cuando el atributo está presente con cualquier valor. Por ejemplo, `{:avp, {264, :any}}` coincide siempre que exista el AVP Origin-Host. `{:via_peer, :any}` coincide con cualquier par de origen.

Limitaciones de Filtros de AVP

Utilice filtros de AVP para AVPs **simples** únicamente, como User-Name, Origin-Host y Destination-Realm. **Los AVPs agrupados no son compatibles** y no coinciden. Los valores binarios complejos no son compatibles. Siempre use la forma `{:avp, {code, value}}`.

Destinos de Ruta

El parámetro `route` toma un mapa de lista de pares o uno de los átomos/tuplas de destino especiales a continuación.

Destino	Ejemplo	Descripción
Lista de pares	`%{peers: [...]}`	Enrutar a uno de los pares listados. Las entradas pueden ser nombres de host simples o mapas de prioridad/peso. Consulte Selección de Pares.
Destination-Host	`:destination_host`	Enrutar al par nombrado en el **Destination-Host AVP (293)** del mensaje. Si el AVP está ausente, la regla no se aplica y el DRA utiliza el enrutamiento estándar.
Drop	`:drop`	Descartar el mensaje sin respuesta al par solicitante. Use esto para bloquear tráfico no deseado, como suscriptores de prueba o rangos no deseados.
Error	`{:error, 3004}`	Responder con una respuesta de error Diameter que lleva el Result-Code dado. El DRA completa automáticamente Origin-Host, Origin-Realm y Session-Id. El DRA no reenvía el mensaje a ningún par.

Para una aplicación de **reenvío**, solo los códigos de resultado 3xxx (protocolo) son estrictamente válidos en las respuestas generadas (`3002`, `3003`, `3004`, `3007`). Algunas implementaciones también utilizan `5012` (DIAMETER_UNABLE_TO_COMPLY) para rechazar comandos específicos. Consulte las tablas de referencia para los significados.

Parámetros de Pares (forma de prioridad/peso)

Parámetro	Tipo	Requerido	Por Defecto	Descripción
`host`	Cadena	Sí	-	El Origin-Host del par. Debe coincidir exactamente con la identidad Diameter configurada del par y es sensible a mayúsculas y minúsculas.
`priority`	Entero	No	`100`	El nivel. El DRA prefiere números más bajos. Utiliza niveles de números más altos solo cuando todos los pares conectados en un nivel inferior no están disponibles.
`weight`	Entero	No	`100`	La parte del tráfico dentro de un nivel. El DRA divide el tráfico entre los pares de un nivel en proporción a sus pesos. Un peso de `0` excluye un par, a menos que todos los pares en el nivel sean `0`.

Ejemplos de Configuración

Failover a un Par de Respaldo

Enrutar Solicitudes de Información de Autenticación S6a a un HSS primario. Recaer a un HSS de recuperación ante desastres solo cuando el primario está fuera de servicio.

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

Cómo funciona: Mientras `hss01` esté conectado, cada solicitud va a él. Si `hss01` se desconecta, las solicitudes fallan sobre `hss-dr`. Si ambos están fuera de servicio, el DRA responde a las solicitudes con 3002.

Balanceo de Carga Equitativo

Distribuir el tráfico de Gx de manera uniforme entre dos 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}
    ]
  }
}
```

Cómo funciona: El DRA divide nuevas sesiones (CCR-Initial) aproximadamente 50/50 entre los dos PCRFs. El DRA fija los mensajes a mitad de sesión al PCRF que manejó la sesión. Utiliza el Destination-Host que llevan los mensajes.

Balanceo de Carga Ponderado con un Respaldo

Enviar más tráfico a un PCRF más grande y menos a uno más pequeño. Mantener un respaldo para cuando ambos no estén disponibles.

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

Cómo funciona: Mientras cualquiera de los PCRFs de prioridad 10 esté conectado, el DRA divide nuevas sesiones 70/30 entre ellos. El tráfico se mueve a `pcrf-dr` solo cuando **ambos** PCRFs de prioridad 10 están fuera de servicio. Las sesiones establecidas permanecen fijas a su servidor durante toda su vida útil.

Enrutar por Par de Origen

Enrutar tráfico S6a que llega de DRAs ascendentes específicos a un par de HSS local. Esto es útil durante una migración o cambio.

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

Cómo funciona: El DRA enruta a los nodos HSS locales solo las solicitudes S6a que llegaron de `dra01` o `dra02` y llevan el Origin-Realm coincidente (AVP 296). Todo lo demás pasa a reglas posteriores o al enrutamiento estándar.

Roaming por Patrón de IMSI

Enrutar tráfico de roaming entrante basado en patrones 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"]
  }
}
```

Cómo funciona: La regla coincide con mensajes S6a del Origin-Realm dado cuyo User-Name es exactamente `505571234567` o comienza con `999001`. El DRA enruta estos mensajes a los pares designados. El valor de la lista aplica lógica OR a través de la cadena exacta y la regex.

Honrar el AVP Destination-Host

Enrutar al par que el remitente nombró, solo para tráfico coincidente.

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

Cómo funciona: Cuando el IMSI coincide, el DRA envía la solicitud al par nombrado en su AVP Destination-Host (293). Si falta el AVP Destination-Host, el DRA trata la coincidencia como un fallo y aplica el enrutamiento estándar.

Descartar Tráfico No Deseado

Descartar silenciosamente tráfico de rango de prueba.

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

Cómo funciona: El DRA descarta mensajes S6a cuyo IMSI comienza con `999999`. No envía respuesta. Use esto para filtrar tráfico de prueba o para bloquear rangos de suscriptores específicos.

Protección contra Sobrecarga con una Respuesta de Error

Devolver DIAMETER_TOO_BUSY a un par sobrecargado específico.

```
%{
  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}
}
```

Cómo funciona: El DRA responde al tráfico S6a del par nombrado con **DIAMETER_TOO_BUSY (3004)**. Esto señala al par que se detenga. El DRA no reenvía nada hacia arriba.

Bloquear un Comando Específico

Rechazar un tipo de comando particular con un error explicativo en lugar de un descarte silencioso.

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

Cómo funciona: El DRA responde a las Solicitudes de Purga de UE S6a (comando 321) con **DIAMETER_UNABLE_TO_COMPLY (5012)**. Esto desactiva esa operación y deja otros comandos intactos.

Leyendo la Configuración Activa

Puede leer las reglas activas a través de la API de gestión en `GET /api/rules`. La respuesta incluye las reglas de enrutamiento, las reglas de transformación y el `routing_algorithm` global.

Para cada regla que enruta a una lista de pares, la API informa cómo se comporta la selección:

- `route.selection` es `"priority_weight"` cuando la regla utiliza mapas de prioridad/peso. De lo contrario, es el algoritmo global (`"failover"` o `"random"`).
- En modo `priority_weight`, la API devuelve cada par con su `priority` y `weight` efectivos. Completa los valores por defecto para que la distribución sea 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": []
  }
}
```


Mejores Prácticas

1. **Especificidad primero.** Ordene las reglas de la más específica a la más general. La primera coincidencia gana.
2. **Coincidencias comunes temprano.** Coloque las coincidencias de mayor volumen cerca de la parte superior. Esto reduce la sobrecarga de evaluación.
3. **Validar regex.** Pruebe patrones de expresiones regulares contra mensajes capturados antes de la implementación.
4. **Nombrar claramente.** Use valores descriptivos de `rule_name`. Los registros y la interfaz de gestión se explican por sí mismos.
5. **Preferir prioridad/peso para aplicaciones con estado.** Use mapas de pares de prioridad/peso para Gx/Gy. El DRA luego aplica automáticamente

la afinidad de sesión.

Solución de Problemas

Sesiones Rechazadas con UNKNOWN_SESSION_ID

Síntomas: Las sesiones Gx o Gy se establecen correctamente, pero el servidor rechaza las actualizaciones a mitad de sesión con el Result-Code 5002.

Causas posibles:

- Los clientes (PCEF/OCF) no establecen Destination-Host en las solicitudes a mitad de sesión, por lo que el DRA no puede fijarlas.
- Una regla enruta este tráfico con nombres de host simples y el algoritmo global `:random`. Este algoritmo no aplica afinidad de sesión.

Resolución:

1. Confirme que las solicitudes a mitad de sesión lleven un Destination-Host que nombre al servidor que respondió a la solicitud inicial.
2. Convierta la lista de pares de la regla a mapas de prioridad/peso. El DRA luego aplica automáticamente la afinidad de sesión.

Tráfico Coincidente Respondido con 3002

Síntomas: Una regla debería enrutar ciertas solicitudes, pero el DRA las responde con DIAMETER_UNABLE_TO_DELIVER.

Causas posibles:

- Ninguno de los pares listados en la regla coincidente está actualmente conectado.
- Los nombres de host en la regla no coinciden exactamente con los valores de Origin-Host de los pares.

Resolución:

1. Verifique el estado de conexión de los pares en la interfaz de gestión o en los registros.
2. Confirme que los valores de `host` en la regla coincidan exactamente con los Origin-Host de los pares. La coincidencia es sensible a mayúsculas y minúsculas.

Una Regla No Está Teniendo Efecto

Síntomas: El tráfico que debería coincidir con una regla pasa por el enrutamiento estándar en su lugar.

Causas posibles:

- `enabled` está configurado en `false`.
- Una regla anterior en la lista coincide primero y gana la ruta.
- Un valor de filtro no coincide. Esto puede ser un desajuste de mayúsculas y minúsculas en un valor de AVP, o un AVP agrupado no compatible.

Resolución:

1. Confirme que `enabled` esté en `true`. Reinicie el DRA después del cambio.
2. Revise el orden de las reglas. La primera regla coincidente gana.
3. Confirme los valores de filtro contra un mensaje capturado. Recuerde que el DRA no puede coincidir con AVPs agrupados.

Referencia

Códigos de Resultado Diameter

Código	Nombre	Descripción	Referencia
3002	DIAMETER_UNABLE_TO_DELIVER	No hay par conectado disponible para entregar la solicitud	RFC 6733 §7.1.3
3003	DIAMETER_REALM_NOT_SERVED	Ninguna ruta configurada sirve al realm de la solicitud	RFC 6733 §7.1.3
3004	DIAMETER_TOO_BUSY	El nodo está sobrecargado y no puede procesar la solicitud	RFC 6733 §7.1.3
5002	DIAMETER_UNKNOWN_SESSION_ID	El servidor no reconoce el Session-Id. Por ejemplo, un mensaje a mitad de sesión llegó al servidor incorrecto	RFC 6733 §7.1.5

Código	Nombre	Descripción	Referencia
5012	DIAMETER_UNABLE_TO_COMPLY	El DRA no puede honrar la solicitud. Use esto para rechazar comandos específicos	RFC 6733 §7.1.5

AVPs Relevantes

AVP	Código	Descripción	Referencia
Destination-Host	293	Nombra el par específico que debe recibir una solicitud. Esta es la base para la afinidad de sesión	RFC 6733 §6.5
Session-Id	263	Identifica una sesi��n durante su vida útil	RFC 6733 §8.8
CC-Request-Type	416	Distingue solicitudes INITIAL, UPDATE y TERMINATION en Gx/Gy	RFC 4006 §8.3

Identificadores de Aplicación

ID	Interfaz	Descripción	Referencia
16777238	Gx	Control de políticas entre PCEF y PCRF	3GPP TS 29.212
16777251	S6a/S6d	Autenticación MME/SGSN y gestión de suscripciones hacia HSS	3GPP TS 29.272
4	Gy	Carga en línea entre OCF y OCS	RFC 4006

Documentación Relacionada

- Enrutamiento Estándar - Enrutamiento Destination-Host / Destination-Realm utilizado cuando ninguna regla coincide.
- Transformación Avanzada - Reescritura de contenido para solicitudes y respuestas, compartiendo el mismo motor de filtros.
- Referencia de Configuración - Referencia completa de parámetros `config/runtime.exs`, incluyendo el `peer_selection_algorithm` global.

Transformación Avanzada

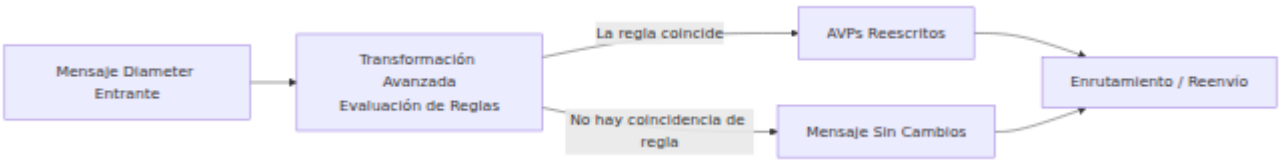
El módulo de **Transformación Avanzada** reescribe los Pares de Atributo-Valor (**AVPs**) dentro de los mensajes Diameter que coinciden con los criterios definidos por el operador. Úsalo para corregir, traducir o normalizar el contenido de los mensajes a medida que pasan a través de OmniDRA. No necesitas cambiar los elementos de red que envían o reciben.

Descripción General

La Transformación Avanzada evalúa una lista de **reglas** contra cada mensaje Diameter. Una regla combina un conjunto de **filtros** (los criterios de coincidencia) con una **transformación** (el cambio a aplicar). Cuando una regla coincide, el DRA aplica su transformación al mensaje y detiene el procesamiento de reglas.

La Transformación Avanzada proporciona tres acciones de transformación. `:edit` cambia el valor de un AVP existente. `:remove` elimina uno o más AVPs por código. `:overwrite` reescribe uno o más AVPs por nombre, incluidos los AVPs agrupados. Consulta Acciones de Transformación.

La Transformación Avanzada comparte su modelo de procesamiento de reglas con el motor de enrutamiento. Consulta Enrutamiento Avanzado para ver cómo el DRA toma decisiones de enrutamiento.

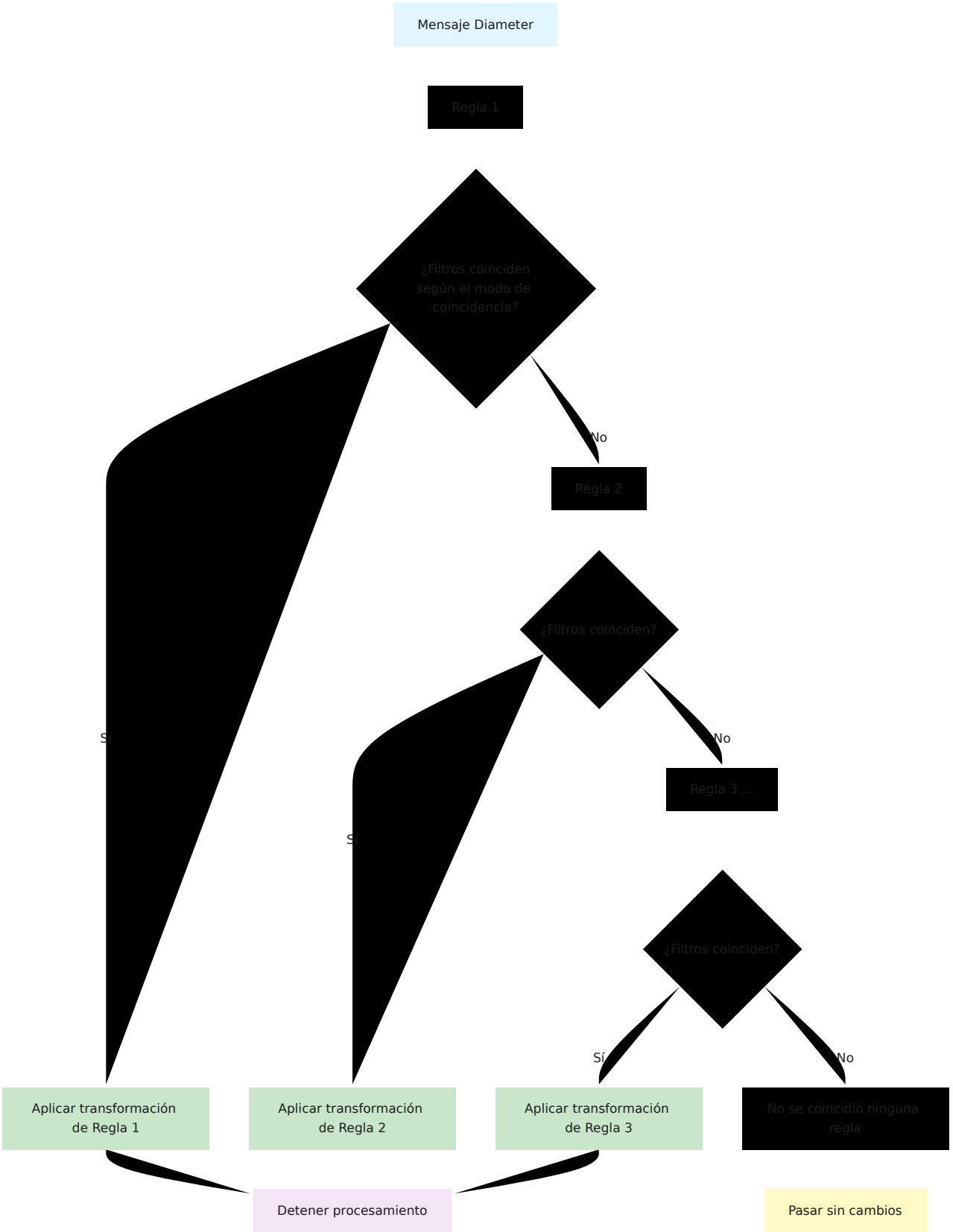


Cómo se Procesan las Reglas

El DRA evalúa las reglas **en orden, de arriba a abajo**, tal como aparecen en la configuración. La **primera regla que coincide gana**. Tan pronto como los filtros de una regla coinciden, el DRA aplica su transformación y no considera más reglas. Si ninguna regla coincide, el mensaje pasa **transparente** sin modificación.

Dentro de una sola regla, el parámetro `match` controla cómo se combinan los filtros individuales:

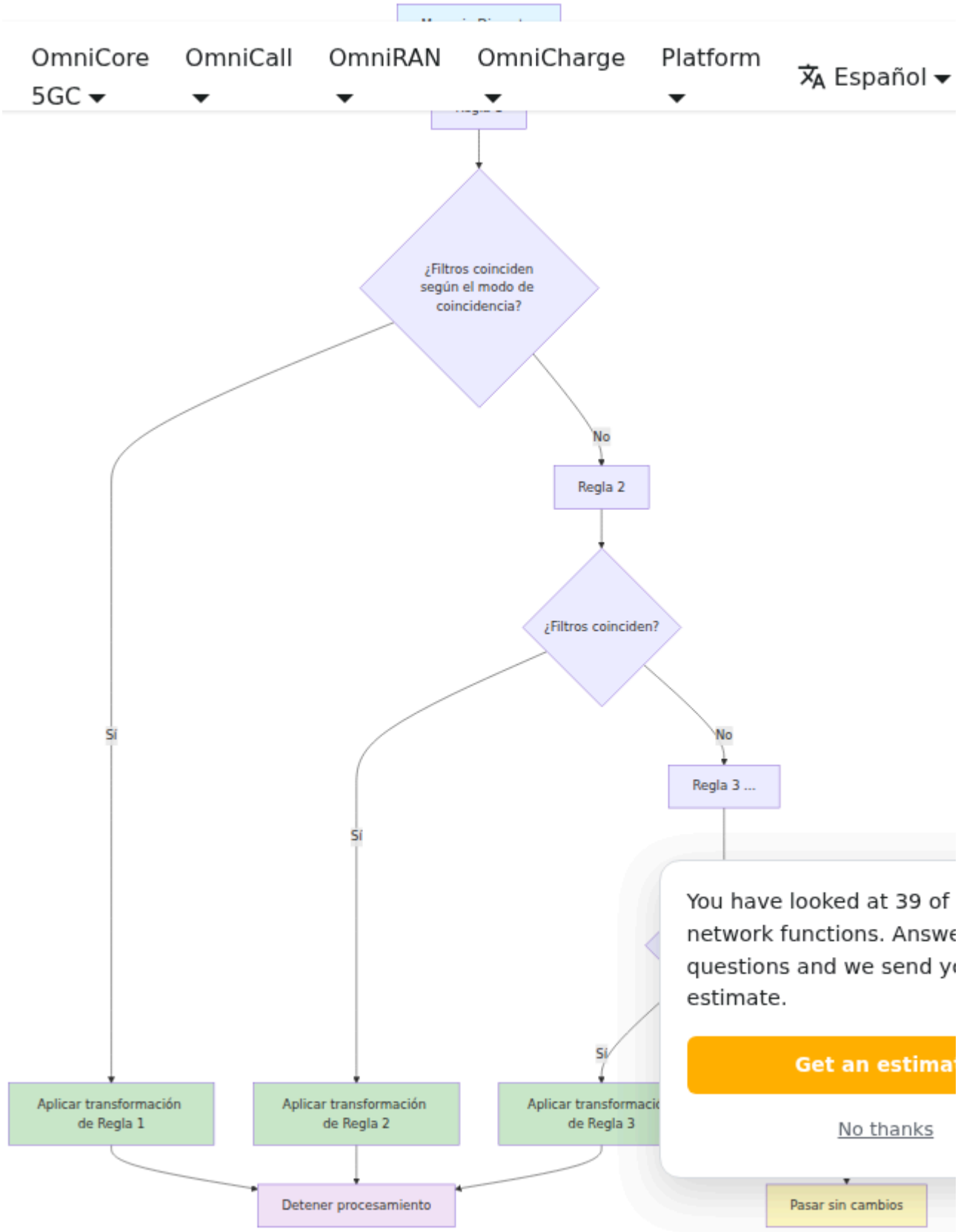
- `:all` - lógico **Y**. Cada filtro debe pasar para que la regla coincida.
- `:any` - lógico **O**. Al menos un filtro debe pasar para que la regla coincida.
- `:none` - lógico **NOR**. Ningún filtro puede pasar para que la regla coincida (coincidencia inversa).



Las reglas de transformación operan sobre el contenido del mensaje. No seleccionan un destino. Una regla puede reescribir un AVP, por ejemplo, un dominio o una identidad. El motor de enrutamiento luego utiliza ese AVP para reenviar el mensaje.

Evaluación del Modo de Coincidencia

Los tres modos de coincidencia evalúan la misma lista de filtros de manera diferente:



Filtros

Los filtros son las condiciones que deciden si una regla se aplica a un mensaje. La Transformación Avanzada proporciona seis tipos de filtros.

Filtro	Formato	Descripción
`:packet_type`	`{:packet_type, :request}`	Coincide con la **dirección** del mensaje. Acepta solo `:request` o `:answer`; no se acepta una lista de valores. Consulta Filtrado por Tipo de Paquete.
`:application_id`	`{:application_id, <id>}`	Coincide con el Application-Id de Diameter del mensaje (por ejemplo, `16777251` para S6a/S6d). Consulta IDs de Aplicación.
`:command_code`	`{:command_code, <code>}`	Coincide con el Código de Comando de Diameter del mensaje (por ejemplo, `318` para AIR). Consulta Códigos de Comando.
`:avp`	`{:avp, {<code>, <value>}}`	Coincide con un AVP por su código numérico y valor esperado. Consulta Filtrado de AVP.
`:to_peer`	`{:to_peer, "hss01.example.com"}`	Coincide con el **peer de destino resuelto** de una solicitud. Coincide solo con paquetes de **solicitud** y nunca coincide con respuestas. Acepta una lista de nombres de host (O). Consulta Filtrado de Peer.
`:from_peer`	`{:from_peer, "hss01.example.com"}`	Coincide con el **peer que respondió** a un mensaje.

Filtro	Formato	Descripción
		Coincide solo con paquetes de **respuesta** y nunca coincide con solicitudes. Acepta una lista de nombres de host (O). Consulta Filtrado de Peer.

Filtrado por Tipo de Paquete

El filtro `:packet_type` restringe una regla a una dirección de tráfico. Acepta exactamente uno de dos valores:

- `:request` - coincide solo con mensajes de **solicitud** Diameter.
- `:answer` - coincide solo con mensajes de **respuesta** Diameter.

A diferencia de `:application_id`, `:command_code` y `:avp`, el filtro `:packet_type` **no** acepta una lista de valores.

Filtrado de AVP

El filtro `:avp` tiene la estructura `{avp_id, avp_value}`. El `avp_id` es el código numérico del AVP. El `avp_value` es el valor con el que comparar, expresado como una cadena.

Utiliza el filtro `:avp` para **AVPs simples** como `User-Name` y `Origin-Host`. **Los AVPs agrupados no son compatibles** y no coinciden. Los valores binarios complejos no coinciden.

El filtro coincide con valores de varias maneras:

- **Cadena exacta** - `{:avp, {1, "999999000000001"}}` coincide solo con ese valor exacto.
- **Lista de valores** - `{:avp, {1, ["50557...", "505057..."]}}` coincide si el AVP es igual a **cualquiera** de los valores de la lista. Esta es lógica OR dentro de la lista.

- **Expresión regular** - `{:avp, {1, [~r"9999999.*"]}}` coincide con valores contra un patrón regex.

Utiliza la sintaxis `~r"pattern"` para coincidencias de expresiones regulares:

Patrón	Coincide
`~r"999001.*"`	Valores de `User-Name`/IMSI que comienzan con `999001`
`~r"^310[0-9]{3}.*"`	Valores con prefijos específicos de MCC/MNC
`~r".*test$"`	Valores que terminan con `test`

Filtrado de Peer

Los filtros `:to_peer` y `:from_peer` coinciden con un **nombre de host de peer**, no con el contenido del mensaje. Son específicos de dirección y complementarios:

- **`:to_peer`** coincide con el **peer de destino resuelto** al que el DRA está a punto de reenviar una solicitud. Coincide solo con paquetes de **solicitud**. Nunca coincide con una respuesta.
- **`:from_peer`** coincide con el **peer que respondió** a un mensaje. Coincide solo con paquetes de **respuesta**. Nunca coincide con una solicitud.

Ambos filtros aceptan un solo nombre de host o una **lista de nombres de host**. Con una lista, el filtro coincide si el peer es igual a **cualquiera** de los nombres de host en la lista. Esta es lógica OR.

Filtro	Ejemplo	Coincide
`:to_peer`	`{:to_peer, "hss01.example.com"}`	Solicitudes enrutadas al peer `hss01.example.com`
`:to_peer`	`{:to_peer, ["hss01.example.com", "hss02.example.com"]}`	Solicitudes enrutadas a **cualquiera** de los peers listados
`:from_peer`	`{:from_peer, "hss01.example.com"}`	Respuestas recibidas del peer `hss01.example.com`

Acciones de Transformación

El mapa `transform` describe la transformación. Su clave `action` selecciona **una de tres** transformaciones. La **dirección** del mensaje determina qué acciones están permitidas:

Acción	Qué hace	Solicitudes	Respuestas
`:edit`	Reescribe el **valor** de un AVP existente.	Sí	No
`:remove`	Elimina el/los AVP(s) nombrados del mensaje.	Sí	Sí
`:overwrite`	Reescribe AVP(s) por nombre, incluidos los AVPs **agrupados**, utilizando un diccionario de decodificación.	Sí	Sí

La Acción `:edit`

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

`:edit` se comporta de la siguiente manera:

- **Modifica el valor** del AVP nombrado en el mensaje Diameter al valor proporcionado.
- Afecta solo a los AVPs que **ya existen** en el mensaje. Si el AVP objetivo no está presente, el DRA **no realiza ningún cambio** y pasa el mensaje tal como fue recibido.
- Cada entrada en `avps` es `{:avp, {<code>, <new_value>}}`. Establece ese AVP al nuevo valor.
- `:edit` se aplica a **solicitudes solamente**. Para reescribir AVPs en respuestas, o para alcanzar AVPs agrupados, utiliza `:overwrite`.

La Acción `:remove`

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

`:remove` se comporta de la siguiente manera:

- **Elimina** el/los AVP(s) nombrados en `avps` del mensaje.
- Coincide solo por **código de AVP**. **Ignora** el valor en cada entrada de `avps` para la eliminación. Utiliza `:any` como valor para hacer esto explícito.
- Se aplica tanto a **solicitudes como a respuestas**.

La Acción `:overwrite`

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

`:overwrite` se comporta de la siguiente manera:

- **Reescribe** el/los AVP(s) nombrados por nombre. Utiliza el `dictionary` proporcionado para decodificar y re-encodificar el mensaje.
- Decodifica el mensaje completo, por lo que `:overwrite` puede alcanzar **AVPs agrupados / de registro** que `:edit` no puede.
- **Requiere** una clave `dictionary`. Esta clave nombra el diccionario Diameter que describe el mensaje (por ejemplo, `:diameter_gen_3gpp_s6a`).
- Se aplica tanto a **solicitudes como a respuestas**.

Parámetros del Mapa de Transformación

Parámetro	Tipo	Requerido	Predeterminado	Descripci
`action`	Átomo	Sí	-	La transformació aplicar: `:edit` (re un valor de AVP e - **solo solicitude** `:remove` (elimina por código - solici respuestas), o `:overwrite` (rees AVP(s) por nombr incluidos AVPs ag - solicitudes y respuestas).
`avps`	Lista	Sí	-	Lista de objetivos Para `:edit`/`:ove` cada entrada es `{<code>, <new_v` y establece ese A `<new_value>`. Par `:remove`, el DRA solo `<code>` e igr valor (utiliza `:any`
`dictionary`	Átomo	Solo para `:overwrite`	`:not_used`	El diccionario Dia que decodifica y encodifica el mer para que el DRA p sobrescribir AVPs nombrados (agru por ejemplo, `:diameter_gen_3` Las acciones `:ed`

Parámetro	Tipo	Requerido	Predeterminado	Descripci
				`:remove` ignoran clave.

Configuración

Configura la Transformación Avanzada bajo `module_advanced_transform` en `config/runtime.exs`. El módulo está deshabilitado por defecto. No hace nada hasta que configures `enabled` a `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"}}]
      }
    }
  ]
}
```

Parámetros del Módulo

Parámetro	Tipo	Requerido	Predeterminado	Descripción
`enabled`	Booleano	No	`false`	Habilita el módulo. Cuando es `false`, el DRA no evalúa reglas y todos los mensajes pasan sin cambios.
`rules`	Lista	Sí (cuando está habilitado)	-	Lista ordenada de reglas de transformación El DRA las evalúa de arriba a abajo. La primera regla que coincide gana. Consulta Parámetros de Regla.

Parámetros de Regla

Cada entrada en `rules` es un mapa con las siguientes claves.

Parámetro	Tipo	Requerido	Predeterminado	Descripción
`rule_name`	Cadena	Sí	-	Identificador único y descriptivo para la regla. Soporta claridad operativa y monitoreo.
`match`	Átomo	Sí	-	Cómo se combinan los filtros: `:all` (Y), `:any` (O) o `:none` (NOR). Consulta Cómo se Procesan las Reglas.
`filters`	Lista	Sí	-	Lista de condiciones de filtro que un mensaje debe satisfacer según el modo de `match`. Consulta Filtros.
`transform`	Mapa	Sí	-	El cambio a aplicar cuando la regla coincide. Consulta Parámetros del

Parámetro	Tipo	Requerido	Predeterminado	Descripción
				Mapa de Transformación.

Ejemplos

Ejemplo 1: Reescribir un User-Name en Solicitudes S6a AIR

```
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"}}]
      }
    }
  ]
}
```

Cómo funciona: Con `match: :all`, cada filtro debe pasar. La regla coincide solo con mensajes de Solicitud de Información de Autenticación (`318`) de S6a (`16777251`) cuyo `User-Name` (AVP `1`) sea igual a `999999000000001`. Al coincidir, la acción `:edit` reescribe `User-Name` a `999999000000002`. Si un mensaje no lleva un AVP `User-Name`, el DRA no realiza ningún cambio.

Caso de uso: Corregir o normalizar una identidad de suscriptor específica mientras transita por el DRA. Por ejemplo, mapear un IMSI legado a su valor

actual durante una migración.

Ejemplo 2: Reescritura de Dominio Basada en Rango 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"}}]
      }
    }
  ]
}
```

Cómo funciona: La regla coincide cuando el `Origin-Realm` (AVP `296`) es igual al dominio del socio de itinerancia **y** el `User-Name`/IMSI (AVP `1`) comienza con `50557`. Luego reescribe el `Destination-Realm` (AVP `283`). El motor de enrutamiento luego reenvía el mensaje hacia la red asociada correcta. La primera regla que coincide gana, así que coloca los rangos de IMSI más específicos por encima de los más generales.

Caso de uso: Dirigir el tráfico de suscriptores de itinerancia o MVNO a la red central hospedada correcta. El DRA selecciona la red según el rango de IMSI y el dominio de origen.

Ejemplo 3: Eliminar un AVP de Respuestas de un 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}}]
      }
    }
  ]
}
```

Cómo funciona: El filtro `:packet_type` limita la regla a **respuestas**. El filtro `:from_peer` limita a respuestas de `hss01.partner.example.com`. El filtro `:command_code` limita a AIA (`318`). Cuando las tres coinciden, la acción `:remove` elimina el `Origin-Host` (AVP `264`) independientemente de su valor. La eliminación coincide solo por código de AVP.

Caso de uso: Eliminar un AVP de las respuestas de una red asociada. Usa esto cuando el socio complete el AVP incorrectamente, o cuando el AVP no debe ir hacia abajo.

Ejemplo 4: Sobrescribir un AVP Agrupado en Solicitudes 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"}}]
      }
    }
  ]
}
```

Cómo funciona: La regla coincide con solicitudes S6a (`16777251`) AIR (`318`). La acción `:overwrite` decodifica el mensaje con el diccionario `:diameter_gen_3gpp_s6a`, reescribe el/los AVP(s) nombrados y re-encodifica el mensaje. Decodifica el mensaje completo, por lo que `:overwrite` puede alcanzar AVPs anidados dentro de estructuras agrupadas/registro que `:edit` no puede tocar.

Caso de uso: Corregir un AVP que vive dentro de un AVP agrupado, por ejemplo, un campo dentro de un grupo de datos de suscripción o específico del proveedor. Un simple `:edit` no coincidiría con tal AVP.

Tablas de Referencia

IDs de Aplicación

ID	Interfaz	Descripción	Referencia
16777251	S6a/S6d	Autenticación y suscripción MME/SGSN ↔ HSS	3GPP TS 29.272

Códigos de Comando

Código	Comando	Interfaz	Referencia
318	Solicitud/Respuesta de Información de Autenticación (AIR/AIA)	S6a/S6d	3GPP TS 29.272 §7.2.5
316	Solicitud/Respuesta de Actualización de Ubicación (ULR/ULA)	S6a/S6d	3GPP TS 29.272 §7.2.3

Códigos de AVP Comunes

Código	AVP	Descripción	Referencia
1	User-Name	Identidad del suscriptor (IMSI en S6a).	3GPP TS 29.272 / RFC 6733 §8.14
264	Origin-Host	Identidad Diameter del originador del mensaje.	RFC 6733 §6.3
283	Destination-Realm	El dominio al que está destinado el mensaje. El DRA lo utiliza para el enrutamiento.	RFC 6733 §6.6
296	Origin-Realm	Dominio del originador del mensaje.	RFC 6733 §6.4

Mejores Prácticas

1. **Especificidad** - Ordena las reglas de la más específica a la más general. La primera regla que coincide gana.
2. **Rendimiento** - Coloca las reglas más comúnmente coincidentes primero. Esto reduce la sobrecarga de evaluación.
3. **Pruebas** - Valida los patrones de expresiones regulares antes de la implementación.
4. **Nomenclatura** - Utiliza valores descriptivos de `rule_name` para claridad operativa y monitoreo.
5. **Monitoreo** - Realiza un seguimiento de las tasas de coincidencia de reglas. Esto confirma que las reglas se comportan como se espera.

Documentación Relacionada

- Enrutamiento Avanzado - Selección de destino utilizando el mismo modelo de procesamiento de reglas.

Visión General de la Arquitectura DRA

OmniDRA es un **Agente de Enrutamiento Diameter (DRA)**. Se sitúa en el centro de una red de señalización Diameter. Reenvía mensajes de solicitud y respuesta entre clientes como nodos MME/SGSN y servicios de backend como el HSS, PCRF y OCS. OmniDRA está construido en Elixir/Erlang sobre la biblioteca OTP `:diameter`. Opera como un **Relay Diameter** y reenvía mensajes entre pares. No termina las aplicaciones que los mensajes transportan.

Tabla de Contenidos

1. Qué es el DRA
2. Contexto de Red
3. Pipeline de Procesamiento de Solicitudes
4. Registro de Pares
5. Reenvío de Mensajes
6. Tablas de Referencia
7. Lecturas Adicionales

Qué es el DRA

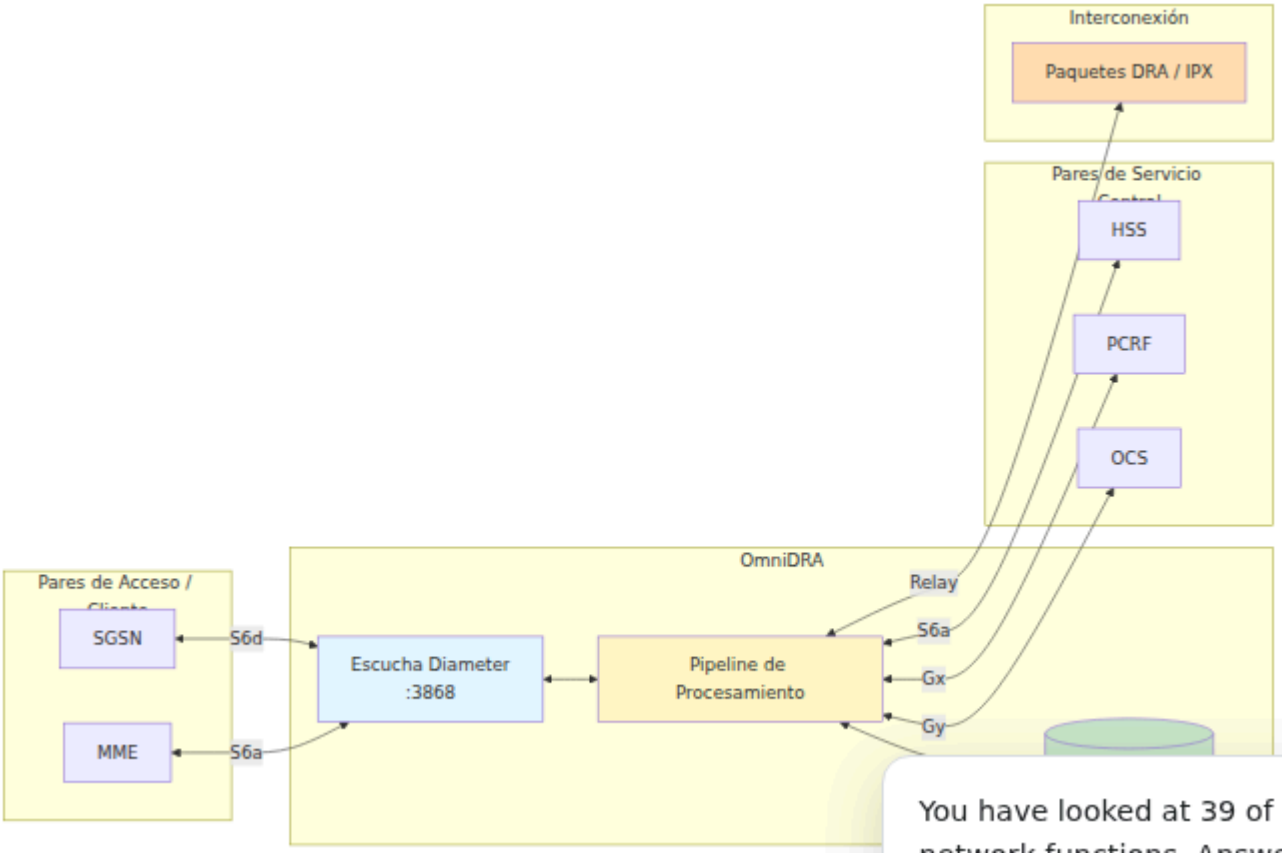
Un **Relay Diameter** recibe mensajes Diameter en un socket de escucha. Decide qué par conectado debe manejar cada mensaje. Luego reenvía el mensaje sin cambios, aparte de los AVPs de enrutamiento que gestiona el relay. OmniDRA funciona como un relay, por lo que no implementa la lógica de aplicación de las interfaces que transporta. Enruta **S6a**, **Gx**, **Gy**, y cualquier otra aplicación Diameter de manera transparente. Los pares deben anunciar soporte para la aplicación.

Características clave:

- **Enrutamiento basado en estándares** - El enrutamiento de solicitudes sigue el mecanismo de prioridad del Protocolo Base Diameter (RFC 6733). Selecciona un par con el **Destination-Host**, **Destination-Realm**, y **Application-Id**.
- **Correlación de respuestas con estado** - Las respuestas siempre regresan a lo largo del camino inverso de su solicitud. El DRA correlaciona cada respuesta por el **Hop-by-Hop Identifier**.
- **Módulos de procesamiento opcionales** - La dirección de roaming, la búsqueda de suscriptores, el enrutamiento avanzado y los módulos de transformación extienden el comportamiento estándar. No lo reemplazan.
- **Pipeline de seguridad** - La limitación de tasa, el firewall Diameter, la sanitización de AVP y el ocultamiento de topología protegen la red en el borde de señalización.

Contexto de Red

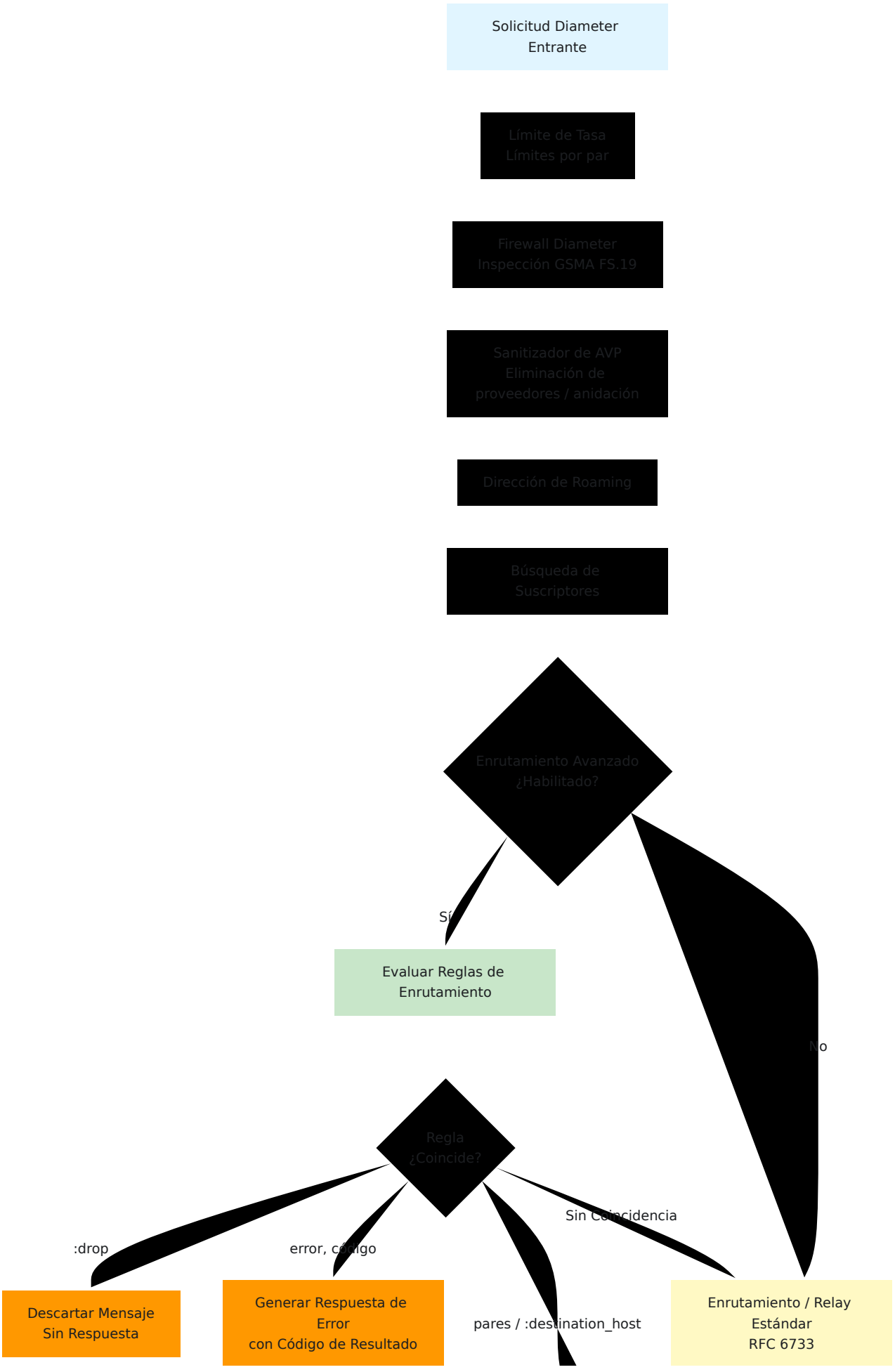
Despliegue el DRA entre los nodos de señalización del **lado de acceso** (como MME y SGSN) y los servicios del **lado central** (como HSS, PCRF y OCS). Cada par establece una conexión Diameter con el DRA. El DRA nunca origina tráfico de aplicación propio.



El DRA escucha conexiones Diameter entrantes en el puerto TCP/SCTP `3868`. Este es el puerto estándar definido por RFC 6733. Cada par anuncia los dominios y aplicaciones que soporta durante el **Intercambio de Capacidades** (CER/CEA). El DRA registra esto en su registro de pares para su uso en decisiones de enrutamiento.

Pipeline de Procesamiento de Solicitudes

Cada solicitud entrante pasa a través de un pipeline ordenado. Las etapas de seguridad y dirección se ejecutan primero. Las etapas de enrutamiento se ejecutan a continuación. Las respuestas omiten este pipeline. Ver Reenvío de Mensajes.


Solicitud Diameter Entrante
Límite de Tasa Límites por par
Firewall Diameter Inspección GSMA FS.19
Sanitizador de AVP Eliminación de proveedores / anidación
Dirección de Roaming
Búsqueda de Suscriptores
Enrutamiento Avanzado ¿Habilitado?
Sí
Evaluar Reglas de Enrutamiento
No
Regla ¿Coincide?
:drop
error, código
pares / :destination_host
Sin Coincidencia
Descartar Mensaje Sin Respuesta
Generar Respuesta de Error con Código de Resultado
Enrutamiento / Relay Estándar RFC 6733



Etapas del Pipeline

Etapa	Propósito
Limitador de Tasa	Hace cumplir límites de tasa de solicitud por par para proteger los servicios aguas abajo de sobrecargas.
Firewall Diameter	Inspecciona solicitudes contra las verificaciones de bajo nivel y Categoría 1/2/3 de GSMA FS.19. Ver Seguridad.
Sanitizador de AVP	Elimina AVPs de proveedores no permitidos y hace cumplir límites de profundidad de anidación de AVP.
Dirección de Roaming	Aplica dirección definida por el operador para influir en qué red se dirige a un suscriptor en roaming.
Búsqueda de Suscriptores	Resuelve el contexto de enrutamiento específico del suscriptor antes de la evaluación de reglas.
Enrutamiento Avanzado	Evalúa el enrutamiento basado en reglas que puede descartar, generar error o dirigir una solicitud a pares específicos. Ver Enrutamiento Avanzado.

Etapa	Propósito
Enrutamiento / Relay Estándar	Recurre al enrutamiento por prioridad RFC 6733 cuando no se aplica ninguna regla avanzada. Ver Enrutamiento Estándar.
Asociación de Sesión Rx	Redirige una solicitud Rx al PCRF que mantiene su sesión. Esta es la única etapa que se ejecuta *después* del Enrutamiento Avanzado. Una solicitud `Session-Termination-Request` de Rx no lleva nada más para enrutarse. Ver Asociación de Sesión Rx.
Ocultamiento de Topología	Reescribe o elimina AVPs identificativos (Route-Record, Origin-Host/Realm) en mensajes salientes. Ver Seguridad.

Cuando el Enrutamiento Avanzado produce un resultado `:drop`, el DRA descarta la solicitud sin respuesta. Un resultado de `error` genera una respuesta Diameter que lleva el **Código de Resultado** configurado. De lo contrario, el DRA reenvía solicitudes coincidentes a los pares especificados o al par **Destination-Host**. Las solicitudes no coincidentes caen en el enrutamiento estándar.

La Asociación de Sesión Rx es la única excepción a que el Enrutamiento Avanzado tenga la última palabra. Su alcance es estrecho. Solo redirige una solicitud que ya iba a ser reenviada, y solo para Rx. Deja un resultado `:drop`, un resultado de `error`, y un `Destination-Host` explícito sin tocar.

Prioridad de Enrutamiento Estándar

Cuando una solicitud llega al enrutamiento estándar, el DRA selecciona un par utilizando el mecanismo de prioridad de RFC 6733 §6.1:

1. **Destination-Host AVP (293)** - Si la solicitud contiene este AVP, el DRA enruta directamente al par nombrado. Este es el mecanismo de mayor prioridad. Si ese par no está conectado, el enrutamiento falla.

2. **Destination-Realm AVP (283)** - Si el Destination-Host está ausente, el DRA selecciona un par conectado que anuncia el dominio objetivo. El DRA balancea la carga entre los pares coincidentes.
3. **Application-Id** - El DRA filtra los pares candidatos a aquellos que anuncian soporte para la aplicación del mensaje. Los pares anuncian este soporte durante el Intercambio de Capacidades.

El comportamiento completo del enrutamiento estándar, incluido el algoritmo de selección de pares, está documentado en Enrutamiento Estándar.

Registro de Pares

El DRA mantiene un **registro de pares** en memoria. El registro rastrea cada conexión de par y las capacidades que el par anunció durante el Intercambio de Capacidades. El registro es la fuente autorizada para decisiones de enrutamiento. Solo los pares en estado **conectado** son elegibles para recibir tráfico. El DRA excluye automáticamente a los pares desconectados o inactivos.

Para cada par conectado, el registro registra:

- La **Identidad Diameter** del par (Origin-Host / FQDN).
- Los **dominios** que el par anuncia.
- Los **Application-Ids** que el par soporta.
- El **estado de conexión** actual.

El DRA aprende el soporte de dominio y aplicación dinámicamente de CER/CEA. Para agregar capacidad a una interfaz, conecte un par adicional que anuncie el dominio y la aplicación relevantes. El DRA comienza a balancear la carga automáticamente al nuevo par.

Reenvío de Mensajes

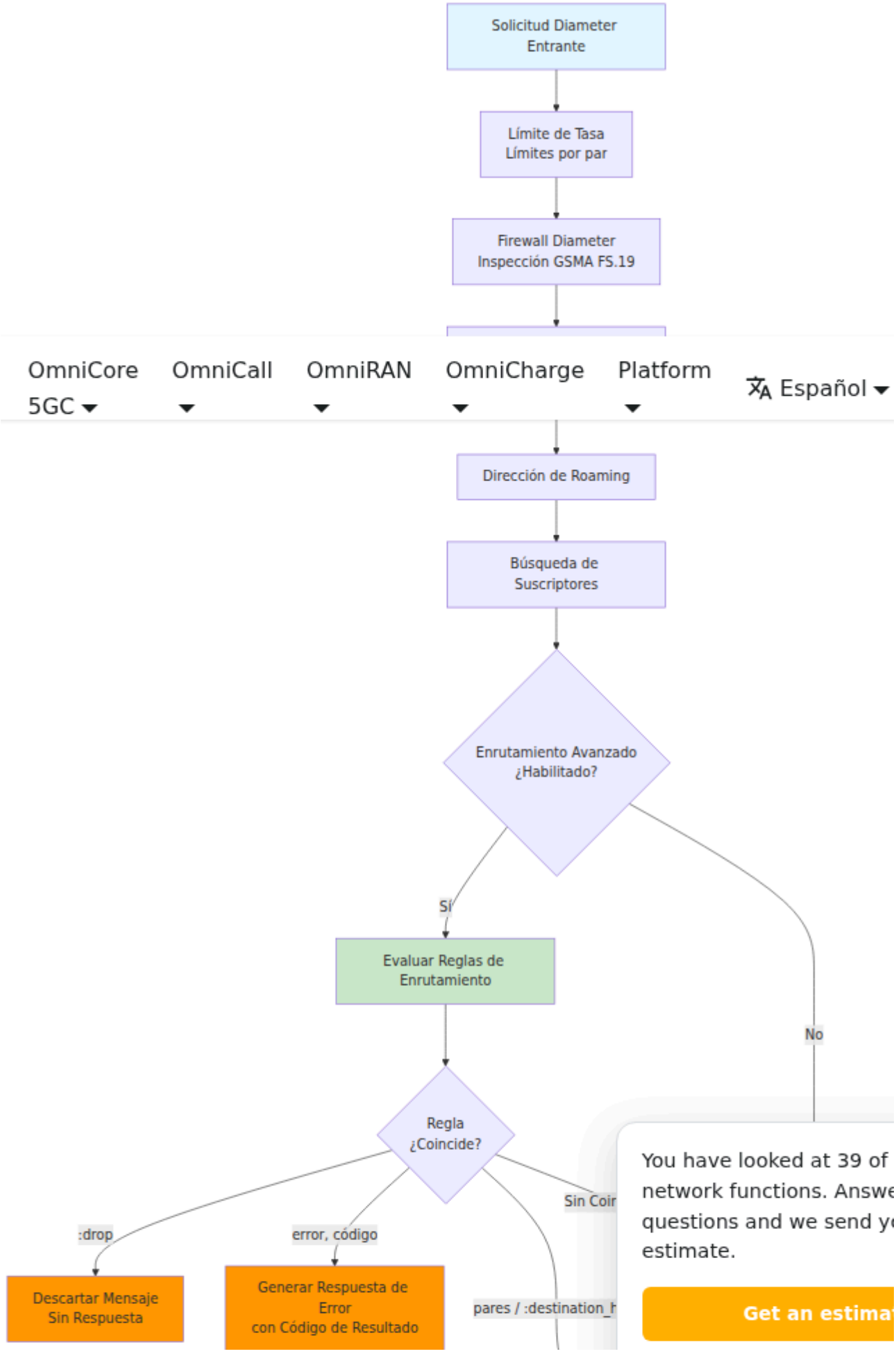
El DRA reenvía dos clases de mensajes fundamentalmente diferentes y los trata de manera diferente.

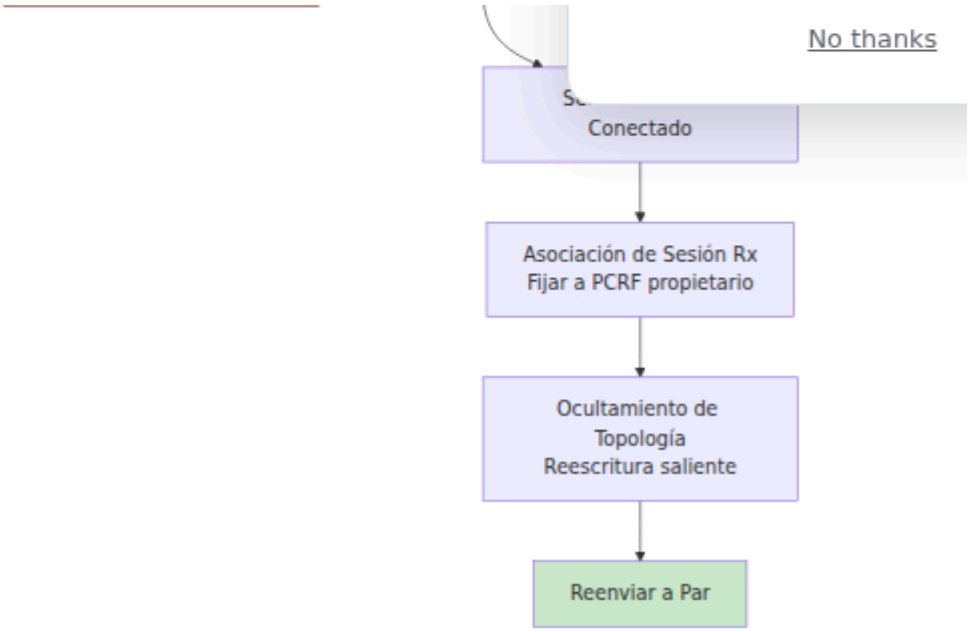
Solicitudes

Las solicitudes atraviesan todo el pipeline de procesamiento. El DRA selecciona un par de destino, actualiza los AVPs de enrutamiento que gestiona y reenvía el mensaje. El DRA es un relay, por lo que pasa la carga útil de la aplicación sin cambios.

Respuestas

Las respuestas nunca vuelven a entrar en el pipeline de enrutamiento. En su lugar, el DRA correlaciona cada respuesta con su solicitud original. Luego, el DRA devuelve la respuesta a lo largo del camino inverso.


Solicitud Diameter Entrante
Límite de Tasa Límites por par
Firewall Diameter Inspección GSMA FS.19
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
Dirección de Roaming
Búsqueda de Suscriptores
Enrutamiento Avanzado ¿Habilitado?
Sí
No
Evaluar Reglas de Enrutamiento
Regla ¿Coincide?
:drop
error, código
Sin Coir
Descartar Mensaje Sin Respuesta
Generar Respuesta de Error con Código de Resultado
pares / :destination_h
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima



Las reglas de reenvío para respuestas son:

- **Enrutamiento basado en sesión** - Las respuestas siempre siguen el camino inverso de su solicitud.
- **Preservación del ID de extremo a extremo** - El **Identificador de Extremo a Extremo** permanece sin cambios en cada salto.
- **Enrutamiento hop-by-hop** - El **Hop-by-Hop Identifier** cambia en cada salto. El DRA lo utiliza para mantener el estado de enrutamiento y para hacer coincidir una respuesta con el par que envió la solicitud.
- **Sin evaluación de reglas** - El DRA no evalúa reglas de enrutamiento ni inspecciona el contenido de AVP para respuestas.
- **Correlación con estado** - Las tablas de enrutamiento internas rastrean qué par envió cada solicitud. El DRA limpia este estado después de reenviar la respuesta.

El enrutamiento de respuestas es determinista. El protocolo Diameter requiere que una respuesta regrese al par original a lo largo del camino de solicitud establecido. Esto preserva la correcta gestión de sesiones y previene bucles de enrutamiento. Ver RFC 6733 §6.2 para detalles sobre el enrutamiento de mensajes de respuesta.

Tablas de Referencia

IDs de Aplicación 3GPP Comunes

ID	Interfaz	Descripción	Referencia
16777251	S6a/S6d	Autenticación y suscripción MME/SGSN ↔ HSS	3GPP TS 29.272
16777238	Gx	Control de políticas y carga PCEF ↔ PCRF	3GPP TS 29.212
4	Gy	Carga en línea (control de crédito) ↔ OCS	3GPP TS 32.299

AVPs de Enrutamiento Comunes

Código	Nombre	Descripción	Referencia
293	Destination-Host	Objetivo de enrutamiento a nivel de host explícito (máxima prioridad)	RFC 6733 §6.5
283	Destination-Realm	Dominio utilizado para seleccionar un par conectado	RFC 6733 §6.4
282	Route-Record	Registra la identidad de los relays que ha atravesado un mensaje	RFC 6733 §6.7.1

Lecturas Adicionales

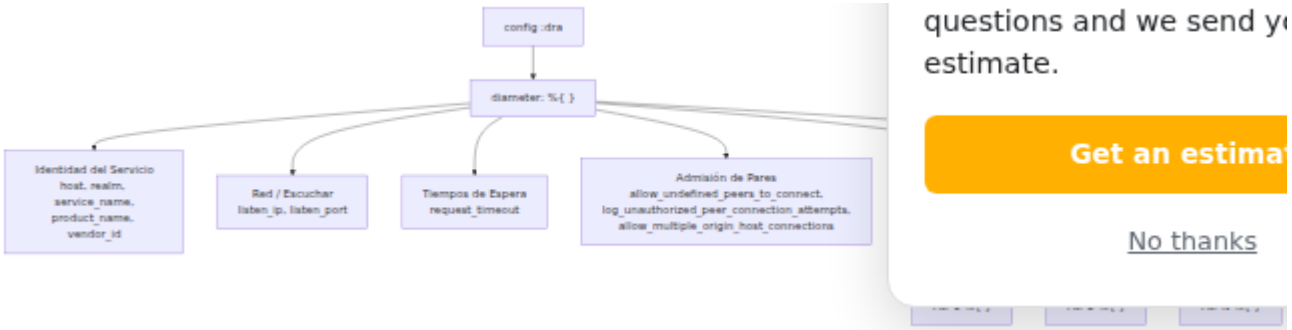
- **Enrutamiento Estándar** - Enrutamiento por prioridad RFC 6733, selección de pares y enrutamiento de respuestas en detalle.
- **Enrutamiento Avanzado** - Enrutamiento basado en reglas, resultados de descarte/error y orientación de pares.
- **Configuración** - Configuración base del DRA, transportes y ajustes de módulos.
- **Seguridad** - Limitación de tasa, firewall Diameter GSMA FS.19, sanitización de AVP y ocultamiento de topología.

Configuración Base de DRA

La **Configuración Base de DRA** define la identidad de OmniDRA, los oyentes de red, los tiempos de espera, la política de admisión de pares y la lista de pares Diameter a los que se conecta. Este bloque es la base para cada decisión de enrutamiento que toma el agente. Decláralo bajo `config :dra, diameter: %{...}` en `config/runtime.exs`.

Visión General

Todos los ajustes base residen en un único mapa `diameter`. El mapa contiene un conjunto de claves de nivel superior que describen el DRA en sí. También contiene una lista `peers`. Cada entrada en la lista describe un par Diameter.



Estructura de Configuración

```
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: [
      # Configuraciones de pares...
    ]
  }
```

Parámetros de Nivel Superior

Estas claves configuran la identidad del DRA, su oyente de red, los tiempos de espera y su política de admisión y selección de pares. Cada clave aparece una vez en el mapa `diameter`.

Parámetro	Tipo	Requerido	P
`service_name`	Átomo	Sí	-
`listen_ip`	Cadena o Lista	Sí	-
`listen_port`	Entero	Sí	-
`host`	Cadena	Sí	-
`realm`	Cadena	Sí	-
`product_name`	Cadena	Sí	-

Parámetro	Tipo	Requerido	P
`vendor_id`	Entero	Sí	-
`request_timeout`	Entero	No	5
`allow_undefined_peers_to_connect`	Booleano	Sí	f
`log_unauthorized_peer_connection_attempts`	Booleano	Sí	t
`allow_multiple_origin_host_connections`	Booleano	Sí	f

Parámetro	Tipo	Requerido	P
`peer_selection_algorithm`	Átomo	Sí	-
`peer_settle_time_ms`	Entero	No	`0`
`watchdog_timer_ms`	Entero	No	`3`
`peers`	Lista	Sí	-

Tamaño del Búfer de Socket SCTP

El tamaño del búfer de socket es una clave de nivel **superior** `:dra`. Configúralo con `config :dra, diameter_socket_buffer_size:`. **No** es un miembro del mapa `diameter:`.

```
config :dra, diameter_socket_buffer_size: 4_194_304

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

Parámetro	Tipo	Requerido	Predeterminado
`diameter_socket_buffer_size`	Entero	No	`4194304` (4 MB)

Selección de Pares

Cuando más de un par configurado coincide con los criterios de enrutamiento para una solicitud, `peer_selection_algorithm` determina qué par la recibe. Esta configuración afecta cómo el DRA distribuye el tráfico y cómo se comporta durante las interrupciones de pares. Ver Enrutamiento Estándar para el flujo completo de decisiones de enrutamiento. Ver Enrutamiento Avanzado para anulaciones basadas en reglas.

Valor	Comportamiento
`:random`	El DRA elige un par coincidente al azar para cada solicitud. Esto distribuye la carga entre todos los pares elegibles.
`:failover`	El DRA prefiere el primer par coincidente. Utiliza pares posteriores solo cuando los pares de mayor prioridad no están disponibles.

Configuración de Pares

Cada entrada en la lista `peers` es un mapa que describe un único par Diameter. El mapa define su identidad, su dirección de red, el transporte que utiliza y si el DRA marca al par o espera ser marcado.

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

Parámetros por Par

Parámetro	Tipo	Requerido	Predeterminado	De
`host`	Cadena	Sí	-	La Id Diam par (valor coinc exac es se mayu minú regla enru Enru Avan requ coinc exac
`realm`	Cadena	Sí	-	El re Diam par.
`ip`	Cadena	Sí	-	La di princ utiliz cone
`additional_ips`	Lista	No	-	Lista direc para asoc mult Cuar estab

Parámetro	Tipo	Requerido	Predeterminado	De
				DRA Esta aplic `tran` `:dia` Ver S Multi
`port`	Entero	Sí	-	El pu Diam par (`3868`
`transport`	Átomo	Sí	-	Proto trans la co `:dia` para `:dia` para refer Trans conti
`source_ip`	Cadena	No	`listen_ip`	Direc local enlaz el DR este cone salie defe utiliz `list` servi

Parámetro	Tipo	Requerido	Predeterminado	De
`source_port`	Entero	No	-	Puert para cuan marc par (cone salie Cuar estab siste oper un pu efím pares comp mism puert desti estab valor Cada utiliz local
`initiate_connection`	Booleano	Sí	-	Cuar `true` cone (rol d Cuar `fals` espe par s (rol d Ver M Cone

Parámetro	Tipo	Requerido	Predeterminado	De
`origin_host`	Cadena	No	`service host.realm`	El `Or` que e prese par. l es se `host` es la ident este Estab clave par e el DR host (por realn El DR anula `CER`/ mens proto en es cone Ident Par.
`origin_realm`	Cadena	No	`service realm`	El `Or` que e prese par. l es se `real` que e realn par. l

Parámetro	Tipo	Requerido	Predeterminado	De
				esta con `orig` por s Ver l por F

> **SCTP multihoming:** un par que utiliza `:diameter_sctp` puede enlazar múltiples direcciones locales y remotas para resiliencia. SCTP Multihoming describe la configuración de multihoming y las listas de direcciones por par.

Transportes

Valor de Transporte	Protocolo	Referencia
`:diameter_tcp`	TCP	RFC 6733 (Base Diameter)
`:diameter_sctp`	SCTP	RFC 4960 (SCTP) - ver SCTP Multihoming

Modos de Conexión

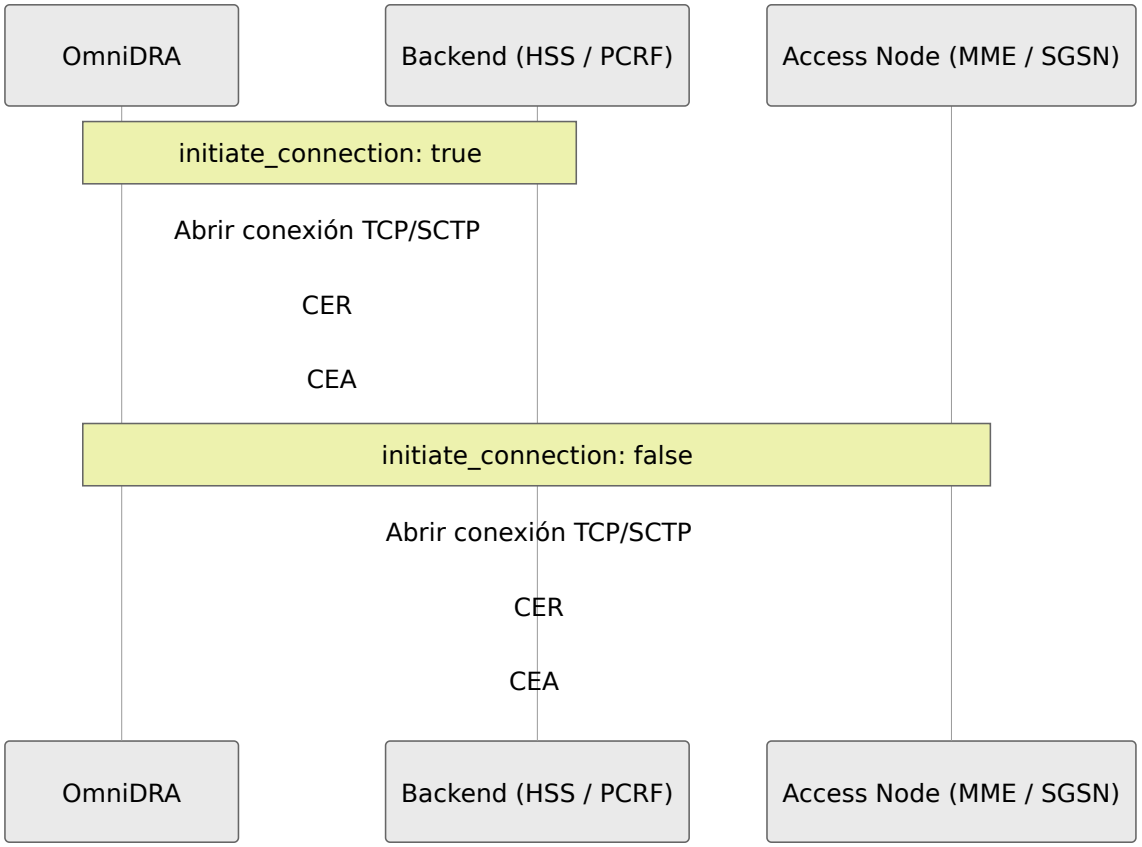
La clave `initiate_connection` controla si el DRA marca a un par o espera ser marcado.

Iniciar Conexión (`initiate_connection: true`)

- El DRA actúa como el cliente Diameter e inicia la conexión TCP/SCTP al par.
- Usa este modo para sistemas de backend como un HSS o PCRF.
- Si el par es inalcanzable, el DRA reintenta la conexión.

Aceptar Conexión (`initiate_connection: false`)

- El DRA actúa como el servidor Diameter y espera a que el par se conecte.
- Usa este modo para nodos del lado de acceso como un MME, SGSN o P-GW.
- El par debe estar presente en `peers`, a menos que establezcas `allow_undefined_peers_to_connect` en `true`.
- El DRA diferencia los pares entrantes por el `Origin-Host` que publicitan en el CER. Dos nodos Diameter pueden compartir una IP de origen. El DRA los empareja de forma independiente cuando cada uno presenta un `host` distinto.



Identidad por Par

El DRA tiene una identidad a nivel de servicio. Combina `host` y `realm` en el `Origin-Host` (por ejemplo `dra01.production.omnitouch`) y publicita `realm` como el `Origin-Realm`. Por defecto, el DRA presenta esta identidad a cada par.

Algunos pares esperan que el DRA presente una identidad diferente. Un socio de interconexión o de roaming a menudo requiere que el DRA use un host y realm del esquema de nombres del socio. Establece `origin_host` y

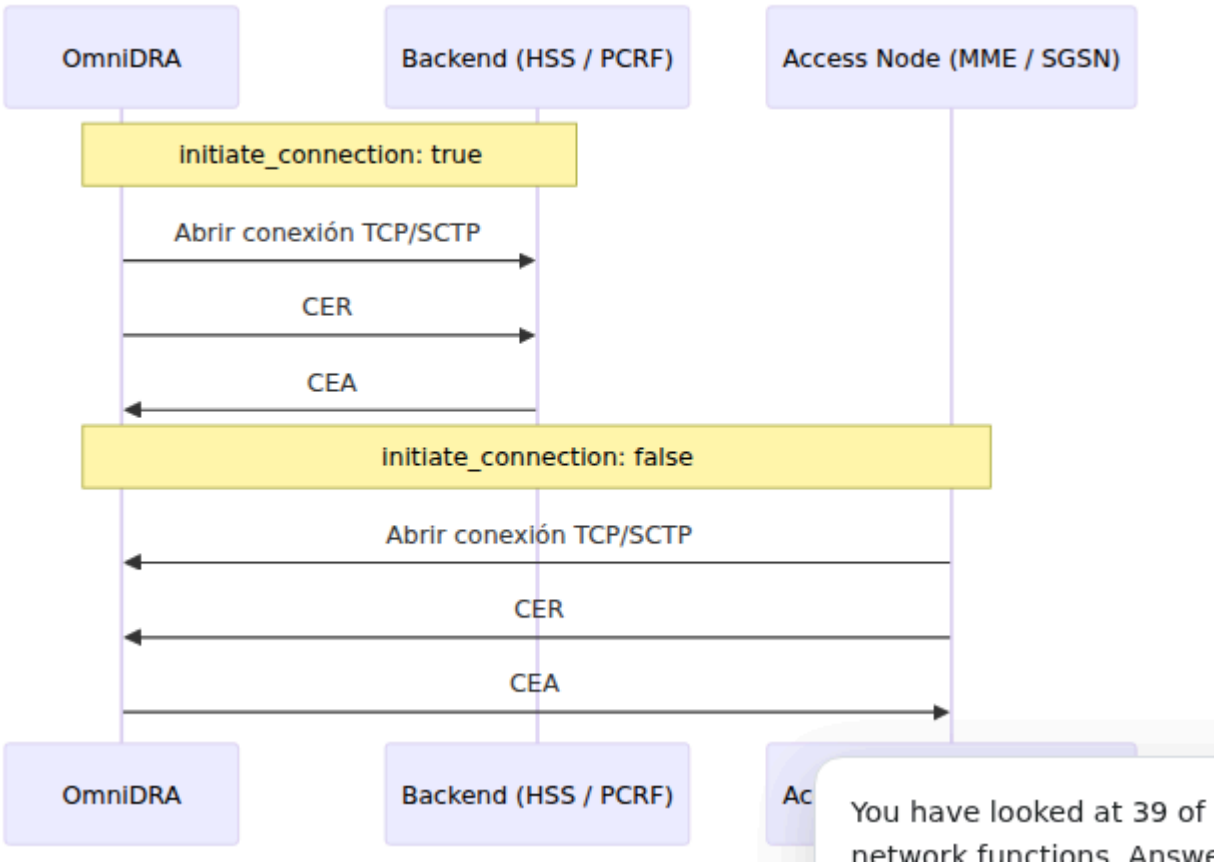
`origin_realm` en ese par para cambiar la identidad que el DRA presenta en esa conexión.

Estas dos claves son la identidad que el DRA presenta **a** el par. Son separadas de la propia `host` y `realm` del par, que son la identidad que el par publicita **al** DRA. El DRA aplica la anulación solo a esa conexión. Cada otro par continúa viendo la identidad a nivel de servicio.

```
peers: [
  %{
    # La propia identidad de este par (lo que publicita al DRA):
    host: "hss.partner.example",
    realm: "partner.example",
    ip: "10.0.0.9",
    port: 3868,
    transport: :diameter_sctp,
    initiate_connection: true,
    # La identidad que el DRA presenta A este par:
    origin_host: "dra-partner.epc.mnc001.mcc001.3gppnetwork.org",
    origin_realm: "epc.mnc001.mcc001.3gppnetwork.org"
  }
]
```

Ver Parámetros por Par para las definiciones de parámetros `origin_host` y `origin_realm`.

El diagrama a continuación muestra al DRA presentando su identidad predeterminada a un par y una identidad de anulación a un par asociado.



Comportamiento

La anulación cambia la identidad en el Intercambio de Capacidades (`CER` / `CEA`) en esa conexión. También cambia la identidad en los mensajes de protocolo base en esa conexión: los mensajes de vigilancia (`DWR` / `DWA`) y los mensajes de desconexión (`DPR` / `DPA`).

Puedes establecer cualquiera de las claves por su cuenta. Una clave no establecida vuelve al valor del servicio. Si estableces solo `origin_realm`, el DRA cambia solo el realm y mantiene el `Origin-Host` a nivel de servicio.

El DRA utiliza el `Origin-Host` de anulación para la detección de bucles en esa conexión. Si una solicitud ya lleva un AVP `Route-Record` que coincide con la anulación, el DRA aún detecta el bucle según RFC 6733 §6.1.8.

El DRA fija la identidad cuando envía el `CER`. Si cambias una anulación y recargas la configuración en tiempo de ejecución, el DRA restablece esa conexión para que la nueva identidad entre en efecto. Las conexiones a otros pares no se ven afectadas.

Ejemplo Completo

El siguiente ejemplo configura un DRA. El DRA acepta conexiones de un MME a través de SCTP. Marca a un HSS y a un PCRF a través de 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 a que el 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 la conexión desde un puerto local de
origen explícito
      %{
        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 la conexión
      %{
        host: "pcrf01.mvno.example.com",
        realm: "mvno.example.com",
        ip: "10.100.3.22",
        port: 3868,
        transport: :diameter_tcp,
        initiate_connection: true
```

```
      }
    ]
  }
```

Notas Operativas

- **Coincidencia de Nombres de Host** - El DRA coincide los valores de `host` de los pares exactamente y de manera sensible a mayúsculas y minúsculas. Las reglas en Enrutamiento Avanzado deben hacer referencia a la misma cadena `host` que configuras aquí.
- **Intercambio de Capacidades** - Al conectarse, el DRA y cada par intercambian aplicaciones e identidades soportadas a través de CER/CEA. El DRA publicita los valores de `host`, `realm`, `product_name` y `vendor_id` en esta etapa.
- **Vendor-ID 10415** - Este es el estándar `Vendor-Id` de 3GPP. Es el valor esperado para interfaces Diameter de 3GPP.
- **Selección de Pares** - Cuando múltiples pares coinciden con una solicitud, `peer_selection_algorithm` decide cuál usa el DRA. Ver Enrutamiento Estándar.

Consideraciones de Seguridad

- Establece `allow_undefined_peers_to_connect: false` en producción para que solo los pares conocidos puedan conectarse.
- Habilita `log_unauthorized_peer_connection_attempts: true` para registrar intentos de conexión rechazados para monitoreo de seguridad.
- Asegúrate de que las reglas del firewall estén alineadas con la `listen_ip` y `listen_port` configuradas.

Documentación Relacionada

- SCTP Multihoming - Transporte SCTP multi-homed y configuración de direcciones por par.

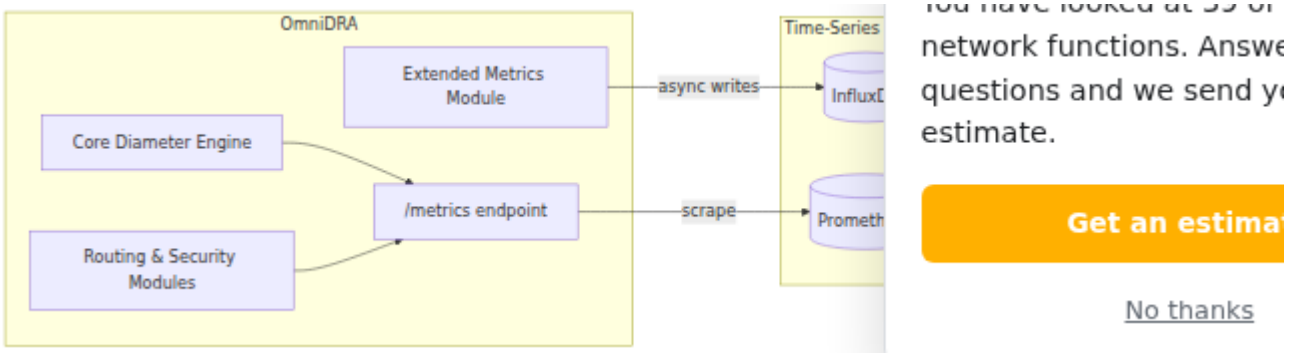
- Enrutamiento Estándar - Cómo el DRA selecciona pares y enruta solicitudes, incluyendo el comportamiento de `peer_selection_algorithm`.
- Enrutamiento Avanzado - Anulaciones de enrutamiento basadas en reglas claveadas en valores de `host` de pares.

Métricas y Monitoreo

OmniDRA expone métricas **Prometheus** completas. Estas cubren conectividad entre pares, tasas de mensajes, resultados de éxito y error, latencia de respuesta, tráfico descartado e intentos de conexión no autorizados. Esta página documenta cada métrica, sus etiquetas y consultas PromQL listas para usar. Úselas para construir paneles de control y alertas.

Flujo de Observabilidad

OmniDRA publica todas las métricas en formato de exposición de Prometheus en el endpoint `/metrics`. Prometheus extrae este endpoint a intervalos regulares y almacena las muestras. Luego, los paneles de control o herramientas de alerta consultan la serie temporal resultante.



Prometheus extrae las **métricas estándar de Prometheus** de `/metrics`. El **Módulo de Métricas Extendidas** publica telemetría de alta cardinalidad a nivel de suscriptor directamente en **InfluxDB**. Esto mantiene el conteo de series de Prometheus manejable (ver Módulo de Métricas Extendidas).

> OmniDRA también expone **métricas de tiempo de ejecución de BEAM/Erlang** en `/metrics`. Esta página no las documenta. Todos los contadores son acumulativos. Consúltalos con `rate()` para obtener tasas por segundo.

Métricas de Diameter Core

Estas métricas están siempre disponibles. Describen la salud de los pares y el flujo de mensajes a través del DRA.

Conectividad entre Pares

Estado del Par

- **Métrica**: `diameter_peer_status`
- **Tipo**: Gauge
- **Descripción**: Si el par está conectado (`1`) o no (`0`).

Etiquetas:

Etiqueta	Descripción
`origin_host`	Identidad Diameter del par (FQDN).
`ip`	Dirección IP del par.

Consultas de ejemplo:

```
# Verificar si un par específico está conectado
diameter_peer_status{origin_host="hss01.example.com"}

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

Para diagnosticar pares que fluctúan o que nunca alcanzan un estado conectado, consulte Solución de Problemas.

Aplicaciones Soportadas

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

- **Descripción**: Las aplicaciones que un par anunció durante el intercambio de capacidades (CER/CEA), cuando se conocen.

Etiquetas:

Etiqueta	Descripción
`origin_host`	Identidad Diameter del par (FQDN).
`supported_applications`	Las aplicaciones que el par anunció como soportadas.

Consultas de ejemplo:

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

Tasas de Mensajes

Conteo de Mensajes

- **Métrica**: `diameter_peer_message_count_total`
- **Tipo**: Counter
- **Descripción**: Número total de mensajes Diameter intercambiados con pares.

Etiquetas:

Etiqueta	Descripción
`origin_host`	Identidad Diameter del par.
`received_from`	Par que envió el mensaje.
`application_id`	Identificador de Aplicación Diameter. Ver Identificadores de Aplicación.
`cmd_code`	Código de Comando Diameter. Ver Códigos de Comando.
`application_name`	Nombre de la aplicación legible por humanos (por ejemplo, `3GPP_S6a`).
`cmd_name`	Nombre del comando legible por humanos (por ejemplo, `AIR`).
`direction`	`request` o `response`.

Consultas de ejemplo:

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

# Tasa total de mensajes por aplicación
sum by (application_name)
(rate(diameter_peer_message_count_total[5m]))
```

Tasas de Éxito y Error

Códigos de Resultado de Respuesta

- **Métrica**: `diameter_peer_message_result_code_count_total`
- **Tipo**: Counter
- **Descripción**: Número total de respuestas Diameter por código de resultado.

Etiquetas:

Etiqueta	Descripción
`origin_host`	Solicitante original.
`routed_to`	Par que envió la respuesta.
`application_id`	Identificador de Aplicación Diameter.
`cmd_code`	Código de Comando Diameter.
`application_name`	Nombre de la aplicación.
`cmd_name`	Nombre del comando.
`result_code`	Código de Resultado Diameter o Código de Resultado Experimental.

Consultas de ejemplo:

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

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

Códigos de Resultado Comunes:

Código	Nombre	Referencia
`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

Latencia

Retraso de Respuesta

- **Métrica**: `diameter_peer_last_response_delay`
- **Tipo**: Gauge
- **Descripción**: Retraso de respuesta más reciente en milisegundos (DRA → Par → DRA).

Etiquetas:

Etiqueta	Descripción
`origin_host`	Solicitante original.
`routed_to`	Par que envió la respuesta.
`application_name`	Nombre de la aplicación.
`cmd_name`	Nombre del comando.

Consultas de ejemplo:

```
# Retraso de respuesta más reciente desde HSS
diameter_peer_last_response_delay{routed_to="hss01.example.com"}

# Mayor retraso de respuesta reciente entre pares S6a
max(diameter_peer_last_response_delay{application_name="3GPP_S6a"})
```

> Este es un gauge que mantiene el retraso más reciente por par. No es un histograma, por lo que funciones de cuantil como `histogram_quantile` no se aplican.

Solicitudes No Respondidas

Conteo de Solicitudes No Respondidas

- **Métrica**: `diameter_peer_unanswered_request_count_total`
- **Tipo**: Counter
- **Descripción**: Solicitudes que el DRA envió pero que el par no respondió dentro del período de tiempo de espera.

Etiquetas:

Etiqueta	Descripción
`origin_host`	Solicitante original.
`routed_to`	Par que no respondió.
`application_id`	Identificador de Aplicación Diameter.
`cmd_code`	Código de Comando Diameter.
`application_name`	Nombre de la aplicación.
`cmd_name`	Nombre del comando.

Consultas de ejemplo:

```
# Tasa de solicitudes no 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])
))
```

Conexiones No Autorizadas

Intentos de Conexión No Autorizados

- **Métrica**: `diameter_peer_unauthorized_connection_count_total`
- **Tipo**: Counter
- **Descripción**: Intentos de conexión de pares no autorizados.

Etiquetas:

Etiqueta	Descripción
`origin_host`	Identidad reclamada del par no autorizado.
`supported_applications`	Aplicaciones anunciadas por el par.
`peer_ip`	Dirección IP del intento de conexión.

Consultas de ejemplo:

```
# Intentos de conexión no autorizados
rate(diameter_peer_unauthorized_connection_count_total[5m])

# Alertar sobre acceso no autorizado
diameter_peer_unauthorized_connection_count_total > 0
```

Para obtener más detalles sobre las métricas del módulo de seguridad (cortafuegos, limitación de tasa, ocultación de topología, saneamiento de AVP), consulte Seguridad.

Métricas del Módulo de Enrutamiento Avanzado

Estas métricas describen los resultados que producen las reglas de enrutamiento avanzadas. Consulte la documentación del módulo de seguridad en Seguridad para la telemetría de control de tráfico relacionada.

Tráfico Descartado

Conteo de Descartes

- **Métrica**: `diameter_advanced_routing_drop_count_total`
- **Tipo**: Counter

- **Descripción**: Solicitudes que la acción de enrutamiento avanzado `:drop` descartó. El DRA no envía respuesta.

Etiquetas:

Etiqueta	Descripción
`application_id`	Identificador de Aplicación Diameter.
`cmd_code`	Código de Comando Diameter.
`application_name`	Nombre de la aplicación.
`cmd_name`	Nombre del comando.

Consultas de ejemplo:

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

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

Cuando el DRA descarta tráfico:

- Una regla de enrutamiento avanzado coincide con una acción de **drop**.
- El DRA descarta el mensaje.
- El DRA no envía un mensaje de respuesta al par solicitante.

Limitación de Tasa y Sobrecarga

Respuestas de Error

- **Métrica**: `diameter_advanced_routing_error_count_total`

- **Tipo**: Counter
- **Descripción**: Respuestas de error que generó la acción de error de enrutamiento avanzado.

Etiquetas:

Etiqueta	Descripción
`result_code`	Código de Resultado Diameter enviado en la respuesta de error.
`application_id`	Identificador de Aplicación Diameter.
`cmd_code`	Código de Comando Diameter.
`application_name`	Nombre de la aplicación.
`cmd_name`	Nombre del comando.

Consultas de ejemplo:

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

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

# Detección de limitación de tasa o sobrecarga
diameter_advanced_routing_error_count_total{
  result_code="3004"
} > 100
```

Cuando el DRA genera respuestas de error:

- Una regla de enrutamiento avanzado coincide con una acción de **error** para un Código de Resultado especificado.
- El DRA genera una respuesta Diameter con el Código de Resultado especificado.
- El DRA envía la respuesta de vuelta al par solicitante. No reenvía la respuesta.

Códigos de Error Comunes Usados:

Código	Nombre	Significado
`3002`	DIAMETER_UNABLE_TO_DELIVER	Fallo de enrutamiento.
`3003`	DIAMETER_REALM_NOT_SERVED	Dominio no soportado.
`3004`	DIAMETER_TOO_BUSY	Protección contra sobrecarga.
`5012`	DIAMETER_UNABLE_TO_COMPLY	Error general.

Módulo de Métricas Extendidas

El **Módulo de Métricas Extendidas** proporciona telemetría y análisis avanzados. Úselo para analizar patrones de tráfico Diameter más allá de las métricas estándar. Actualmente rastrea intentos de conexión de suscriptores LTE. Correlaciona pares de mensajes de Solicitud de Información de Autenticación (AIR) y Respuesta (AIA) S6a.

Estas métricas llevan campos de **alta cardinalidad** como IMSI. Por lo tanto, el módulo las escribe de forma asincrónica en **InfluxDB** y no las expone en el endpoint `/metrics` de Prometheus. Esto mantiene el conteo de series de Prometheus limitado y evita la explosión de métricas.

Configuración

El módulo está deshabilitado por defecto. Configúrelo en `config/runtime.exs`:

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

Parámetro	Tipo	Requerido	Predeterm
`enabled`	Booleano	No	`false`
`attach_attempt_reporting_enabled`	Booleano	No	`false`

Conexión a InfluxDB

El Módulo de Métricas Extendidas escribe de forma asincrónica en InfluxDB a través de una conexión Instream (Instream). Configure el objetivo en `config :dra, DRA.Metrics.InfluxDB`.

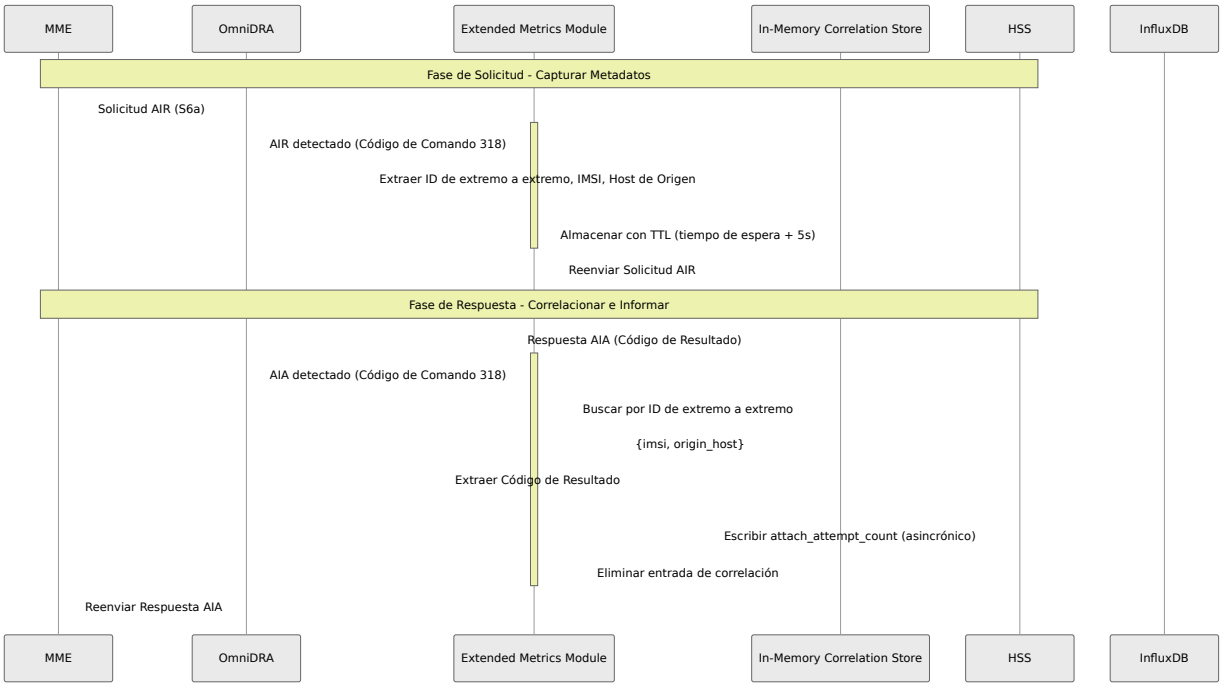
```
config :dra, DRA.Metrics.InfluxDB,
  auth: [username: "example_user", password: "example_password"],
  database: "dra",
  host: "127.0.0.15",
  port_udp: 8089,
  writer: Instream.Writer.Line
```

Parámetro	Tipo	Requerido	Predeterminado	Desc
`auth`	Lista de palabras clave	No	-	Credencial InfluxDB c `[username` `password:`
`database`	Cadena	Sí	-	Base de da objetivo qu mediciones métricas.
`host`	Cadena	Sí	-	Nombre de dirección I servidor In
`port_udp`	Entero	No	`8089`	Puerto UDP escribe dat protocolo d InfluxDB. U del oyente InfluxDB (8 defecto), n de la API H
`writer`	Módulo	No	`Instream.Writer.Line`	Módulo esc Instream. `Instream.` escribe cor protocolo d

Seguimiento de Intentos de Conexión

El módulo monitorea pares AIR/AIA S6a. Correlaciona cada solicitud con su respuesta a través del ID de extremo a extremo de Diameter. Luego emite una

métrica a nivel de suscriptor. La métrica lleva el par de origen y el código de resultado del HSS.



Medición: `attach_attempt_count`

Campos:

Campo	Descripción
`imsi`	El IMSI del suscriptor, tomado del AVP de Nombre de Usuario.

Etiquetas:

Etiqueta	Descripción
`origin_host`	El par que originó la solicitud de conexión.
`result_code`	El código de resultado Diameter de la respuesta del HSS.

Códigos de resultado escritos en esta medición son códigos Diameter estándar:

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

Consulte RFC 6733 para la lista completa de códigos de resultado.

Casos de Uso

- **Análisis de Tasa de Éxito de Conexión** - Rastrear intentos de conexión exitosos versus fallidos por código de resultado.
- **Solución de Problemas a Nivel de IMSI** - Identificar suscriptores individuales que tienen fallos de conexión.
- **Monitoreo del Rendimiento de la Red** - Monitorear patrones de intentos de conexión por origen (MME/SGSN).
- **Análisis de Roaming** - Analizar tasas de éxito de conexión de roaming entrante.

Notas Operativas

- Las métricas de intentos de conexión rastrean solo pares AIR/AIA S6a (Identificador de Aplicación `16777251`, Código de Comando `318`).
- Los metadatos de solicitud expiran después del tiempo de espera de solicitud configurado más 5 segundos.
- El procesamiento de métricas es asincrónico. No bloquea el flujo de mensajes.
- El módulo opera de forma independiente de los módulos de enrutamiento y transformación.

Tablas de Referencia

Referencia de Identificadores de Aplicación

ID	Aplicación	Descripción	Referencia
`16777251`	S6a/S6d	MME/SGSN ↔ HSS autenticación y suscripción	3GPP TS 29.272
`16777238`	Gx	PCEF ↔ PCRF control de políticas y cobros	3GPP TS 29.212
`16777216`	Cx	I-CSCF/S-CSCF ↔ HSS registro IMS	3GPP TS 29.229

Referencia de Códigos de Comando

Código	Comando	Aplicación	Referencia
`318`	AIR/AIA (Información de Autenticación)	S6a	3GPP TS 29.272

Notas

- Todas las métricas estándar están disponibles en el endpoint `/metrics` en formato Prometheus.
- El **Módulo de Métricas Extendidas** proporciona métricas adicionales específicas de S6a a través de InfluxDB.
- OmniDRA también expone **métricas de tiempo de ejecución de BEAM/Erlang**. Esta página no las documenta.
- Todos los contadores son acumulativos. Consúltalos con `rate()` para obtener tasas por segundo.

- Solo el Módulo de Métricas Extendidas utiliza etiquetas de alta cardinalidad como IMSI. Esto evita la explosión de métricas.

Documentación Relacionada

- Solución de Problemas - Diagnóstico de problemas de conectividad, enrutamiento y entrega.
- Seguridad - Detalles sobre métricas del módulo de seguridad (cortafuegos, limitación de tasa, ocultación de topología, saneamiento de AVP).

Conectividad entre pares y tiempo de asentamiento

Este módulo controla cuán rápido un par de Diameter se vuelve enrutable después de conectarse. También previene el tráfico hacia pares que parpadean. Un par parpadeante se conecta y desconecta repetidamente.

Antecedentes

Antes de que OmniDRA enrute a un par, el par debe completar el intercambio de capacidades (CER/CEA) y ser promovido a servicio. Por defecto, el RFC 3539 mantiene a un par *reconectando* en el estado REOPEN. El par permanece allí hasta que pasa sucesivos intercambios de vigilancia (DWR/DWA). El temporizador de vigilancia **Tw** regula estos intercambios, aproximadamente 30 segundos con el valor predeterminado estándar de 30000 ms. Esta retención se aplica incluso si CER/CEA ya tuvo éxito.

OmniDRA elimina esa demora implícita. La vigilancia promueve a los pares reconectando de inmediato. OmniDRA reemplaza la demora con un **tiempo de asentamiento**. El tiempo de asentamiento es una retención explícita y configurable que se aplica después de que el par se conecta. Durante la retención, el par está conectado pero no es enrutable. Si se desconecta dentro de la ventana, OmniDRA cancela la retención y no envía tráfico a él.

Configuración

Ambas claves son opcionales y se encuentran en el mapa de configuración `:diameter`. Si no estableces ninguna clave, los pares se enrutan tan pronto como se conectan.

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

Parámetro	Tipo	Requerido	Predeterminado	Desc
`peer_settle_time_ms`	Entero	No	`0`	Cuánto (ms) un recién conecta permar activo a que el D enrute t él. `0` er inmedia Un valo mantier durante duració Cancela retenci par se descon durante ventana
`watchdog_timer_ms`	Entero	No	`30000`	Sobresc para el tempori vigilanc RFC 35 interval activida DWR/DW Redúce una det más rá pares m Esto ge

Parámetro	Tipo	Requerido	Predeterminado	Desc
				tráfico c vigilanc

Establece el tiempo de asentamiento no más alto que el intervalo de parpadeo en el peor de los casos que debes soportar. Un valor más largo aumenta la protección contra parpadeos. También retrasa el regreso de un par saludable después de un reinicio legítimo.

Monitoreo

Un par en su ventana de asentamiento informa **down** en el medidor `diameter_peer_status` (`0` = down/no enrutable aún, `1` = enrutable). Informa como down hasta que se vuelve enrutable. Por lo tanto, la métrica refleja lo que el DRA puede enrutar.

El inicio produce estas líneas de registro:

Mensaje de registro	Cuándo
`peer connected, settling before routable`	Par conectado, ventana de asentamiento iniciada (incluye `settle_time_ms`).
`peer disconnected during settle window; never became routable`	Par desconectado antes de que se asentara. El DRA no enruto tráfico.
`destination peer is settling; request not yet routable...`	Una solicitud llegó para un par que aún se está asentando. El DRA respondió `DIAMETER_UNABLE_TO_DELIVER (3002)`.
`no suitable peer to route request...`	El DRA no puede enrutar la solicitud a un destino que **no** se está asentando (down, desconocido o mal configurado).

Las dos últimas líneas distinguen un asentamiento transitorio esperado de un destino genuinamente faltante.

Referencia

- RFC 3539 - Máquina de estados del temporizador de vigilancia de Diameter y el temporizador de vigilancia (Tw).
- RFC 6733 - Protocolo base de Diameter (CER/CEA, DWR/DWA).

Dirección de Roaming (SoR)

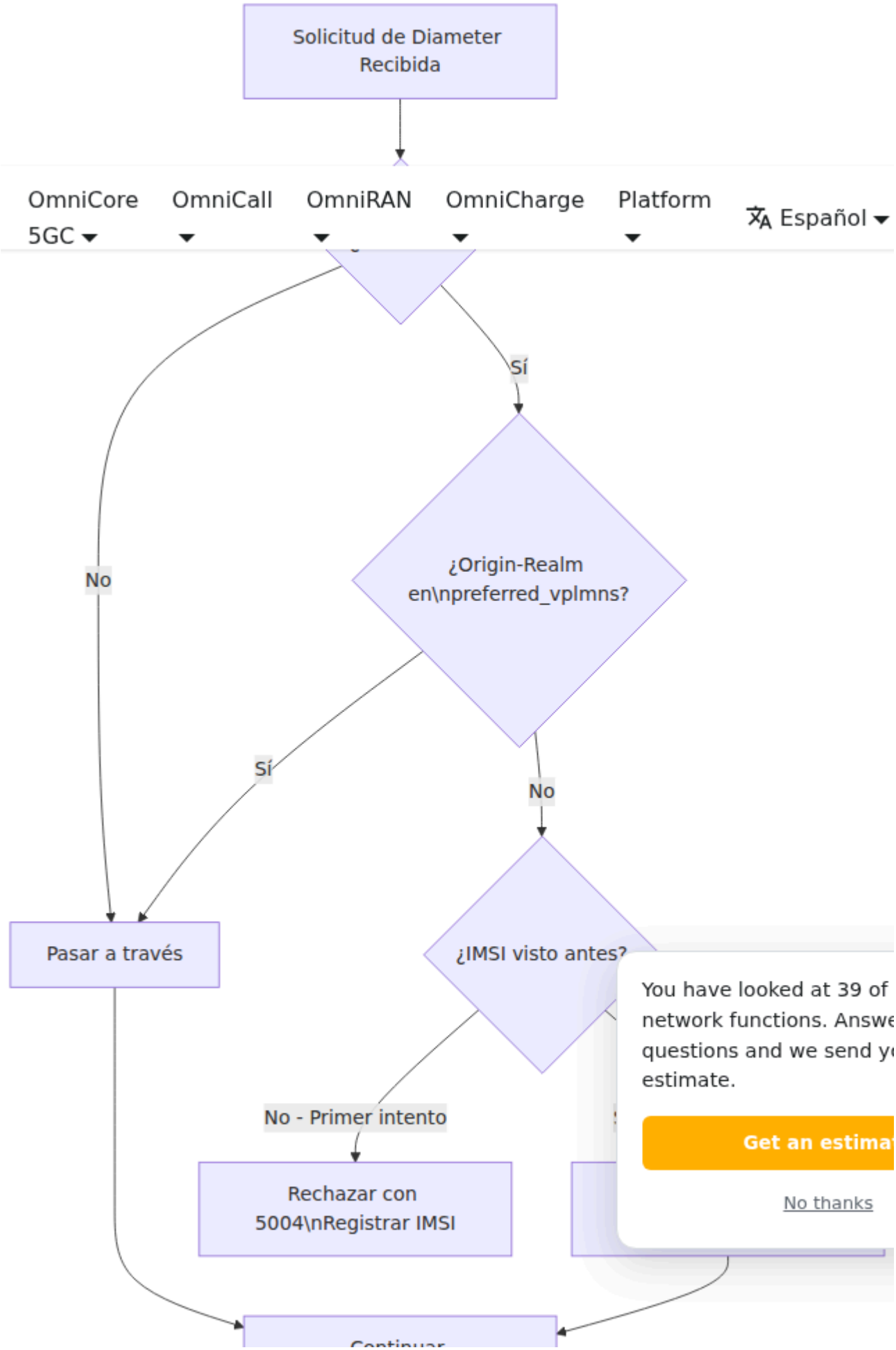
La Dirección de Roaming (SoR) permite que el HPLMN influya en qué VPLMN se conecta un suscriptor de roaming. Cuando un suscriptor intenta registrarse a través de un VPLMN no preferido, el DRA intercepta la S6a Update-Location-Request (ULR). Rechaza la ULR con **DIAMETER_ERROR_ROAMING_NOT_ALLOWED (5004)**. Esto provoca que el UE se desasocie y vuelva a asociarse. El UE selecciona un VPLMN preferido si está disponible.

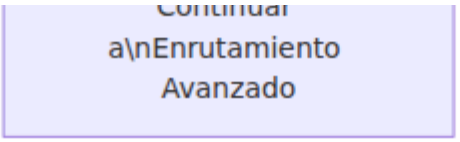
Si la cobertura preferida no está disponible y el suscriptor regresa al mismo VPLMN no preferido, el módulo permite la solicitud en el siguiente intento.

3GPP TS 29.272 Sección 5.2.1.1 y 3GPP TS 23.122 definen SoR.

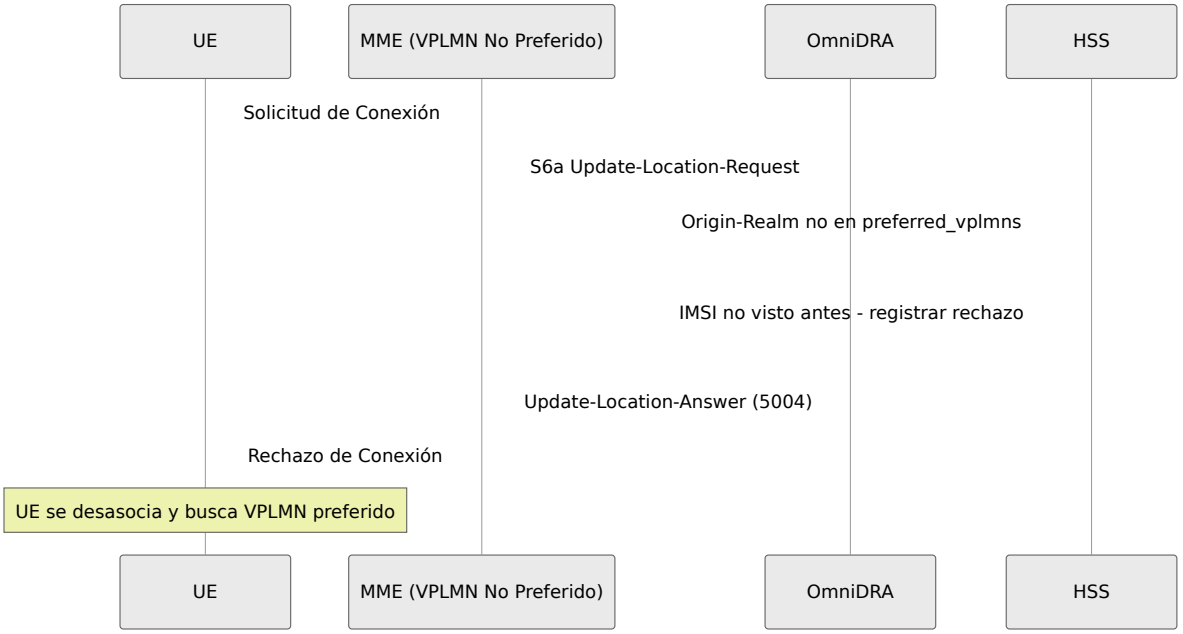
Cómo Funciona

El módulo opera dentro de la tubería de procesamiento de solicitudes del DRA. Se ejecuta **antes** del Enrutamiento Avanzado. El módulo evalúa solo las S6a Update-Location-Requests (Application-Id `16777251`, Command-Code `316`). Todos los demás mensajes de Diameter pasan sin modificar.

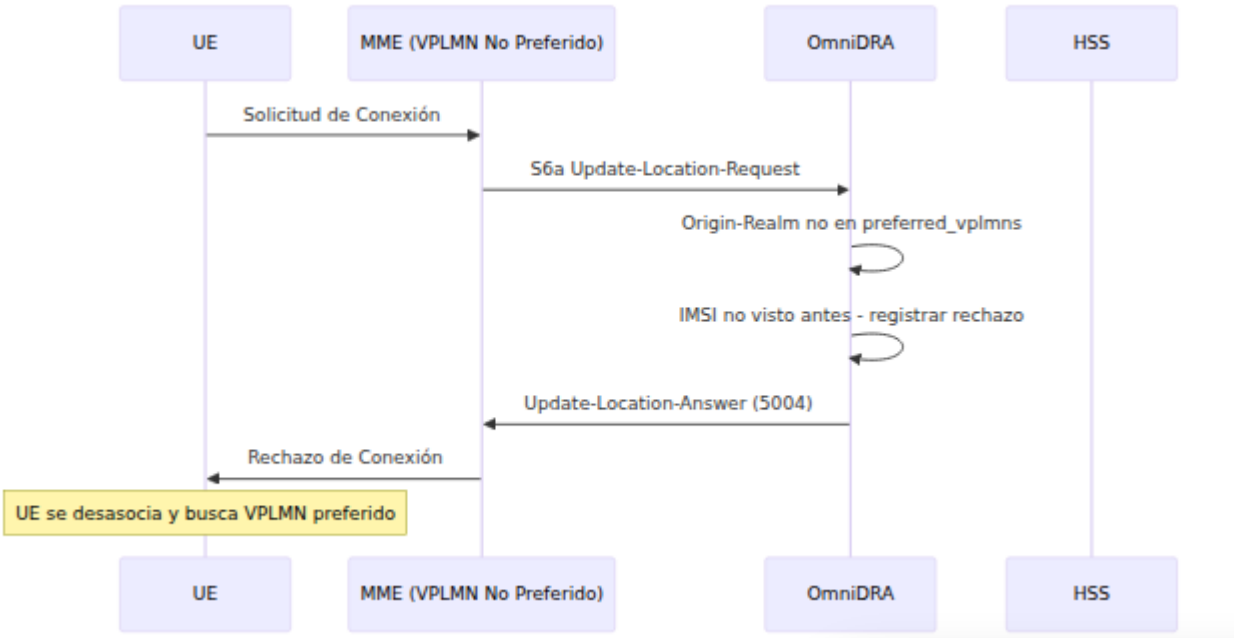

Solicitud de Diameter Recibida
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
Sí
No
¿Origin-Realm en\npreferred_vplmns?
Sí
No
Pasar a través
¿IMSI visto antes?
No - Primer intento
Rechazar con 5004\nRegistrar IMSI
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Secuencia: Primer Intento (Rechazado)



Secuencia: Segundo Intento (Permitido)



Posición en la Tubería

La Dirección de Roaming se ejecuta **antes** del Enrutamiento Avanzado en la tubería de procesamiento de solicitudes. Si SoR rechaza una solicitud, ese mensaje nunca llega al Enrutamiento Avanzado.



Seguimiento de Suscriptores

El módulo rastrea cada suscriptor rechazado por IMSI. Cada registro incluye un conteo de rechazos y una marca de tiempo. El módulo limpia automáticamente los registros después de que expira el TTL configurado.

Cuando un IMSI rastreado alcanza el umbral de `max_rejections` y envía otra ULR, el módulo permite la solicitud. Luego elimina el registro de seguimiento.

Si no llega ninguna ULR adicional dentro de la ventana TTL, el registro expira. El siguiente ciclo de limpieza lo elimina. El módulo trata una ULR posterior del mismo IMSI como un primer intento y lo rechaza nuevamente.

Configuración

Configure el módulo bajo `module_roaming_steering` en `config/runtime.exs`.

```
module_roaming_steering: %{
  # Habilitar o deshabilitar el módulo
  enabled: false,

  # Número de rechazos antes de permitir el paso
  max_rejections: 1,

  # Tiempo en segundos antes de que expire un registro de IMSI
rastreado
  rejection_ttl_seconds: 300,

  # Código de Resultado de Diameter devuelto en la respuesta de
rechazo
  rejection_result_code: 5004,

  # Lista de valores de Origin-Realm considerados VPLMNs
preferidos
  # Las ULR de estos reinos siempre se permiten pasar
  preferred_vplmns: [
    "epc.mnc001.mcc001.3gppnetwork.org",
    "epc.mnc002.mcc001.3gppnetwork.org"
  ]
}
```

Parámetros

Parámetro	Tipo	Requerido	Predeterminado	
`enabled`	Booleano	Sí	`false`	Ha Di las
`children`	Lista	No	`[]`	Pr cu [D pa se
`max_rejections`	Entero	No	`1`	Nú su pe el la
`rejection_ttl_seconds`	Entero	No	`300`	Tie re lo int tra int int
`rejection_result_code`	Entero	No	`5004`	Có re DI se
`preferred_vplmns`	Lista	No	`[]`	Lis ide sie Or

Ejemplos de Configuración

Rechazar Una Vez, Luego Permitir

Este es el comportamiento predeterminado. El módulo rechaza a un suscriptor en un VPLMN no preferido una vez. Si el suscriptor regresa, el módulo reenvía la ULR al 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"
  ]
}
```

Cómo funciona: Una ULR llega de un MME en `epc.mnc003.mcc310.3gppnetwork.org`, que no está en la lista preferida. El módulo lo rechaza con el Código de Resultado 5004. El UE se desasocia y trata de encontrar una red preferida. Si regresa al mismo VPLMN no preferido dentro de 300 segundos, el módulo reenvía la ULR normalmente.

Caso de uso: Dirección de roaming estándar. El operador tiene acuerdos de roaming preferidos con redes asociadas específicas.

Múltiples Rechazos Antes de Permitir

Utilice esto cuando el operador quiera que el UE realice múltiples intentos antes de volver a un VPLMN no preferido.

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

Cómo funciona: El módulo rechaza al suscriptor tres veces. Luego permite que el suscriptor pase en el cuarto intento. El TTL se extiende a 600 segundos para los ciclos de reintento adicionales.

Caso de uso: Áreas urbanas densas donde existe cobertura preferida. El UE puede necesitar múltiples intentos de conexión para adquirirla.

Métricas

Rechazos

Métrica: `diameter.roaming_steering.reject.count` **Tipo**: Contador **Descripción**: El DRA incrementa este contador cada vez que el módulo SoR rechaza una ULR. **Etiquetas**:

- `origin_realm` - Origin-Realm del VPLMN no preferido
- `result_code` - Código de Resultado de Diameter devuelto en el rechazo
- `imsi` - IMSI del suscriptor rechazado

Permitidos (VPLMN Preferido)

Métrica: `diameter.roaming_steering.allow.count` **Tipo**: Contador **Descripción**: El DRA incrementa este contador cuando el módulo permite pasar una ULR. **Etiquetas**:

- `origin_realm` - Origin-Realm del VPLMN de origen

- `reason` - Por qué el módulo permitió la solicitud: `preferred_vplmn` o `max_rejections_reached`
- `imsi` - IMSI del suscriptor. Presente solo cuando la razón es `max_rejections_reached`

Respuestas de Error

Métrica: `diameter.roaming_steering.error.count` **Tipo**: Contador **Descripción**: El DRA incrementa este contador cuando el procesador devuelve un mensaje de respuesta de error para un rechazo de SoR. **Etiquetas**:

- `result_code` - Código de Resultado de Diameter
- `application_id` - Application-Id de Diameter (numérico)
- `cmd_code` - Command-Code de Diameter (numérico)
- `application_name` - Nombre de la aplicación legible por humanos
- `cmd_name` - Nombre del comando legible por humanos

Solución de Problemas

Suscriptores Siempre Rechazados (Nunca Permitidos)

Síntomas: El módulo rechaza repetidamente a los suscriptores en VPLMNs no preferidos. Nunca se conectan con éxito.

Causas posibles:

- `rejection_ttl_seconds` es demasiado corto. El registro de seguimiento expira antes de que el suscriptor vuelva a intentar.
- `max_rejections` es mayor que el número de reintentos que realiza el UE antes de rendirse.

Resolución:

1. Aumente `rejection_ttl_seconds` para adaptarse al tiempo de reintento del UE.

2. Confirme que `max_rejections` es un valor que el UE puede alcanzar dentro de su ciclo de reintentos.

Suscriptores en VPLMNs Preferidos Siendo Rechazados

Síntomas: El módulo rechaza ULRs de redes asociadas preferidas con 5004.

Causas posibles:

- El Origin-Realm del VPLMN preferido no está en `preferred_vplmns`.
- La cadena de Origin-Realm no coincide exactamente. La coincidencia es sensible a mayúsculas y minúsculas.

Resolución:

1. Verifique el Origin-Realm del par en los registros del DRA en el momento de la conexión.
2. Confirme que la cadena exacta de Origin-Realm esté en la lista de `preferred_vplmns`.

Módulo No Efectivo

Síntomas: El módulo reenvía ULRs de VPLMNs no preferidos sin rechazo.

Causas posibles:

- `enabled` está configurado como `false`.
- El proceso del módulo no se está ejecutando. Verifique el supervisor.

Resolución:

1. Confirme `enabled: true` en la configuración.
2. Confirme que el DRA se reinició después del cambio de configuración.

Referencia

Códigos de Diameter

Código	Nombre	Descripción	Ref
5004	DIAMETER_ERROR_ROAMING_NOT_ALLOWED	El suscriptor no puede hacer roaming en este VPLMN	3GI 29. Sec 7.4

Comandos S6a

Command-Code	Nombre	Descripción	Referencia
316	Update-Location-Request/Answer (ULR/ULA)	El MME envía esto al HSS durante la conexión para actualizar el nodo de servicio del suscriptor	3GPP TS 29.272 Sección 7.2.3

Identificadores de Aplicación

ID	Interfaz	Descripción	Referencia
16777251	S6a/S6d	Autenticación y gestión de suscripciones de MME/SGSN a HSS	3GPP TS 29.272

Vínculo de Sesión Rx

Vínculo de Sesión Rx mantiene cada mensaje de una sesión Rx en el PCRF que la posee. Existe porque Rx es la única interfaz cuya solicitud de terminación de sesión no lleva nada para enrutar.

El Problema

3GPP TS 29.214 §5.6.5 hace que `Destination-Host` sea **opcional** en la `Session-Termination-Request` de Rx. Define **ningún** `Subscription-Id` y **ningún** `Framed-IP-Address` en el mensaje en absoluto. Este segundo punto es estructural. Esos AVPs no están simplemente ausentes en la práctica. No están en la definición del mensaje, por lo que ningún AF conforme puede proporcionarlos.

Por lo tanto, si el STR lleva un `Destination-Host` depende del AF. Algunas implementaciones de AF no envían uno. Una captura representativa muestra estos AVPs:

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

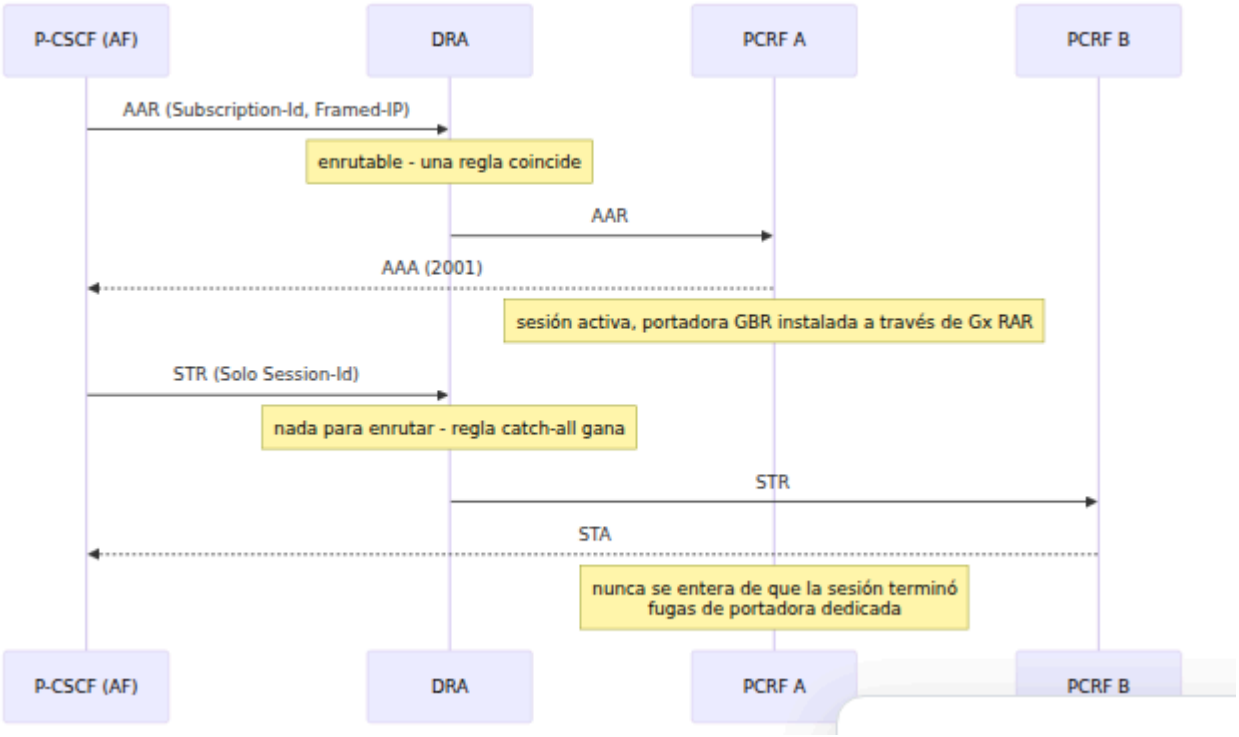
El `Session-Id` es el propio identificador opaco del AF. Según RFC 6733 §8.8, es globalmente único y estable durante la vida de la sesión. No identifica ni al suscriptor ni al servidor.

Compáralo con el `AA-Request` que abrió la sesión. Esa solicitud lleva `Subscription-Id` y `Framed-IP-Address`, por lo que una regla de Enrutamiento Avanzado puede enrutarla. Compáralo también con **Gx**. El ABNF de `CC-Request` (TS 29.212 §5.6.2) incluye `[ Destination-Host ]`, `*[ Subscription-Id ]` **y** `[ Framed-IP-Address ]`. Esta es exactamente la razón por la que Gx no necesita un equivalente de este módulo.

TS 29.213 §7.3.5 pone la carga sobre el DRA de todos modos. Un cliente "debe ser capaz de enviar cada mensaje de una sesión al DRA". Un bypass es solo

algo que "puede ser configurado".

Con un solo PCRF, esta asimetría es invisible. Con más de un PCRF detrás del DRA, no lo es:



Este fallo es silencioso. RFC 6733 §8.4.2 no prescribe ningún código de resultado para un STR que nombra una sesión desconocida. La máquina de estados §8.1 maneja "STR recibido" desde *cualquier* estado con "enviar STA, limpiar". Un PCRF que responde `2001` por una sesión que nunca tuvo es, por lo tanto, conforme, y las implementaciones reales hacen esto. No esperes que los códigos de resultado revelen esto. (`5002 DIAMETER_UNKNOWN_SESSION_ID` está prescrito para el **ASA**, §8.5.2, no para el STA.)

Lo mismo se aplica, menos visiblemente, a un *pool* de PCRFs intercambiables. Una regla que enruta a `peers: [a, b]` selecciona por mensaje. Un AAR y su STR pueden, por lo tanto, aterrizar en diferentes miembros del mismo pool.

Cuándo Lo Necesitas

- Migrar entre PCRFs, o ejecutar dos en paralelo para una prueba por suscriptor.
- Cualquier regla de Enrutamiento Avanzado que envíe Rx a más de un par.

No lo necesitas con un solo PCRF. Está deshabilitado por defecto.

Cómo Funciona

Mensaje	Acción
`AA-Answer` (265, `Result-Code` 2001)	Vincular `Session-Id` → el PCRF que respondió
`AA-Request` (265)	Enrutar al PCRF vinculado
`Session-Termination-Request` (275)	Enrutar al PCRF vinculado
`Session-Termination-Answer` (275)	Liberar el vínculo, **solo** si provino del PCRF vinculado y su `Result-Code` es final (2xxx o 5xxx)

Los Vínculos Siguen Hechos, No Intenciones

El DRA se vincula en la **respuesta**, nunca en la solicitud. Una solicitud solo dice dónde el DRA *intentó* enviar algo. La respuesta prueba que existe una sesión en ese nodo. Esto tiene tres consecuencias:

- Un AAR respondido con un error, o que nunca fue respondido, no deja ningún vínculo.
- El DRA libera una sesión cuando llega su STA, no cuando enruta el STR. RFC 6733 §5.5.4 sanciona explícitamente la retransmisión después de un failover. Una liberación temprana enrutará incorrectamente el reintento, el mismo fallo que este módulo previene.
- Una solicitud nunca reescribe un vínculo. Por lo tanto, una interrupción transitoria de un par no puede migrar silenciosamente una sesión activa a un PCRF diferente.

La liberación requiere **ambas** condiciones. El STA debe provenir del par que posee el vínculo, y su `Result-Code` debe ser final. RFC 6733 §8.4.2 hace que el

servidor libere sus recursos cuando envía el STA, por lo que 2xxx y 5xxx son igualmente finales. Un 3xxx/4xxx es transitorio y el AF puede retransmitir, por lo que el vínculo sobrevive para el reintento.

La condición de propiedad no es una limpieza defensiva. Sin ella, el módulo falla en el escenario exacto para el que fue construido. El propietario se vuelve inaccesible. El STR cae en otro PCRF. Ese PCRF responde `2001` por una sesión que nunca tuvo. El DRA luego destruye el vínculo con el propietario mientras su portadora aún está activa. Esto te deja en una situación peor que sin el módulo, porque la finalización ahora *parece* fijada. La condición de propiedad es el espejo de `insert_new` en el lado del vínculo. Un segundo PCRF no puede robar un vínculo ni destruir uno.

Lo Que No Sobrescribirá

El módulo solo redirige el enrutamiento lejos de una decisión de retransmisión. Todo lo demás es un resultado deliberado, y el módulo lo deja en paz:

Resultado	Fuente	Honrado
`:discard`	una regla `route: :drop`	sí
`{:answer_message, code}`	una regla `route: {:error, code}`	sí
`{:answer_message, 3002}`	Enrutamiento Avanzado fallando cerrado sin par conectado	sí
`Destination-Host` explícito	el AF nombrando su servidor (RFC 6733 §6.1.5)	sí

El costo es que el DRA no puede desmantelar una sesión mientras cada par al que apuntan sus reglas sea inaccesible. Este es el intercambio correcto. Un operador que drena un nodo, o falla cerrado a propósito, no pide ser enrutado alrededor.

Ordenamiento

Esta es la única etapa que se ejecuta **después** del Enrutamiento Avanzado. Tiene que ejecutarse allí. Un STR solo puede coincidir con una regla catch-all, y el vínculo debe poder superarla.

El observador de la ruta de respuesta se ejecuta **antes** de Transformación Avanzada. La Transformación Avanzada puede reescribir o eliminar `Session-Id` y `Result-Code` en las respuestas. El vínculo debe basarse en el `Session-Id` que el AF realmente utilizó. El módulo es deliberadamente insensible a ocultamiento de topología. Lee el par que responde desde sus capacidades de Diameter, no desde un AVP `Origin-Host`. Por lo tanto, una reescritura de AVP no puede engañarlo.

Configuración

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

Clave	Propósito
`enabled`	Desactivado por defecto. Independiente de `module_subscriber_lookup`. Los dos resuelven problemas diferentes, y puedes querer uno solo.
`ttl_seconds`	Guardián de fugas para sesiones cuya STA nunca llega. **No** es un temporizador de sesión.

Ajusta `ttl_seconds` **por encima** del `Authorization-Lifetime` del AF. El DRA nunca desasocia una llamada larga en medio de la sesión. Un P-CSCF que anuncia `Authorization-Lifetime: 36000` necesita más de 10 horas. El valor predeterminado de 12 horas le conviene. Las barridas se realizan cada

`ttl_seconds / 2` (mínimo 60s). El DRA recupera un vínculo filtrado dentro de 1.5 × TTL.

Observabilidad

Tres eventos de telemetría, todos llevando `session_id`:

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

Un cuarto evento, `[:diameter, :rx_session_binding, :reap]`, lleva un `count`. El DRA lo emite solo cuando la barrida de TTL encuentra vínculos expirados. Un conteo de recolección distinto de cero significa que las STAs están desapareciendo. Alerta sobre este conteo.

El DRA expone un conteo de vínculos activos, y puede listarlos como `{session_id, origin_host, inserted_at}`.

Limitaciones

- **Los vínculos están en memoria y son locales al nodo.** Un reinicio del DRA, o un fallo de un hermano bajo el supervisor `:one_for_all`, los elimina. Las sesiones en curso luego regresan a lo que digan las reglas para el resto de la llamada. El DRA no comparte vínculos entre DRAs. Por lo tanto, un AF debe alcanzar el mismo DRA para ambas mitades de una sesión.
- **El DRA vincula al par adyacente.** En modo de retransmisión, el DRA puede agregar un Route-Record pero no insertar un `Destination-Host` (RFC 6733 §6.1.9). Por lo tanto, el DRA no puede fijar un PCRF que esté detrás de otro agente. Solo ese agente puede.

- **Una AA-Answer tardía puede resucitar una sesión liberada.** Si el DRA procesa la respuesta de una actualización de AAR después del STA, `insert_new` recrea la fila. La sesión ha terminado, por lo que esta es una fila filtrada, no un enrutamiento incorrecto. Los Session-Ids no se reutilizan (RFC 6733 §8.8). El valor de `ttl_seconds` limita la fila, y aparece en el contador `:reap`.
- **El DRA no puede liberar una sesión mientras el Enrutamiento Avanzado falla cerrado** para Rx. El resultado `{:answer_message, 3002}` es indistinguible de un `route: {:error, 3002}` deliberado por parte de un operador. Mantén cada PCRF que tenga sesiones activas en la lista de pares de la regla. Este caso entonces no surge.
- **`children` es requerido en la configuración.** `DRA.Supervisor` inicia los procesos de un módulo solo cuando la configuración lleva una lista de `children`. Con `enabled: true` y sin `children`, el GenServer nunca se inicia. Cada punto de entrada se degrada a "sin vínculo", y nada te advierte.
- **El DRA solo vincula la dirección AF → PCRF.** Los `RAR` y `ASR` iniciados por el PCRF llevan un `Destination-Host` obligatorio (RFC 6733 §8.3.1, §8.5.1) y enrutan sobre eso. No necesitan vínculo.
- **El DRA solo vincula el id de aplicación Rx de Rel-7+ (16777236).** No vincula sesiones en el id de Rx/Gq de Rel-6 heredado (16777229).
- **El DRA emite telemetría pero no la exporta.** Los cuatro eventos anteriores no tienen definiciones de `Telemetry.Metrics`, por lo que nada llega a Prometheus hoy. Esta es la misma brecha que tienen los eventos de `subscriber_lookup` preexistentes. Usa el conteo de vínculos activos desde un shell remoto hasta que eso esté conectado.

Multihoming SCTP

El multihoming SCTP proporciona redundancia de red para el transporte de Diameter. Permite que cada punto final se vincule a múltiples direcciones IP dentro de una sola asociación. Si la ruta de red principal falla, SCTP cambia automáticamente a una ruta alternativa. No interrumpe la sesión de Diameter.

Por qué es importante el multihoming

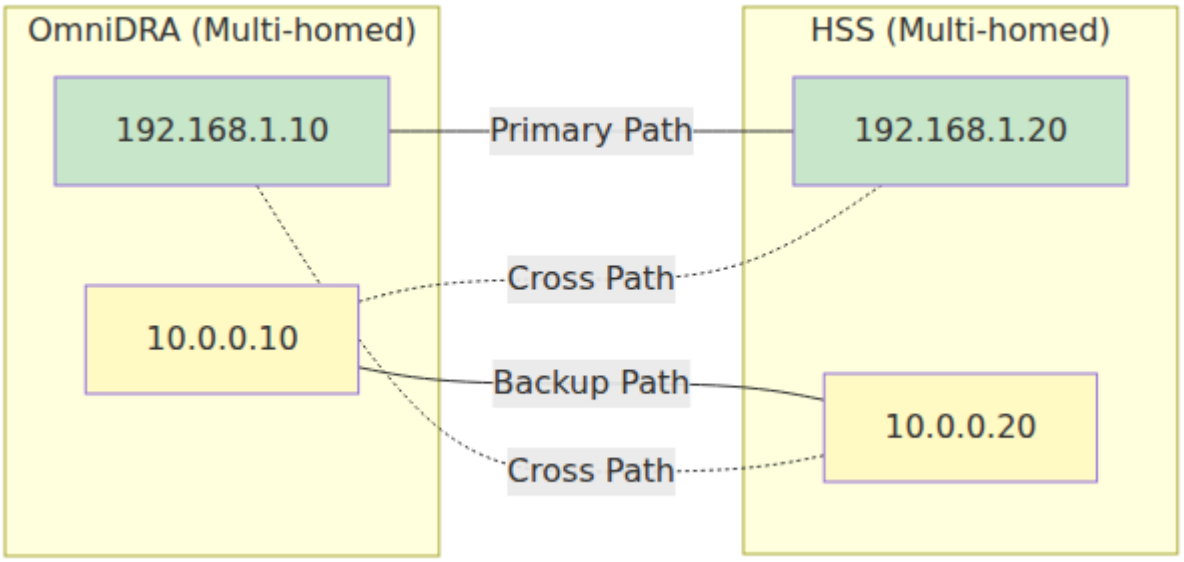
RFC 6733 define Diameter sobre SCTP. El comportamiento de multihoming descrito aquí es una capacidad nativa de **SCTP**. RFC 4960 lo especifica. TCP vincula una asociación a un único par de IP de origen/destino. SCTP permite que una asociación abarque múltiples direcciones IP en cada lado. Esto proporciona a OmniDRA resiliencia a nivel de transporte contra la pérdida de una ruta de red individual, NIC, conmutador o segmento enrutado. La conexión Diameter no se desmantela ni se restablece.

Características clave:

- Los **heartbeats** de SCTP monitorean continuamente todas las rutas de red en la asociación.
- La **conmutación automática** ocurre a nivel de núcleo si la ruta principal se vuelve inalcanzable.
- El cambio de ruta no interrumpe la sesión de Diameter. La asociación y su estado de Diameter persisten.
- El núcleo del sistema operativo selecciona la ruta automáticamente.

Cómo funciona

OmniDRA y un par establecen una asociación multihomed. Cada punto final anuncia varias direcciones IP. El núcleo selecciona una ruta principal para la transmisión de datos. Trata las rutas restantes como rutas de reserva. Si la ruta principal falla, el núcleo promueve automáticamente una ruta de reserva.



En el diagrama anterior, la **Ruta Principal** (verde) transporta tráfico en condiciones normales. La **Ruta de Respaldo** (amarillo) y las **Rutas Cruzadas** (discontinuas) son todas parte de la misma asociación SCTP. Permanecen disponibles para una conmutación por error inmediata.

Configuración

Configure el multihoming en `config/runtime.exs` en dos lugares. El primero son las direcciones de escucha locales a las que OmniDRA se vincula. El segundo son las direcciones remotas de cada par multihomed. Consulte la Referencia de Configuración para la estructura completa de configuración.

Direcciones de Escucha DRA

El campo `listen_ip` controla qué direcciones IP locales OmniDRA se vincula para conexiones entrantes. Acepta una única cadena IP (compatible hacia atrás) o una lista de cadenas IP. Use una lista para habilitar el multihoming.

```
%{
  # IP única (compatible hacia atrás)
  listen_ip: "192.168.1.10",

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

  listen_port: 3868
}
```

Parámetro	Tipo	Requerido	Predeterminado	Descripción
`listen_ip`	Cadena o Lista de Cadenas	Sí	-	Dirección(es) IP local(es) a las que OmniDRA se vincula para conexiones entrantes. Una única cadena vincula una dirección. Una lista de cadenas habilita el multihoming SCTP en todas las direcciones listadas. Para **transporte TCP**, OmniDRA utiliza solo la primera IP en la lista. Para **transporte SCTP**, OmniDRA vincula todas las IP listadas a la asociación.

Parámetro	Tipo	Requerido	Predeterminado	Descripción
`listen_port`	Entero	Sí	-	Puerto local para conexiones Diameter entrantes. El puerto estándar es 3868 según RFC 6733.

Compatibilidad hacia atrás: Una única cadena IP sigue siendo totalmente compatible. Las configuraciones existentes de un solo hogar continúan funcionando sin cambios. Para optar por el multihoming, proporcione una lista.

Configuración del Par

Cada entrada de par declara la(s) dirección(es) remota(s) del par. El campo `ip` lleva la dirección principal y siempre es requerido. El campo opcional `additional_ips` enumera todas las direcciones para la asociación multihomed.

```
peers: [
  %{
    host: "hss01.example.com",
    realm: "example.com",
    ip: "192.168.1.20",                              # IP principal
(requerida)
    additional_ips: ["192.168.1.20", "10.0.0.20"],  # Todas las
IPs para el multihoming
    port: 3868,
    transport: :diameter_sctp,
    initiate_connection: true
  }
]
```

Parámetro	Tipo	Requerido	Predeterminado	De
`host`	Cadena	Sí	-	Ident Diam (FQD coinc exac con l confi par.
`realm`	Cadena	Sí	-	Reinc del p utiliz decis enru
`ip`	Cadena	Sí	-	Direc princ Siem requ inclu comp hacia confi de u
`additional_ips`	Lista de Cadenas	No	-	Toda direc para asoc mult Este opci omit utiliz Para

Parámetro	Tipo	Requerido	Predeterminado	De
				inclu princ lista. Omn este porq admi mult
`port`	Entero	Sí	-	Puer del p está 3868 6733
`transport`	Átomo	Sí	-	Proto trans la co par: `:dia` o `:dia` El mu requ `:dia`
`initiate_connection`	Booleano	Sí	-	Cuar Omn la co el pa es `fa` Omn a qu cone

Compatibilidad hacia atrás: El campo `ip` solo describe un par de un solo hogar. Si omite `additional_ips`, el par permanece de un solo hogar. Por lo tanto, las configuraciones existentes de pares no necesitan cambios.

Ejemplo Completo

La siguiente configuración vincula a OmniDRA a dos direcciones locales. Se conecta a un HSS multihomed y a un MME de un solo hogar.

```
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: [
      # Conexión 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 un solo hogar (compatible hacia atrás)
      %{
        host: "mme01.example.com",
        realm: "example.com",
        ip: "192.168.1.30",
        port: 3868,
        transport: :diameter_sctp,
        initiate_connection: false
      }
    ]
  }
```

Las tablas anteriores documentan `listen_ip`, `listen_port` y todos los parámetros de pares. Los parámetros restantes de nivel superior controlan la identidad Diameter de OmniDRA y el comportamiento de selección de pares. La Referencia de Configuración los documenta:

Parámetro	Tipo	Requerido	Predeterm
`service_name`	Átomo	Sí	-
`host`	Cadena	Sí	-
`realm`	Cadena	Sí	-
`product_name`	Cadena	Sí	-
`vendor_id`	Entero	Sí	-
`request_timeout`	Entero	No	`5000`

Parámetro	Tipo	Requerido	Predeterm
`peer_selection_algorithm`	Átomo	Sí	-
`allow_undefined_peers_to_connect`	Booleano	Sí	`false`
`peers`	Lista	Sí	-

Buffers de Socket SCTP

OmniDRA establece un buffer de recepción y envío de socket explícito en cada transporte SCTP. Utiliza la clave de nivel superior `diameter_socket_buffer_size` (predeterminado 4 MB). A diferencia de `listen_ip` y `peers`, esta clave se encuentra en el **nivel superior** de la configuración de la aplicación `:dra`. **No** está dentro del mapa `diameter:`.

Sin un buffer explícito, el socket SCTP hereda un pequeño `SO_RCVBUF` predeterminado. Esto limita la ventana de recepción SCTP a aproximadamente un MTU y causa bloqueos de ventana cero bajo carga. El núcleo de Linux limita `SO_RCVBUF` a `net.core.rmem_max`. Limita el buffer de envío a `net.core.wmem_max`. Para usar tamaños de buffer más grandes que ese límite, aumente esos sysctls en el host.

```
config :dra,
  # Clave de nivel superior - NO dentro del mapa diameter:.
Aplicada a cada transporte SCTP.
  diameter_socket_buffer_size: 4 * 1024 * 1024,  # 4 MB
(recbuf/sndbuf)
  diameter: %{
    # ... configuración del servicio y del par ...
  }
```

Parámetro	Tipo	Requerido	Predeterminado
`diameter_socket_buffer_size`	Entero	No	`4194304` (4 MB)

Requisitos

El multihoming SCTP necesita estas condiciones en ambos puntos finales:

- El **módulo del núcleo SCTP** debe estar cargado en el host (el paquete `lksctp-tools` en Linux).
- Todas las direcciones IP configuradas deben ser enrutables hacia y desde el par.
- Las reglas del firewall deben permitir tráfico SCTP en cada dirección IP configurada.
- Configure ambos puntos finales para el multihoming para lograr redundancia completa. Un OmniDRA multihomed conectado a un par de un solo hogar obtiene redundancia de ruta solo en el lado de OmniDRA.

Limitaciones

- **TCP no admite multihoming.** Con el transporte `:diameter_tcp`, OmniDRA utiliza solo la IP principal. `listen_ip` utiliza la primera IP en la lista, y OmniDRA ignora `additional_ips` del par.
- **El tiempo de conmutación por error de ruta** depende de los parámetros SCTP del núcleo, como el intervalo de latidos y los umbrales de retransmisión. Debe ajustar estos parámetros fuera de OmniDRA a nivel del sistema operativo.

Referencias

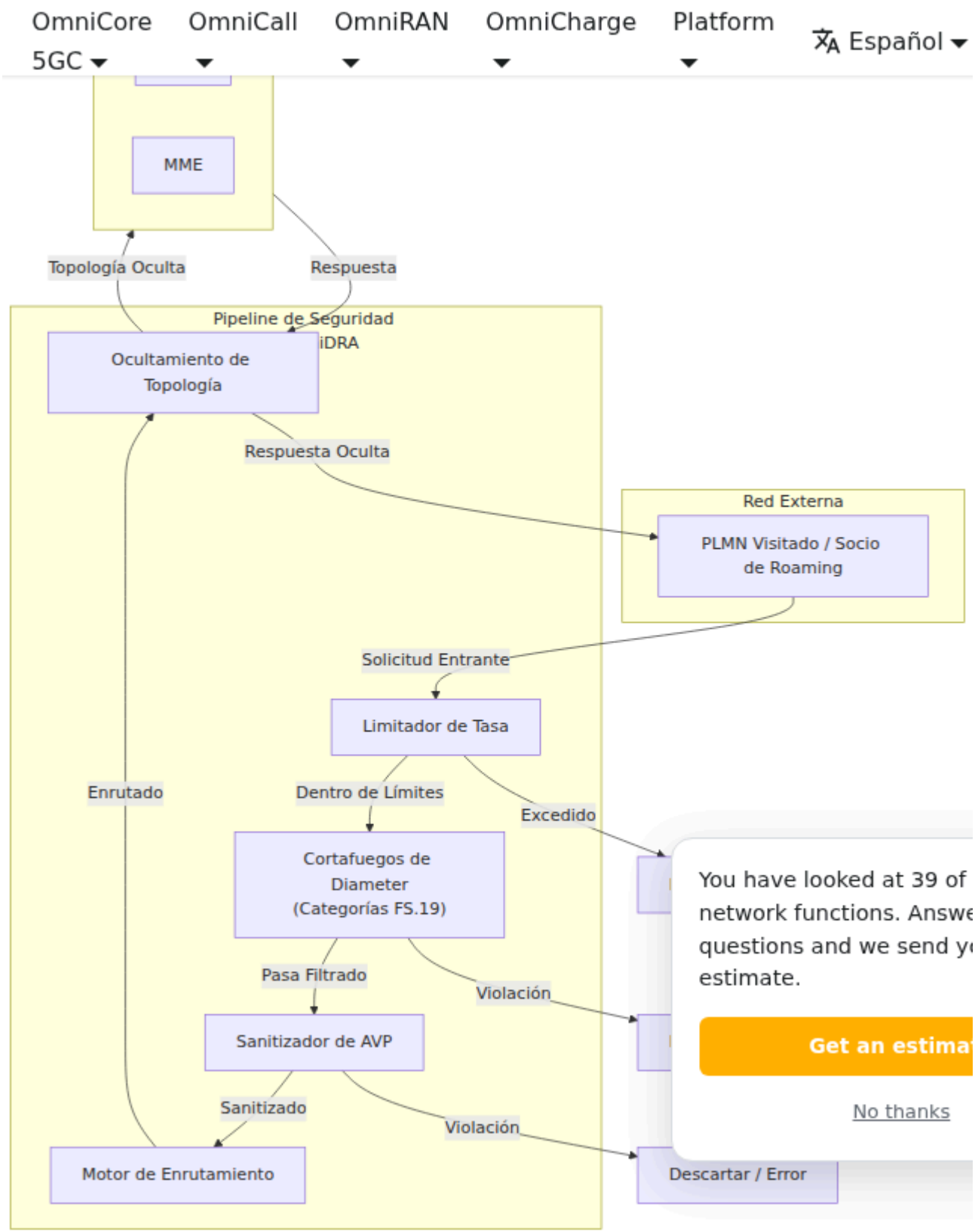
- RFC 4960 - Protocolo de Control de Transmisión de Flujos (SCTP)
- RFC 6733 - Protocolo Base Diameter
- Referencia de Configuración

Seguridad de Diameter

OmniDRA proporciona un conjunto completo de módulos de seguridad de Diameter. Se alinean con **GSMA FS.19** (Seguridad de Interconexión de Diameter, v10.0) y **GSMA FS.21** (Recomendaciones de Seguridad de Señalización de Interconexión, v12.0). Estos módulos protegen la red contra ataques a nivel de señalización en interfaces de interconexión de Diameter.

Puedes configurar cada módulo de forma independiente. Puedes habilitar o deshabilitar cada módulo sin afectar a los demás. Cada módulo emite sus propios eventos de telemetría para monitoreo y alertas.

Descripción General de la Arquitectura



Orden de Procesamiento

Las solicitudes de Diameter entrantes pasan a través de los módulos de seguridad en el siguiente orden. Si una etapa rechaza un mensaje, el DRA nunca lo reenvía a las etapas posteriores.

Orden	Módulo	Dirección	Propósito
1	Limitador de Tasa	Entrante	Protección contra inundaciones volumétricas
2	Cortafuegos de Diameter	Entrante	Filtrado de contenido y protocolo FS.19
3	Sanitizador de AVP	Entrante	Validación y eliminación de AVP
4	Motor de Enrutamiento	-	Enrutamiento estándar de Diameter
5	Ocultamiento de Topología	Saliente	Ocultamiento de la topología de la red

Acciones

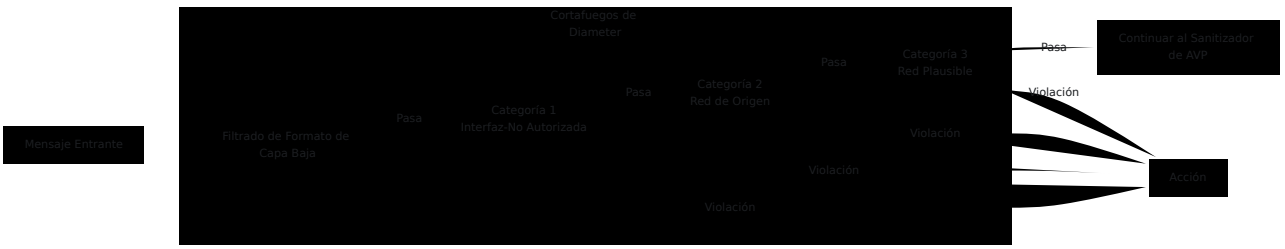
Todos los módulos de seguridad admiten acciones configurables cuando detectan una violación. Puedes configurar las acciones por módulo y por categoría para el Cortafuegos de Diameter.

Acción	Comportamiento	Resultado de Diameter
`{:error, 3002}`	Responder con una respuesta de error de Diameter	DIAMETER_UNABLE_TO_DELIVER
`{:error, 3004}`	Responder con una respuesta de error de Diameter	DIAMETER_TOO_BUSY
`{:error, 3007}`	Responder con una respuesta de error de Diameter	DIAMETER_APPLICATION_UNSUPPORTED
`:drop`	Descartar el mensaje	No se envía respuesta
`:log_only`	Registrar la violación pero permitir el paso del mensaje	El mensaje continúa

---

Cortafuegos de Diameter

El Cortafuegos de Diameter implementa las cuatro capas de filtrado definidas en **GSMA FS.19 Sección 3.3**. También implementa la categorización de paquetes independiente del protocolo de **FS.21 Sección 7**. El cortafuegos evalúa los mensajes a través de cada capa en orden. Una violación en cualquier capa detiene el procesamiento posterior.



Filtrado de Formato de Capa Baja

Referencia GSMA: FS.19 Sección 3.3.4, FS.21 Sección 7.3.1

El filtrado de capa baja detecta violaciones a nivel de protocolo e intentos básicos de suplantación. No necesita entender la semántica de la aplicación de capas superiores. Esta capa captura mensajes mal formados antes de que lleguen a una inspección más profunda.

Comprobaciones realizadas:

Comprobación	Referencia FS.19	Descripción
Duplicación de AVP	Sección 3.3.4, 4.8.1	Detecta instancias duplicadas de AVPs que deben aparecer como máximo una vez (por ejemplo, Origin-Host, Origin-Realm). Previene ataques de evasión de duplicación de AVP.
Orden de Session-Id	Sección 3.3.4	Confirma que Session-Id (AVP 263) es el primer AVP en el mensaje, según RFC 6733 Sección 8.8.
Destination-Host en Respuestas	Sección 3.3.4	Rechaza un mensaje de respuesta que contenga un AVP Destination-Host. Esta es una violación de protocolo que puede indicar evasión de filtro.
Validación de AVP Obligatorios	Sección 3.3.5	Confirma que los AVPs requeridos están presentes para cada código de comando. Por ejemplo, un ULR debe contener User-Name y Visited-PLMN-Id según 3GPP TS 29.272 Sección 7.2.3.

```
low_layer: %{
  enabled: true,
  action: {:error, 3002},
  # Códigos de AVP que no deben aparecer más de una vez (FS.19
Sección 3.3.4)
  # 264 = Origin-Host, 296 = Origin-Realm, 283 = Destination-
Realm, 293 = Destination-Host
  single_instance_avps: [264, 296, 283, 293],
  # Session-Id debe ser el primer AVP (RFC 6733 Sección 8.8)
  enforce_session_id_first: true,
  # Los mensajes de respuesta no deben contener Destination-Host
(FS.19 Sección 3.3.4)
  reject_destination_host_in_answers: true,
  # AVPs obligatorios 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]
  }
}
```

Categoría 1 - Filtrado de Paquetes de Interfaz-No Autorizada

Referencia GSMA: FS.19 Sección 3.3.5, FS.21 Sección 7.2.1

El filtrado de Categoría 1 acepta solo IDs de Aplicación de Diameter y Códigos de Comando autorizados en cada interfaz de interconexión. Esto previene el acceso externo a interfaces solo internas. Por ejemplo, bloquea comandos Sh a través de una interfaz S6a. También impone que un socio de roaming envíe solo los tipos de mensajes que cubre su acuerdo de roaming.

El enfoque de lista blanca sigue la recomendación de FS.19: **bloquear todos los mensajes de Diameter excepto aquellos que requiere una interfaz dada**.

Puedes configurar listas blancas a nivel global, lo que las aplica a todos los pares, o por par. Una lista blanca por par permite que diferentes socios de roaming tengan diferentes conjuntos de mensajes permitidos.

```
category_1: %{
  enabled: true,
  action: {:error, 3007},
  whitelists: %{
    # Lista blanca por par - restringir a este socio a ULR y AIR
solamente (FS.19 Sección 3.3.5)
    "restricted-partner.roaming.com" => %{
      16_777_251 => [316, 318]
    },
    # Lista blanca por defecto - comandos estándar de S6a
permitidos para todos los demás pares
    all: %{
      # S6a/S6d (FS.19 Sección 3.3.5, Tabla 2)
      16_777_251 => [316, 317, 318, 319, 320, 321, 323],
      # S13 - Verificación de Identidad de ME (FS.19 Sección
3.3.5)
      16_777_252 => [324],
      # S6c - SMS a través de HSS (FS.19 Sección 3.3.5.1)
      16_777_312 => [8388647, 8388648],
      # SGd - SMS a través de MME (FS.19 Sección 3.3.5.2)
      16_777_313 => [8388645, 8388646]
    }
  }
}
```

Lista Blanca Común de S6a/S6d:

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

IDs de Aplicación Comunes para interconexi��n de roaming (FS.19 Sección 3.3.5):

ID de Aplicación	Interfaz	Referencia
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

Categoría 2 - Filtrado de Paquetes de Red de Origen

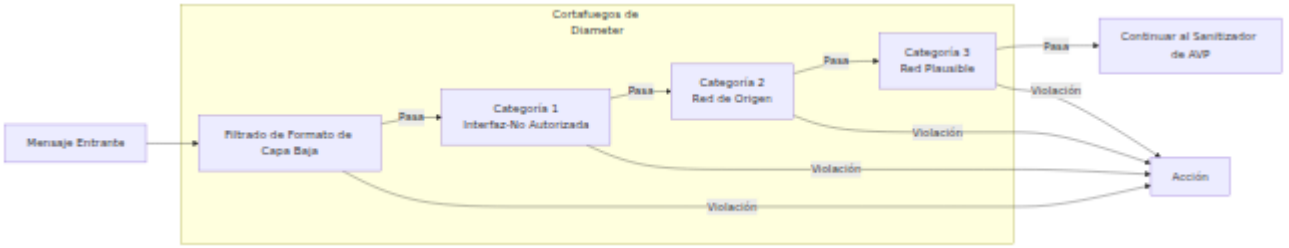
Referencia GSMA: FS.19 Sección 3.3.6, FS.21 Sección 7.2.2

El filtrado de Categoría 2 protege a los suscriptores de origen contra mensajes que los atacan desde la interconexión. El cortafuegos inspecciona mensajes en códigos de comando protegidos para la identidad del suscriptor (IMSI en el AVP User-Name, MSISDN en el AVP 701). Si la identidad coincide con un prefijo de suscriptor de origen, el cortafuegos rechaza el mensaje. El tráfico legítimo para los suscriptores de origen proviene de dentro de la red de origen, no de pares externos.

Esto previene ataques donde una entidad externa envía actualizaciones de ubicación, solicitudes de datos de suscriptores o consultas de autenticación que tienen como objetivo a los propios suscriptores del operador.

```
# Nivel superior: definir prefijos de suscriptor de origen
home_imsi_prefixes: ["31338", "31339"],
home_msisdn_prefixes: ["+1313"],

category_2: %{
  enabled: true,
  action: {:error, 3002},
  # Códigos de comando de S6a que llevan la identidad del
suscriptor (FS.19 Sección 3.3.6, Tabla 12)
  protected_command_codes: [316, 317, 318, 319, 320, 321, 323]
}
```



Categoría 3 - Filtrado de Paquetes de Red Plausible

Referencia GSMA: FS.19 Sección 3.3.7, FS.21 Sección 7.2.3

El filtrado de Categoría 3 detecta cambios de ubicación implausibles. Realiza un seguimiento de la última red vista para cada suscriptor (IMSI). Cuando llega un Update-Location-Request de una red visitada diferente a la anterior, el cortafuegos compara el tiempo transcurrido con un umbral configurable. Un cambio de dominio que ocurre más rápido de lo que es físicamente posible indica un posible ataque, como una actualización de ubicación falsificada.

Este módulo mantiene un seguimiento con estado por IMSI en una tabla ETS. Limpia automáticamente las entradas obsoletas. Elimina entradas más antiguas de 24 horas periódicamente.

Comprobaciones realizadas:

Comprobación	Referencia FS.19	Descripción
Comprobación de Ubicación Anterior	Sección 3.3.7.1	Compara el Origin-Realm del ULR actual con el último Origin-Realm visto para ese IMSI
Comprobación de Velocidad/Tiempo	Sección 3.3.7.2	Marca un cambio de dominio que ocurre dentro de un intervalo de tiempo mínimo configurable

```
category_3: %{
  enabled: true,
  # Comenzar con log_only para observar patrones antes de hacer
cumplir (FS.19 Sección 3.3.7)
  action: :log_only,
  # Velocidad máxima plausible en km/h (FS.19 Sección 3.3.7.2)
  max_velocity_kmh: 1200,
  # Segundos mínimos entre ULRs de diferentes dominios para el
mismo IMSI
  min_time_between_updates_seconds: 2
}
```

Configuración

Habilita el Cortafuegos de Diameter a nivel superior. Configura cada capa de filtrado de forma independiente. Combina los fragmentos de configuración de cada sección de categoría anterior bajo una única clave `module_diameter_firewall`:

```
config :dra,
  module_diameter_firewall: %{
    enabled: true,
    home_imsi_prefixes: ["31338", "31339"],
    home_msisdn_prefixes: ["+1313"],
    low_layer: %{ ... },      # Ver Filtrado de Formato de Capa
Baja arriba
    category_1: %{ ... },     # Ver Categoría 1 arriba
    category_2: %{ ... },     # Ver Categoría 2 arriba
    category_3: %{ ... }      # Ver Categoría 3 arriba
  }
```

Parámetros de Nivel Superior

Parámetro	Tipo	Requerido	Por Defecto	
`enabled`	Booleano	Sí	`false`	Habilitar o de Diameter. Cu inspección.
`children`	Lista	No	`[]`	Procesos que el módulo. Pa `[DRA.Router` El seguimient requiere.
`home_imsi_prefixes`	Lista	No	`[]`	Lista de cade suscriptores Ejemplo: `["3`
`home_msisdn_prefixes`	Lista	No	`[]`	Lista de cade suscriptores Ejemplo: `["+`

Parámetros de Capa Baja

Parámetro	Tipo	Requerido	Por Defecto
`enabled`	Booleano	No	`true`
`action`	Acción	No	`{:error, 3002}`
`single_instance_avps`	Lista	No	`[264, 296, 283, 293]`

Parámetro	Tipo	Requerido	Por Defecto
`enforce_session_id_first`	Booleano	No	`true`
`reject_destination_host_in_answers`	Booleano	No	`true`
`mandatory_avps`	Mapa	No	`%{}`

Parámetro	Tipo	Requerido	Por Defecto

Parámetros de Categoría 1

Parámetro	Tipo	Requerido	Por Defecto	Descripci
`enabled`	Booleano	No	`true`	Habilitar o deshabilitar e Categoría 1.
`action`	Acción	No	`{:error, 3007}`	Acción a tomar cuando detecta una violación de valor por defecto respor DIAMETER_APPLICATION
`whitelists`	Mapa	Sí	-	Mapa de nombre de hos `:all`) a combinaciones Aplicación / Código de C permitidas. La clave `:a` una lista blanca por def entradas específicas de prioridad.

Parámetros de Categoría 2

Parámetro	Tipo	Requerido	Por Defecto	Descrip
`enabled`	Booleano	No	`true`	Habilitar deshabili filtrado d Categoría
`action`	Acción	No	`{:error, 3002}`	Acción a tomar cu un mensa tiene con objetivo a suscripto origen de la intercone
`protected_command_codes`	Lista	No	`[]`	Códigos d comando activan verificaci de identi de suscripto de origen Típicame los códig de coma de S6a q llevan la identidad suscripto

Parámetros de Categoría 3

Parámetro	Tipo	Requerido	Por Defecto
`enabled`	Booleano	No	`false`
`action`	Acción	No	`:log_only`
`max_velocity_kmh`	Entero	No	`1200`

Parámetro	Tipo	Requerido	Por Defecto
`min_time_between_updates_seconds`	Entero	No	`2`

Eventos de Telemetría

Evento	Descripción
`[:diameter, :firewall, :low_layer, :block]`	El cortafuegos detectó y bloqueó una violación de formato de capa baja
`[:diameter, :firewall, :category_1, :block]`	El cortafuegos detectó y bloqueó una violación de Categoría 1
`[:diameter, :firewall, :category_2, :block]`	El cortafuegos detectó y bloqueó una violación de Categoría 2
`[:diameter, :firewall, :category_3, :block]`	El cortafuegos detectó y bloqueó una violación de Categoría 3
`[:diameter, :firewall, :pass, :count]`	El mensaje pasó todas las verificaciones del cortafuegos

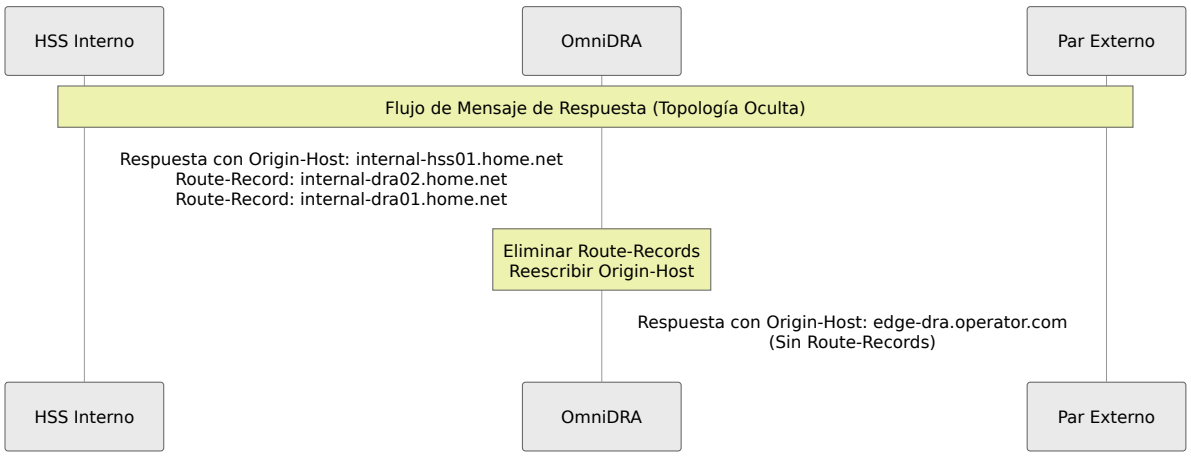
Todos los eventos incluyen estos metadatos: `origin_host`, `application_id`, `command_code`, `application_name`, `command_name`, y `reason`.

Ocultamiento de Topología

Referencia GSMA: FS.19 Sección 2.4, Sección 3.4; FS.21 Sección 3.6

El módulo de Ocultamiento de Topología previene la exposición de la topología interna de la red a pares externos. Cuando los mensajes de Diameter atraviesan múltiples nodos internos, acumulan AVPs Route-Record. También llevan valores de Origin-Host/Origin-Realm que revelan nombres de host internos, estructura de red y conteos de nodos. Un atacante puede usar esta información para mapear la red interna para ataques dirigidos.

El ocultamiento de topología opera en la **ruta saliente**. Se ejecuta después de que el DRA toma decisiones de enrutamiento y antes de que reenvíe mensajes al par de destino.



Características

Característica	Referencia FS.19	Descripción
Eliminación de Route-Record	Sección 2.4	Elimina todos los AVPs Route-Record (282) que revelan rutas de enrutamiento internas y nombres de host de nodos
Reescritura de Origin-Host	Sección 3.4	Reemplaza el AVP Origin-Host con la propia identidad del DRA (o un valor personalizado) en mensajes de respuesta
Reescritura de Origin-Realm	Sección 3.4	Opcionalmente reemplaza el AVP Origin-Realm para ocultar la estructura interna del dominio
Control por Par	-	Aplicar el ocultamiento de topología de manera selectiva, solo a pares externos o a todos los pares

```
module_topology_hiding: %{
  enabled: true,
  # Eliminar AVPs Route-Record que revelan rutas internas (FS.19
Sección 2.4)
  strip_route_records: true,
  # Reescribir Origin-Host en respuestas para ocultar nombres de
nodos internos (FS.19 Sección 3.4)
  rewrite_origin_host: %{enabled: true, replacement: :self},
  # Opcionalmente ocultar la estructura del dominio interno
  rewrite_origin_realm: %{enabled: false, replacement: :self},
  # Aplicar a todos los pares externos, o listar nombres de host
específicos
  external_peers: :all
}
```

Parámetros

Parámetro	Tipo	Requerido	Por Defecto	Descripci
`enabled`	Booleano	Sí	`false`	Habilitar o deshabilitar e módulo de Ocultamiento Topología.
`strip_route_records`	Booleano	No	`true`	Eliminar todo AVPs Route-Record (códig 282) de los mensajes ant de que el DR reenvíe a par externos.
`rewrite_origin_host`	Mapa	No	Ver abajo	Controla la reescritura d Origin-Host e mensajes de respuesta.
`rewrite_origin_realm`	Mapa	No	Ver abajo	Controla la reescritura d Origin-Realm mensajes de respuesta.
`external_peers`	`:all` o Lista	No	`:all`	Pares que el módulo trata como externo módulo aplic ocultamiento

Parámetro	Tipo	Requerido	Por Defecto	Descripci
				topología sol los mensajes destinados a estos pares. `:all` para implementac de interconex Usa una lista nombres de h para aplicaci selectiva.

Parámetros de Reescritura (aplican a `rewrite_origin_host` y `rewrite_origin_realm`):

Parámetro	Tipo	Requerido	Por Defecto	Descripción
`enabled`	Booleano	No	varía	Habilitar reescritura para este AVP. El valor por defecto es `true` para Origin-Host, `false` para Origin-Realm.
`replacement`	`:self` o String	No	`:self`	Valor de reemplazo. `:self` usa la propia identidad del DRA del config de `diameter` (`host.realm`). El módulo usa un valor de cadena tal cual.

Eventos de Telemetría

Evento	Descripción
`[:diameter, :topology_hiding, :route_record, :stripped]`	El módulo eliminó AVPs Route-Record. La medición incluye el `count` de AVPs eliminados.
`[:diameter, :topology_hiding, :origin_host, :rewritten]`	El módulo reescribió el AVP Origin-Host
`[:diameter, :topology_hiding, :origin_realm, :rewritten]`	El módulo reescribió el AVP Origin-Realm

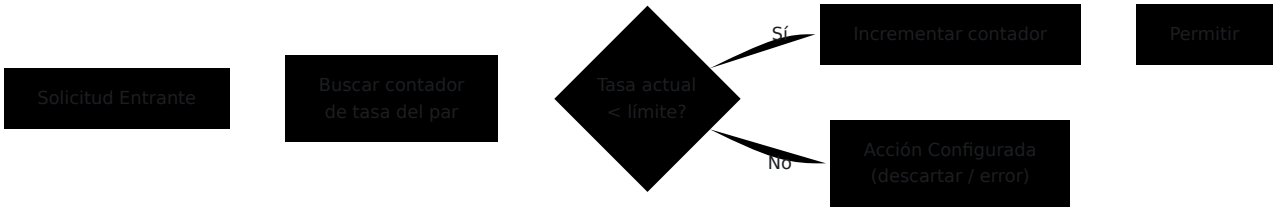
Limitador de Tasa

Referencia GSMA: FS.19 Sección 3.4 (Disponibilidad / Protección contra DoS)

El Limitador de Tasa aplica límites de tasa de mensajes por par. Protege contra ataques volumétricos e inundaciones de mensajes. Opera como la primera verificación de seguridad en el pipeline, antes de cualquier análisis de mensaje o inspección de contenido. Esto reduce la carga excesiva lo antes posible.

El Limitador de Tasa rastrea los límites de tasa por par con un contador de ventana deslizante. Cada par tiene un contador independiente que se restablece cada segundo.

```
module_rate_limiter: %{
  enabled: true,
  # Límite por defecto para todos los pares (FS.19 Sección 3.4)
  default_max_requests_per_second: 1000,
  default_action: {:error, 3004},
  # Sobrescrituras 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}
  }
}
```



Parámetros

Parámetro	Tipo	Requerido	Por Defecto
`enabled`	Booleano	Sí	`false`
`children`	Lista	No	`[]`
`default_max_requests_per_second`	Entero	No	`1000`
`default_action`	Acción	No	`{:error, 3004}`
`peer_limits`	Mapa	No	`%{}`

Parámetros de Sobrescritura por Par:

Parámetro	Tipo	Requerido	Por Defecto	Descripción
`max_requests_per_second`	Entero	No	Hereda el por defecto	Máximo de solicitudes por segundo para este par específico.
`action`	Acción	No	Hereda el por defecto	Acción cuando este par excede su límite de tasa.

Eventos de Telemetría

Evento	Descripción
`[:diameter, :rate_limiter, :throttled]`	El módulo limitó la tasa de un mensaje. Las mediciones incluyen `current_rate` y `limit`.
`[:diameter, :rate_limiter, :allowed]`	Un mensaje estaba dentro de los límites de tasa.

Sanitizador de AVP

Referencia GSMA: FS.19 Sección 3.3.4, 4.8.1, 4.8.2; FS.21 Sección 3.6

El Sanitizador de AVP valida y sanitiza los AVPs de Diameter en la frontera de interconexión. Aborda las recomendaciones de la Sección 3.6 de FS.21. Estas

recomendaciones manejan ataques de manipulación a nivel de protocolo que operan por debajo del nivel de las categorías de filtrado de FS.19.

Características

Característica	Referencia GSMA	Descripción
Eliminación de AVP de Vendedor Desconocido	FS.21 Sección 3.6	Elimina AVPs de vendedores no incluidos en la lista blanca. Previene la inyección de AVPs propietarios que pueden desencadenar comportamientos no deseados en nodos de backend.
Profundidad de Anidación de AVP Agrupados	FS.21 Sección 3.6	Impone una profundidad máxima de anidación para AVPs agrupados. Previene ataques de desbordamiento de pila que utilizan estructuras de AVP profundamente anidadas.

```
module_avp_sanitizer: %{
  enabled: true,
  action: {:error, 3002},
  # Solo permitir AVPs de vendedor estándar en la interconexión
(FS.21 Sección 3.6)
  # 0 = IETF, 10415 = 3GPP, 13019 = ETSI, 5535 = 3GPP2
  allowed_vendor_ids: [0, 10415, 13019, 5535],
  strip_unknown_vendor_avps: true,
  # Prevenir desbordamiento de pila a través de AVPs agrupados
profundamente anidados (FS.21 Sección 3.6)
  max_avp_nesting_depth: 10
}
```

Parámetros

Parámetro	Tipo	Requerido	Por Defecto	De
`enabled`	Booleano	Sí	`false`	Habilit deshab módulo de AVP
`action`	Acción	No	`{:error, 3002}`	Acción cuando detecta de prof anidac elimina vendec los AVP no bloc mensa
`allowed_vendor_ids`	Lista	No	`[0, 10415, 13019, 5535]`	Lista d vendec en la ir El mód AVPs d vendec
`strip_unknown_vendor_avps`	Booleano	No	`true`	Habilit elimina de ven no esta de `allowe`
`max_avp_nesting_depth`	Entero	No	`10`	Profun permit

Parámetro	Tipo	Requerido	Por Defecto	De
				anidac agrupa configu a un m excede profun

IDs de Vendedor Comunes:

ID de Vendedor	Organización	Notas
0	IETF	AVPs estándar de Diameter definidos en RFCs
10415	3GPP	Todos los AVPs definidos por 3GPP (S6a, Gx, Rx, etc.)
13019	ETSI	AVPs definidos por ETSI
5535	3GPP2	AVPs definidos por 3GPP2 (interoperabilidad de CDMA)

Eventos de Telemetría

Evento	Descripción
`[:diameter, :avp_sanitizer, :unknown_vendor, :stripped]`	El módulo eliminó uno o más AVPs de vendedores desconocidos
`[:diameter, :avp_sanitizer, :nesting_depth, :violation]`	Un mensaje excedió la profundidad máxima de anidación de AVP
`[:diameter, :avp_sanitizer, :pass, :count]`	El mensaje pasó todas las verificaciones de sanitización

---

Recomendaciones de Implementación

Orden de Habilitación Recomendado

Cuando implementes los módulos de seguridad por primera vez, habilítalos uno a la vez. Esto evita interrupciones en el tráfico en vivo:

1. **Limitador de Tasa** - Comienza con límites generosos y monitorea los patrones de tráfico. Ajusta los límites después de entender las tasas base.
2. **Sanitizador de AVP** - Bajo riesgo de falsos positivos. Habilita las verificaciones de eliminación y anidación.
3. **Cortafuegos de Diameter (Categoría 1)** - Define listas blancas basadas en acuerdos de roaming. Comienza con combinaciones de ID de Aplicación / Código de Comando conocidas y buenas.
4. **Cortafuegos de Diameter (Capa Baja)** - Habilita las verificaciones de conformidad de protocolo.
5. **Cortafuegos de Diameter (Categoría 2)** - Configura los prefijos IMSI/MSISDN de origen.

6. **Ocultamiento de Topología** - Habilita primero la eliminación de Route-Record, luego la reescritura de Origin-Host.
7. **Cortafuegos de Diameter (Categoría 3)** - Habilítalo con la acción `:log_only`. Observa los patrones de ubicación antes de hacer cumplir.

Defensa en Profundidad

Estos módulos implementan el principio de **defensa en profundidad** de FS.19 Sección 3.4. Cada capa aborda una clase diferente de ataque:

Clase de Ataque	Defensa Primaria	Defensa Secundaria
DoS Volumétrico	Limitador de Tasa	Cortafuegos de Diameter (todas las categorías)
Abuso de Interfaz	Categoría 1	Sanitizador de AVP
Objetivo de Suscriptor de Origen	Categoría 2	Categoría 1 (restricción de interfaz)
Suplantación de Ubicación	Categoría 3	Categoría 2 (verificación de suscriptor de origen)
Descubrimiento de Topología	Ocultamiento de Topología	-
Inyección de AVP	Sanitizador de AVP	Filtrado de Capa Baja
Evasión de Protocolo (Duplicación de AVP)	Filtrado de Capa Baja	Sanitizador de AVP

Referencia Cruzada de Documentos GSMA

Módulo / Característica	FS.19 v10.0	FS.21 v12.0
Filtrado de Formato de Capa Baja	Sección 3.3.4	Sección 7.3.1
Filtrado de Categoría 1	Sección 3.3.5, Anexo B.3.3	Sección 7.2.1
Filtrado de Categoría 2	Sección 3.3.6, Anexo B.3.4	Sección 7.2.2, Sección 16
Filtrado de Categoría 3	Sección 3.3.7, Anexo B.3.5	Sección 7.2.3
Ocultamiento de Topología	Sección 2.4, Sección 3.4	Sección 3.6
Limitación de Tasa	Sección 3.4	-
Sanitización de AVP	Sección 4.8.1, 4.8.2	Sección 3.6
Defensa en Profundidad	Sección 3.4	Sección 3.15
Categorías de Filtrado (Independientes del Protocolo)	Anexo A	Sección 7

Enrutamiento de Diámetro Estándar

Enrutamiento de Diámetro Estándar es el comportamiento predeterminado de reenvío de solicitudes del DRA. El DRA lo aplica cuando ninguna regla de Enrutamiento Avanzado coincide con una solicitud entrante. Sigue el mecanismo de enrutamiento basado en prioridades definido en el Protocolo Base de Diámetro (RFC 6733). Selecciona un destino de los contenidos de AVP y el conjunto de pares conectados del DRA.

Cuándo se Aplica el Enrutamiento Estándar

El DRA siempre intenta el enrutamiento estándar como su comportamiento base. Si el módulo de Enrutamiento Avanzado está habilitado y una de sus reglas coincide con la solicitud, esa regla decide el resultado en su lugar. La regla puede enrutar a pares nombrados, enrutar por `Destination-Host`, generar una respuesta de error o descartar la solicitud. Cuando el Enrutamiento Avanzado está deshabilitado, o ninguna regla coincide, la solicitud pasa al orden de precedencia de enrutamiento estándar descrito aquí.

Una excepción estrecha se aplica después de ambas etapas. Cuando está habilitado, Rx Session Binding redirige una solicitud Rx al PCRF que mantiene su sesión. Solo anula una decisión de retransmisión. Un resultado `:drop`, una respuesta de error y un `Destination-Host` explícito se mantienen.

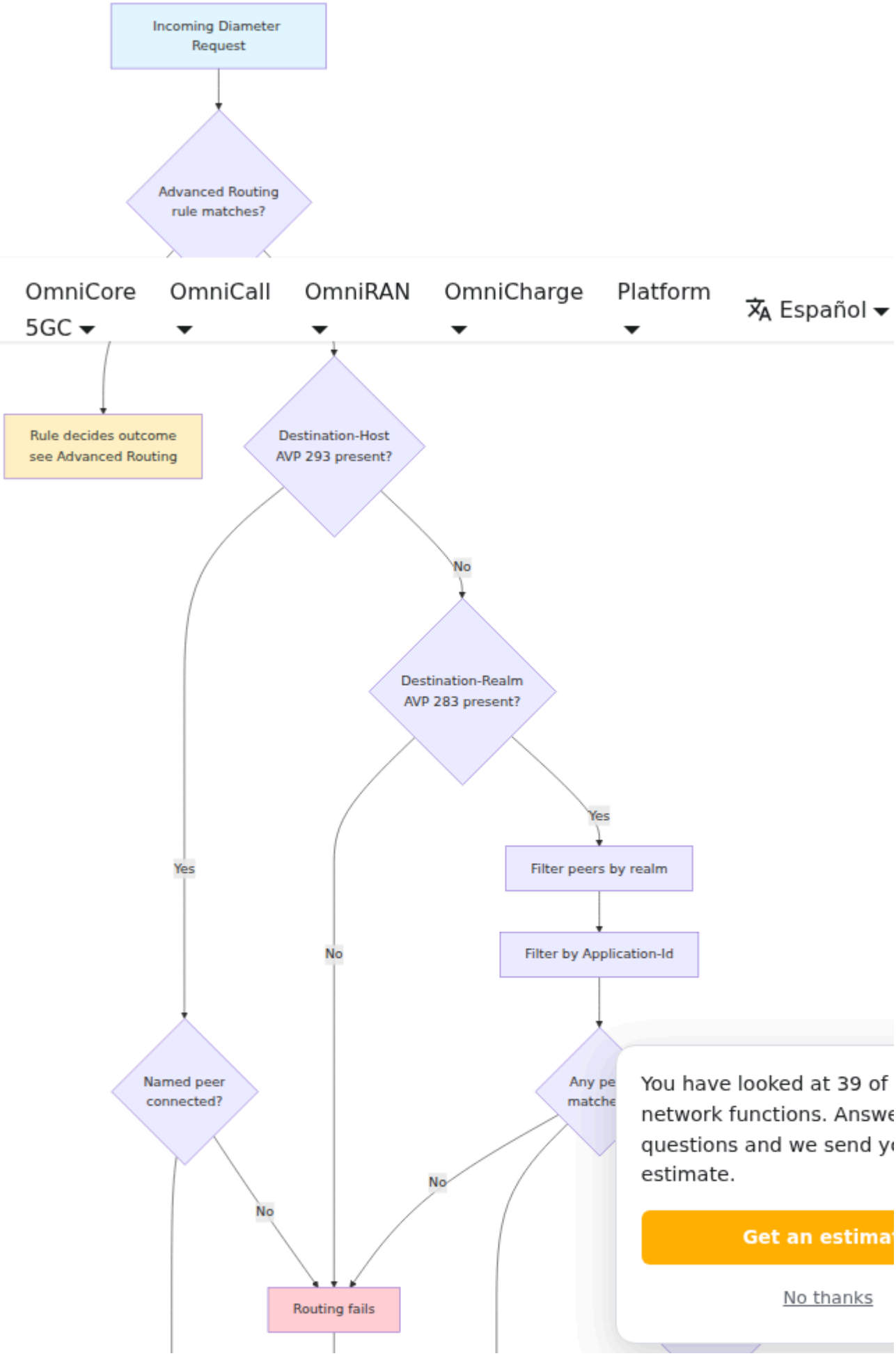
Enrutamiento Avanzado describe las anulaciones basadas en contenido. Estas anulan el enrutamiento en valores de AVP como patrones de IMSI, traducción de dominio y decisiones de pares de origen. Esa página también detalla **afinidad de sesión** a través del AVP `Destination-Host`.

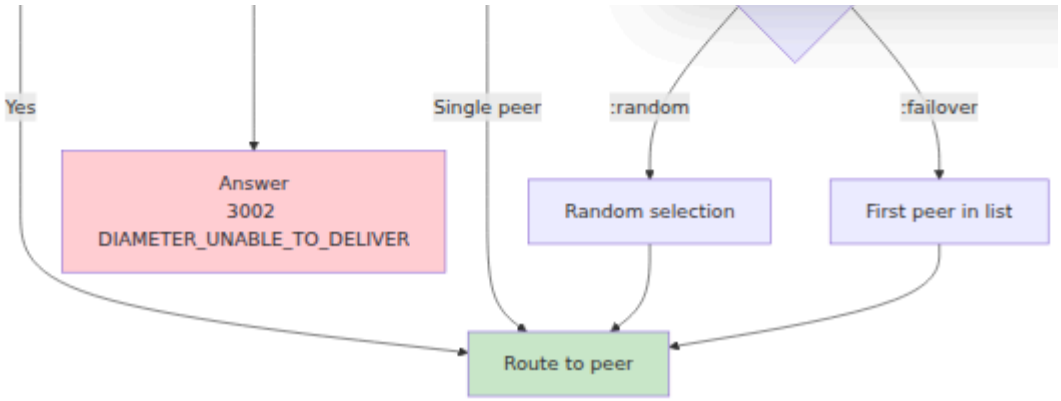
Precedencia de Decisión de Enrutamiento

El enrutamiento estándar evalúa cada solicitud contra un orden de precedencia fijo. El primer mecanismo aplicable gana.

1. **`Destination-Host` AVP (293)** - Este es el mecanismo de mayor prioridad. Si la solicitud lleva un AVP `Destination-Host`, el DRA enruta directamente a ese host. Esto proporciona control de enrutamiento explícito a nivel de host. Si el par nombrado no está conectado, el enrutamiento falla.
2. **`Destination-Realm` AVP (283) + `Application-Id`** - Si el `Destination-Host` está ausente, el DRA selecciona un par conectado que anuncia soporte para el **dominio** objetivo. El DRA luego filtra los pares candidatos por el **Application-Id** del mensaje. Solo los pares que anuncian soporte para esa aplicación permanecen. Los pares anuncian este soporte durante el Intercambio de Capacidades. Cuando más de un par coincide, el algoritmo de selección de pares configurado elige entre ellos.
3. **Fallo de retransmisión / enrutamiento** - Si no hay un `Destination-Host` alcanzable ni un par coincidente de dominio más aplicación disponible, el DRA no puede entregar la solicitud. El DRA devuelve una respuesta `3002 DIAMETER_UNABLE_TO_DELIVER`.

El siguiente diagrama de flujo muestra el camino completo de decisión. Incluye el punto en el que el Enrutamiento Avanzado puede interceptar la solicitud antes de que se ejecute el enrutamiento estándar.


Incoming Diameter Request
Advanced Routing rule matches?
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
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?
No
Any pe
matche
No
Routing fails
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Solo los pares en estado **conectado** son elegibles en cualquier etapa. El DRA excluye automáticamente a los pares desconectados o caídos del filtrado de dominio y aplicación. Una solicitud `Destination-Host` dirigida a un par desconectado falla inmediatamente.

Selección de Pares

Cuando múltiples pares conectados coinciden con los criterios de dominio y Application-Id, la configuración global `peer_selection_algorithm` determina qué par recibe la solicitud. Esta es una configuración a nivel de servicio. No se configura por dominio o por par.

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

Parámetro	Tipo	Requerido	Predeterminado	D
`peer_selection_algorithm`	Átomo	No	`:random`	Es de de pa m pa co co co sc `:` se ur al co co `:` si se pr er co Ut pa re sc el no di

- **`:random`** - Distribuye la carga de manera uniforme entre todos los pares coincidentes. Utiliza esto cuando los pares son intercambiables y de costo igual.
- **`:failover`** - Prefiere un único par primario. Trata la lista de pares ordenados como una secuencia de prioridad. Utiliza esto para diseños

activos/en espera. Un par específico lleva tráfico mientras esté conectado.

Consulta Configuración para el bloque de configuración completo del servicio y definiciones de pares.

Cuándo No Hay Pares Disponibles

La evaluación de precedencia puede fallar en identificar un destino alcanzable. Esto sucede cuando el par `Destination-Host` está desconectado, cuando ningún par conectado anuncia el dominio objetivo, cuando ningún par coincidente admite el Application-Id, o cuando la solicitud carece de los AVPs necesarios para tomar una decisión. En cada uno de estos casos, el DRA no puede reenviar la solicitud.

El DRA luego devuelve una respuesta con `Result-Code` establecido en `3002 DIAMETER_UNABLE_TO_DELIVER`, como se define en RFC 6733 §7.1.5. Esto le dice al par de origen que el DRA recibió la solicitud pero no pudo entregarla a un siguiente salto adecuado. El originador puede entonces aplicar su propia lógica de reintento o failover.

Referencia

Códigos de Resultado

Código	Nombre	Descripción	Referencia
2001	DIAMETER_SUCCESS	El DRA completó la solicitud con éxito	RFC 6733 §7.1.1
3002	DIAMETER_UNABLE_TO_DELIVER	El DRA no pudo entregar la solicitud a un siguiente salto adecuado. No había ningún par coincidente y conectado disponible	RFC 6733 §7.1.3

Códigos AVP Comunes

Código	Nombre AVP	Tipo	Uso	Referencia
1	User-Name	UTF8String	Identificador del suscriptor (IMSI en 3GPP)	RFC 6733 §8.14
264	Origin-Host	DiameterIdentity	Nombre de host del par de origen	RFC 6733 §6.3
268	Result-Code	Unsigned32	Código de resultado estándar	RFC 6733 §7.1
283	Destination-Realm	DiameterIdentity	Dominio objetivo	RFC 6733 §6.6
293	Destination-Host	DiameterIdentity	Host objetivo (opcional)	RFC 6733 §6.5
296	Origin-Realm	DiameterIdentity	Dominio de origen	RFC 6733 §6.4
297	Experimental-Result	Grouped	Código de resultado específico del proveedor	RFC 6733 §7.6

Códigos de Comando Comunes

Los códigos de comando son parte del encabezado del mensaje de Diámetro, no de los AVPs.

Código	Nombre del Comando	Descripción	Referencia
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 Aplicación Comunes de 3GPP

Los pares anuncian sus Application-Ids soportados durante el Intercambio de Capacidades. El enrutamiento estándar filtra los pares candidatos por el

Application-Id de la solicitud.

Application-Id	Interfaz	Descripción	Referencia
16777251	S6a/S6d	Autenticación y datos de suscripción MME/SGSN ↔ HSS	3GPP TS 29.272
16777252	S13/S13'	Comprobación de identidad de equipo MME ↔ EIR	3GPP TS 29.272
16777238	Gx	Control de políticas y carga PCEF ↔ PCRF	3GPP TS 29.212
16777267	S9	Política de roaming PCRF en casa ↔ PCRF visitado	3GPP TS 29.215
16777272	S6b	PDN-GW ↔ 3GPP AAA para acceso no 3GPP	3GPP TS 29.273
16777216	Cx	I-CSCF/S-CSCF ↔ HSS registro IMS	3GPP TS 29.229
16777217	Sh	AS ↔ HSS datos de usuario IMS	3GPP TS 29.329
16777255	SLg	Servicios de localización MME/SGSN ↔ GMLC	3GPP TS 29.172
16777291	SLh	Información de suscriptor de localización GMLC ↔ HSS	3GPP TS 29.173
16777302	Sy	Vinculación de sesión PCRF ↔ OCS	3GPP TS 29.219
16777310	S6m	MTC-IWF ↔ HSS/HLR para dispositivos M2M	3GPP TS 29.336

Application-Id	Interfaz	Descripción	Referencia
16777312	S6c	Enrutamiento SMS-SC/IP-SM-GW ↔ HSS	3GPP TS 29.338
16777345	S6t	SCEF ↔ HSS monitoreo de eventos	3GPP TS 29.336
16777236	Rx	AF ↔ autorización de medios PCRF	3GPP TS 29.214

Las aplicaciones de carga (Gy/Ro) siguen 3GPP TS 32.299.

Documentación Relacionada

- Configuración - Referencia completa de configuración del servicio y de pares.
- Enrutamiento Avanzado - Reglas de enrutamiento basadas en contenido, afinidad de sesión a través de `Destination-Host`, y anulaciones impulsadas por reglas que tienen prioridad sobre el enrutamiento estándar.

Función de Búsqueda de Suscriptores (SLF)

La Función de Búsqueda de Suscriptores (SLF) aprende dinámicamente qué elemento de red atiende a cada suscriptor. Observa el tráfico de señalización Diameter que pasa a través del DRA. Utiliza los enlaces aprendidos para enrutar solicitudes posteriores, como consultas de servicios de localización, directamente al nodo de servicio correcto. No necesita reglas de enrutamiento estáticas.

Esto es útil para solicitudes de servicios de localización SLg/SLh. El GMLC necesita alcanzar el MME de servicio para un suscriptor dado, pero no sabe cuál es.

3GPP TS 29.172 y 3GPP TS 29.173 describen el concepto de SLF para servicios de localización. 3GPP TS 29.272 define el registro de suscriptores.

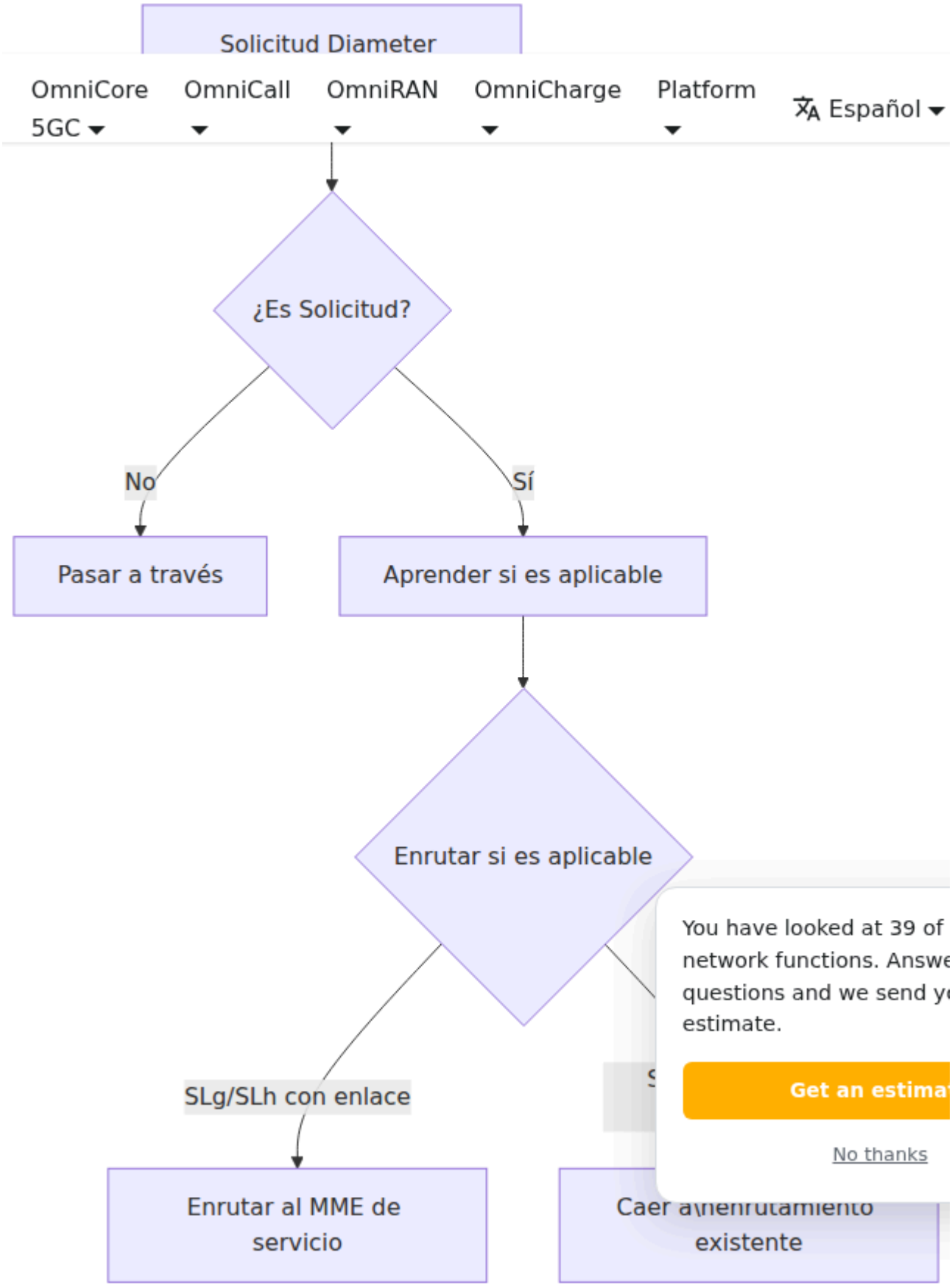
Cómo Funciona

El módulo opera en dos fases: **aprendizaje** y **enrutamiento**.

Durante el **aprendizaje**, el módulo observa los mensajes de registro y desregistro que fluyen a través del DRA. Cuando un suscriptor se registra (por ejemplo, a través de una solicitud de actualización de ubicación S6a), el módulo registra qué elemento de red ahora atiende a ese suscriptor.

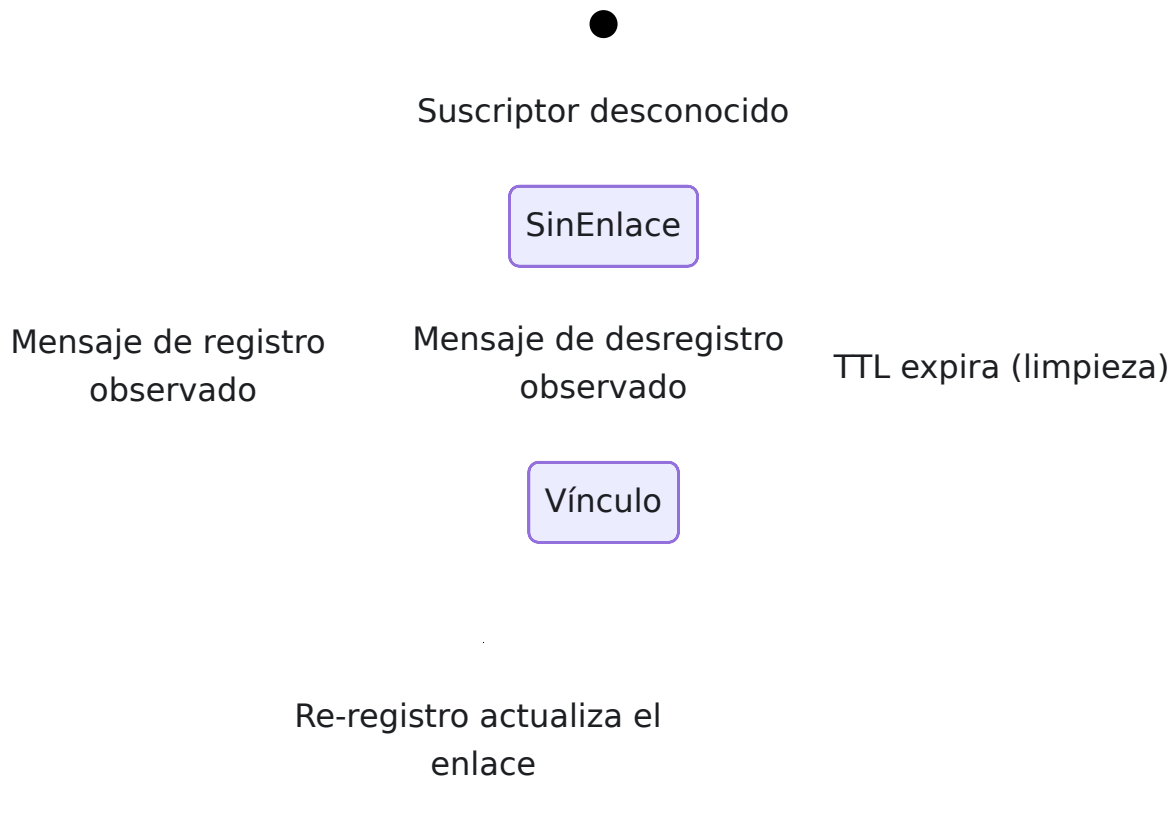
Durante el **enrutamiento**, puede llegar una solicitud que debe alcanzar el nodo de servicio de un suscriptor (por ejemplo, una solicitud de proporcionar ubicación SLg). El módulo busca el enlace y enruta directamente al par correcto.

Si no existe un enlace para un suscriptor, la solicitud cae en la lógica de enrutamiento existente, incluyendo el Enrutamiento Avanzado.



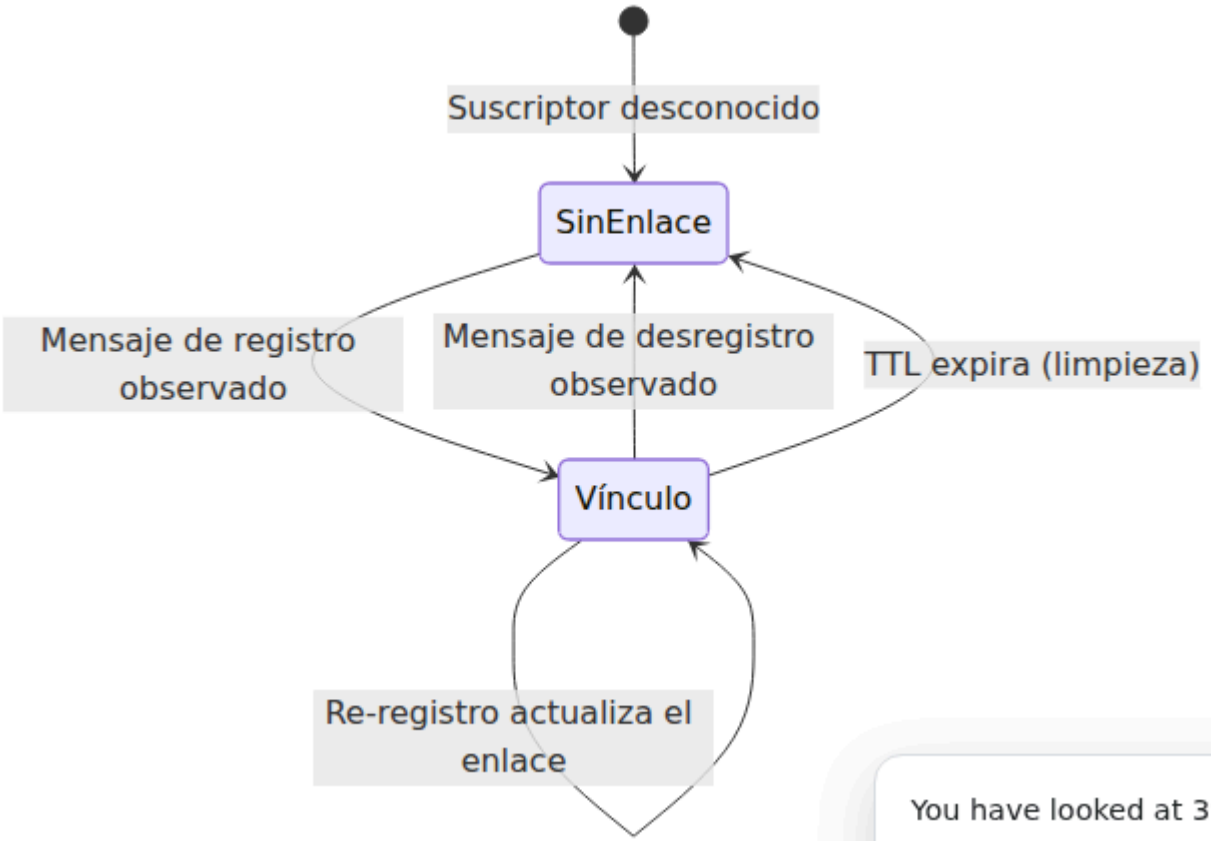
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

Ciclo de Vida del Enlace



Aprendizaje: Mensajes que Crean Enlaces

El módulo aprende enlaces de suscriptor a nodo de servicio a partir de los siguientes mensajes Diameter:



Interfaz	Mensaje	Código de Comando	Enlace Creado	Fuente de IMSI
S6a	Solicitud de Actualización de Ubicación (ULR)	316	`serving_mme`	AVP de Nombre de Usuario (1)
Gx	Solicitud de Control de Crédito Inicial (CCR-I)	272 (CC-Request-Type=1)	`serving_pgw`	AVP de Identificación de Suscripción (443)
Cx	Solicitud de Asignación de Servidor (SAR)	301 (Server-Assignment-Type=1)	`serving_cscf`	AVP de Identidad Pública (601)

Aprendizaje: Mensajes que Remueven Enlaces

Interfaz	Mensaje	Código de Comando	Enlace Removido
S6a	Solicitud de Cancelación de Ubicación (CLR)	317	`serving_mme`
S6a	Solicitud de Purga de UE (PUR)	321	`serving_mme`
Gx	Solicitud de Control de Crédito de Terminación (CCR-T)	272 (CC-Request-Type=3)	`serving_pgw`
Cx	Solicitud de Asignación de Servidor (SAR)	301 (Server-Assignment-Type=4)	`serving_cscf`

Cuando el módulo elimina el último enlace para un suscriptor, elimina todo el registro del suscriptor de la tabla.

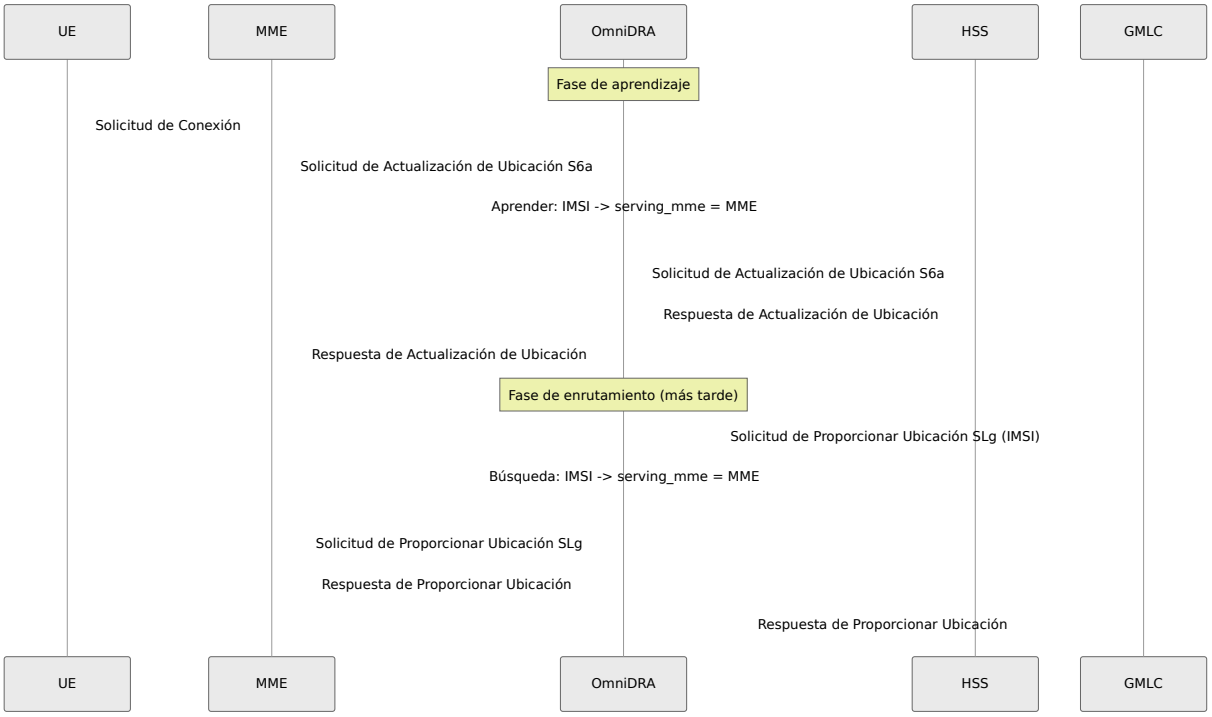
Enrutamiento: Mensajes que Usan Enlaces

Interfaz	Mensaje	Código de Comando	Enlace Usado
SLg	Solicitud de Proporcionar Ubicación (PLR)	8388620	`serving_mme`
SLh	Solicitud de Información de Enrutamiento LCS (RIR)	8388622	`serving_mme`

El DRA retransmite una solicitud enrutada por un enlace aprendido al nodo de servicio. Utiliza el tiempo de espera base de retransmisión de la configuración

`diameter.request_timeout` (ver Conectividad de Pares o Configuración). El módulo no define su propio tiempo de espera para estas solicitudes enrutadas.

Flujo de Ejemplo: Solicitud de Servicio de Localización



Posición en la Canalización

La búsqueda de suscriptores se ejecuta **antes** del Enrutamiento Avanzado en la canalización de procesamiento de solicitudes. Si el SLF encuentra un enlace y sobrescribe la ruta, el Enrutamiento Avanzado aún se ejecuta. La ruta del SLF tiene prioridad.



Configuración

Configura el módulo bajo `module_subscriber_lookup` en `config/runtime.exs`.

```
module_subscriber_lookup: %{
  # Habilitar o deshabilitar el módulo
  enabled: false,

  # Cuánto tiempo mantener los enlaces antes de expirar (segundos)
  binding_ttl_seconds: 86400
}
```

Parámetros

Parámetro	Tipo	Requerido	Predeterminado	
`enabled`	Booleano	Sí	`false`	Habi Búsq `fals` modi enla
`children`	Lista	No	`[]`	Proce cuan `[DRA` para se ej
`binding_ttl_seconds`	Entero	No	`86400`	Tiem enla limpi 24 h mita 60 se

Ejemplos de Configuración

Implementación Estándar

Usa esto para la mayoría de las redes donde los suscriptores se re-registran dentro de 24 horas.

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

Cómo funciona: El módulo aprende enlaces MME/PGW/CSCF del tráfico que pasa y los mantiene durante 24 horas. El DRA enruta automáticamente las solicitudes SLg y SLh al MME de servicio correcto. El módulo refresca un enlace cada vez que observa un nuevo mensaje de registro para el mismo suscriptor.

Caso de uso: Redes con requisitos de servicio de localización (LCS). El GMLC debe alcanzar el MME de servicio sin asignaciones estáticas de suscriptor a MME.

Entorno de Alta Rotación

Usa esto para redes con movilidad frecuente de suscriptores. Los enlaces obsoletos deben expirar rápidamente.

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

Cómo funciona: Los enlaces expiran después de 1 hora. Usa esto cuando los suscriptores se mueven a menudo entre MMEs. Los enlaces obsoletos causarían solicitudes de ubicación mal dirigidas. El intervalo de limpieza se ejecuta cada 30 minutos.

Caso de uso: Redes urbanas densas o eventos con alta movilidad de suscriptores.

Métricas

Actualizaciones de Enlaces

Métrica: `diameter.subscriber_lookup.binding.update` **Tipo**: Contador **Descripción**: El DRA incrementa este contador cada vez que crea o actualiza un enlace de suscriptor. **Etiquetas**:

- `imsi` - IMSI del suscriptor
- `binding_type` - Tipo de enlace: `serving_mme`, `serving_pgw`, o `serving_cscf`
- `serving_host` - Origin-Host del elemento de red de servicio

Eliminaciones de Enlaces

Métrica: `diameter.subscriber_lookup.binding.delete` **Tipo**: Contador **Descripción**: El DRA incrementa este contador cada vez que un mensaje de desregistro elimina un enlace de suscriptor. **Etiquetas**:

- `imsi` - IMSI del suscriptor
- `binding_type` - Tipo de enlace eliminado

Solicitudes Enrutadas

Métrica: `diameter.subscriber_lookup.route.count` **Tipo**: Contador **Descripción**: El DRA incrementa este contador cada vez que enruta una solicitud SLg/SLh con un enlace aprendido. **Etiquetas**:

- `imsi` - IMSI del suscriptor
- `binding_type` - Tipo de enlace que el DRA utilizó para el enrutamiento (actualmente siempre `serving_mme`)
- `serving_host` - Origin-Host del par al que el DRA enruto la solicitud

Solución de Problemas

Solicitudes de Ubicación No Enrutadas al MME de Servicio

Síntomas: El DRA no enruta solicitudes de Proporcionar Ubicación SLg o solicitudes de Información de Enrutamiento LCS SLh al MME esperado. Caen en el enrutamiento predeterminado en su lugar.

Causas posibles:

- El suscriptor no se ha registrado a través de este DRA, por lo que no existe un enlace.
- El enlace ha expirado. Se ha superado el TTL.
- El par MME de servicio no está conectado a este DRA.
- `enabled` está configurado como `false`.

Resolución:

1. Confirma que el módulo está habilitado en la configuración.
2. Verifica que el tráfico ULR S6a para el suscriptor fluya a través de este DRA. El módulo aprende enlaces solo del tráfico observado.
3. Confirma que `binding_ttl_seconds` es lo suficientemente largo para cubrir el intervalo entre el registro y la solicitud de ubicación.
4. Confirma que el MME de servicio está conectado como un par.

Enlaces No Aprendidos

Síntomas: La tabla de enlaces permanece vacía a pesar de que el tráfico S6a/Gx/Cx pasa a través del DRA.

Causas posibles:

- El módulo no está habilitado.
- El proceso del módulo no está en ejecución. Verifica el supervisor.
- Los mensajes no contienen un IMSI válido en el AVP esperado.

Resolución:

1. Confirma `enabled: true` en la configuración.
2. Confirma que el DRA se reinició después del cambio de configuración.
3. Verifica los registros de depuración del DRA para entradas de `SubscriberLookup: Learned`. Estas confirman la actividad de enlace.

Enlaces Obsoletos Causando Solicitudes Mal Enrutadas

Síntomas: El DRA enruta solicitudes de ubicación a un MME que ya no atiende al suscriptor.

Causas posibles:

- La Solicitud de Cancelación de Ubicación (CLR) o la Solicitud de Purga de UE (PUR) no pasaron a través de este DRA.
- `binding_ttl_seconds` está configurado demasiado alto para el patrón de movilidad de la red.

Resolución:

1. Reduce `binding_ttl_seconds` para que coincida con el intervalo de re-registro esperado del suscriptor.
2. Confirma que todo el tráfico de desregistro S6a fluya a través de este DRA.

Referencia

IDs de Aplicación

ID	Interfaz	Descripción	Referencia
16777251	S6a/S6d	Autenticación y gestión de suscripciones de MME/SGSN a HSS	3GPP TS 29.272
16777238	Gx	Control de políticas y cobro de PCEF a PCRF	3GPP TS 29.212
16777216	Cx	Registro IMS de I-CSCF/S-CSCF a HSS	3GPP TS 29.229
16777255	SLg	Servicios de localización de GMLC a MME	3GPP TS 29.172
16777291	SLh	Información de enrutamiento LCS de GMLC a HSS/DRA	3GPP TS 29.173

Códigos de Comando

Código	Nombre	Interfaz	Descripción
272	Solicitud/Respuesta de Control de Crédito (CCR/CCA)	Gx	Control de políticas y cobro a nivel de sesión
301	Solicitud/Respuesta de Asignación de Servidor (SAR/SAA)	Cx	Registro y desregistro IMS
316	Solicitud/Respuesta de Actualización de Ubicación (ULR/ULA)	S6a	Actualización de ubicación del suscriptor en MME
317	Solicitud/Respuesta de Cancelación de Ubicación (CLR/CLA)	S6a	Cancelación de ubicación iniciada por HSS
321	Solicitud/Respuesta de Purga de UE (PUR/PUA)	S6a	Purga de UE iniciada por MME
8388620	Solicitud/Respuesta de Proporcionar Ubicación (PLR/PLA)	SLg	Solicitud de servicio de ubicación al MME de servicio
8388622	Solicitud/Respuesta de Información de Enrutamiento LCS (RIR/RIA)	SLh	Búsqueda de información de enrutamiento LCS

Tipos de Enlace

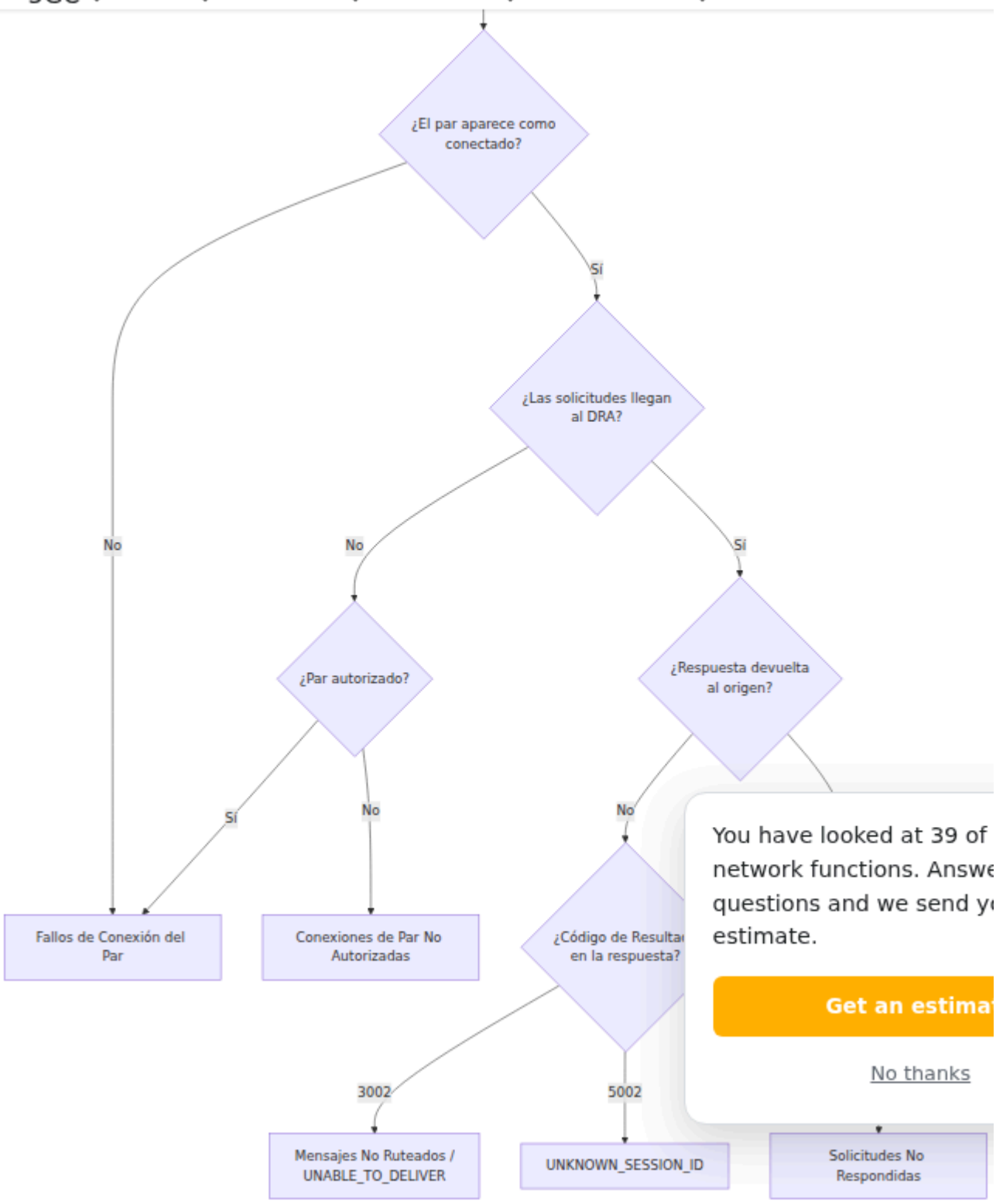
Tipo de Enlace	Aprendido de	Usado por	Descripción
`serving_mme`	S6a ULR	SLg PLR, SLh RIR	El MME que actualmente atiende al suscriptor
`serving_pgw`	Gx CCR-I	-	El PGW que maneja la sesión del suscriptor (reservado para enrutamiento futuro)
`serving_cscf`	Cx SAR (Registro)	-	El S-CSCF que atiende el registro IMS del suscriptor (reservado para enrutamiento futuro)

Solución de Problemas

Esta guía cubre los problemas más comunes en la operación de OmniDRA. Cada entrada enumera los síntomas que un operador observa, las causas probables y un conjunto ordenado de pasos. Utilice los pasos para aislar y resolver el problema. Para los nombres de las métricas en esta guía, consulte Métricas.

Árbol de Decisión de Diagnóstico

Utilice este flujo para encontrar dónde se encuentra un problema antes de ir a una sección específica.



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

Reglas de Ruteo

Regla No Coincidente

Síntomas: Una regla de ruteo avanzada configurada nunca se activa. El tráfico cae en el ruteo estándar o en una regla posterior en su lugar.

Causas posibles:

- Una o más condiciones de filtro no coinciden con el contenido real del mensaje.
- El código AVP en el filtro no coincide con el AVP utilizado por la aplicación Diameter en cuestión.
- Un patrón de expresión regular es incorrecto o está sobre-/sub-anclado.
- El tipo de mensaje está fuera del alcance de coincidencia de la regla. Por ejemplo, un filtro solo para solicitudes se aplica a una respuesta.
- La regla utiliza el tipo de filtro incorrecto para el módulo.

Resolución:

1. Verifique cada condición de filtro contra un mensaje capturado. Todas las condiciones en una regla deben coincidir para que la regla se active.
2. Confirme que los códigos AVP coincidan con la aplicación Diameter. Consulte la referencia de código AVP en Ruteo Avanzado.
3. Pruebe los patrones de expresión regular contra valores AVP reales antes de implementarlos.
4. Confirme que el tipo de mensaje esté dentro del alcance de coincidencia de la regla.
5. Revise los tipos de filtro disponibles en Ruteo Avanzado. Confirme que esté utilizando el correcto para el módulo.

Ruteo Inesperado

Síntomas: El DRA reenvía solicitudes, pero a un par diferente al previsto. Alternativamente, el DRA utiliza ruteo estándar cuando esperaba que se aplicara una regla.

Causas posibles:

- Orden de las reglas. El ruteo avanzado utiliza el primer acierto, por lo que una regla anterior y más amplia puede eclipsar una posterior y más específica.
- El nombre del par previsto es incorrecto o el par no es accesible.
- Dos reglas tienen filtros superpuestos, y la incorrecta gana.
- Ninguna regla coincidió, por lo que Ruteo Estándar tomó el control.

Resolución:

1. Revise el orden de las reglas. Coloque las reglas más específicas antes que las más amplias. La primera regla que coincida gana. Consulte Ruteo Avanzado.
2. Confirme que los nombres de los pares coincidan exactamente con la configuración y que el par objetivo esté conectado (`diameter_peer_status` informa `1`).
3. Verifique si hay reglas en conflicto con filtros superpuestos. Reordénelas o ajustelas.
4. Confirme que el comportamiento de caída en Ruteo Estándar coincida con lo que observa cuando no se aplica ninguna regla.

Transformación No Aplicada

Síntomas: Una regla de ruteo coincide, pero una transformación de agregar/editar/eliminar AVP no tiene efecto en el mensaje.

Causas posibles:

- La transformación apunta a un código AVP que es incorrecto para el caso de uso.
- Una acción de `editar` se ejecutó en un AVP que no existe en el mensaje. `editar` modifica un AVP existente pero nunca crea uno.
- Los filtros de la regla no coinciden con el flujo de mensaje previsto, por lo que el mensaje nunca llega a la transformación.
- El filtro de tipo de paquete (`:request` vs `:answer`) excluye el mensaje.
- La acción no admite el tipo de paquete. `editar` se aplica solo a solicitudes.

Resolución:

1. Confirme que los códigos AVP son correctos para su caso de uso. Consulte la referencia de código AVP en Ruteo Avanzado.
2. Para una acción de `editar`, confirme que el AVP ya existe en el mensaje. Utilice `agregar` si el AVP puede estar ausente.
3. Confirme que los filtros de la regla coincidan con el flujo de mensaje previsto.
4. Confirme que cualquier filtro de tipo de paquete (`:request` vs `:answer`) sea correcto para la dirección que transforma.
5. Confirme que la acción admite el tipo de paquete. `editar` funciona solo en solicitudes.

Conectividad del Par

Fallos de Conexión del Par

Síntomas: Un par configurado nunca alcanza el estado conectado. `diameter_peer_status{origin_host="..."}` permanece en `0`, y el ruteo a ese par falla.

Causas posibles:

- Un firewall bloquea el `listen_port` configurado (por defecto `3868`) o el puerto del par.
- La configuración del par tiene un `ip`, `port` o `transport` incorrecto.
- Transporte no coincidente. Un lado está configurado para `:diameter_tcp` y el otro para `:diameter_sctp`.
- El Origin-Host publicitado por el par no coincide con el `host` configurado. Las identidades deben coincidir exactamente.
- `initiate_connection` está configurado de la misma manera en ambos extremos, por lo que ninguno de los lados marca, o ambos marcan.
- El par no publicita una aplicación soportada durante el Intercambio de Capacidades (CER/CEA).

- Para un par saliente, el servicio del par no está en ejecución o no está escuchando.

Resolución:

1. Confirme que las reglas del firewall permitan tráfico en el `listen_port` y el puerto del par para el transporte configurado. SCTP requiere el módulo SCTP del kernel y reglas de firewall que permitan SCTP, no solo TCP.
2. Confirme que los valores de `ip`, `port` y `transport` del par coincidan exactamente con el punto final real del par.
3. Confirme que ambos extremos utilicen el mismo protocolo de transporte.
4. Confirme que el Origin-Host publicitado por el par coincida con el `host` configurado. Las identidades Diameter deben coincidir exactamente para el ruteo.
5. Verifique los roles de `initiate_connection`. Exactamente un lado debe iniciar. Establezca `initiate_connection: true` en el lado que marca (típicamente el DRA hacia un HSS o PCRF). Establezca `false` en el lado que espera (típicamente hacia un MME, SGSN o P-GW).
6. Confirme que el par publicite al menos una aplicación que el DRA sirva durante CER/CEA. Un par sin aplicación común no es utilizable para ruteo.
7. Para un par saliente, confirme que el servicio remoto esté en ejecución y escuchando. El DRA reintenta automáticamente los pares inalcanzables.

Fluctuación del Par / Tráfico Enviado Demasiado Pronto Después de Reconexión

Síntomas: Después de que un par se reconecta, el DRA despacha solicitudes a él, y estas fallan o se agotan inmediatamente mientras el par aún se estabiliza. Alternativamente, un par que se recuperó recientemente parece utilizable pero descarta tráfico.

Causas posibles:

- El par es inestable en la capa de red y se reconecta repetidamente (fluctuando).
- El tiempo de asentamiento a nivel de aplicación es demasiado corto para este par, por lo que el DRA rutea tráfico antes de que el par esté listo.

- Un tiempo de asentamiento excesivamente largo retrasa la recuperación legítima y extiende la ventana de interrupción.

Resolución:

1. Investigue primero la inestabilidad del transporte subyacente (ruta de red, conmutación por error de ruta SCTP, o reinicios del lado del par). Un par que fluctúa es un síntoma, no un problema de configuración de ruteo.

2. Ajuste el tiempo de asentamiento para que un par recién conectado deba permanecer activo durante un intervalo definido antes de que el DRA rutee tráfico a él. Un par que se desconecta dentro de esta ventana nunca entra en el conjunto ruteable. Esto mantiene el tráfico fuera de los pares fluctuantes.

```
config :dra, :gateway,
  # Mantenga un par recién conectado este número de ms antes de
enrutar tráfico a él.
  # 0 = enrutar inmediatamente una vez que CER/CEA se complete.
  peer_settle_time_ms: 2_000,

  # Anulación opcional (ms) para el temporizador de vigilancia
Tw de RFC 3539.
  # watchdog_timer_ms: 30_000
```

Parámetro	Tipo	Requerido	Predeterminado	D
`peer_settle_time_ms`	Entero	No	`0`	Mili que reci con peri acti que rute él. inm des CER com par des dur ven nun com Esto el t de l fluc
`watchdog_timer_ms`	Entero	No	`30000`	Anu opc tem de RFC inte acti Disp Wat (DW Déj

Parámetro	Tipo	Requerido	Predeterminado	D
				con mer ten esp cam cad acti

3. `diameter_peer_status` refleja la ruteabilidad real. Un par dentro de su ventana de asentamiento aún informa como no ruteable en métricas. Utilice esto para confirmar que el comportamiento de asentamiento funciona como espera.

4. Aumente `peer_settle_time_ms` para pares que fluctúan crónicamente para suprimir el despacho prematuro. Mantenlo bajo (o `0`) para pares estables para minimizar el tiempo de recuperación.

> El DRA promueve a un par reconectado a ruteable inmediatamente después de que el tiempo de asentamiento a nivel de aplicación transcurre. No espera el retraso implícito de reconexión del watchdog de RFC 3539. El tiempo de asentamiento anterior es el único control explícito sobre cuánto tiempo el DRA mantiene un par en recuperación.

Conexiones de Par No Autorizadas

Síntomas: El DRA rechaza la conexión de un par. Alternativamente, `diameter_peer_unauthorized_connection_count_total` se incrementa con un `origin_host` / `peer_ip` inesperado.

Causas posibles:

- El par que se conecta no está en la configuración de `peers`, y `allow_undefined_peers_to_connect` es `false`.
- El Origin-Host publicitado por el par no coincide con ningún `host` configurado.

- Un nodo genuinamente no autorizado o mal configurado intenta conectarse.

Resolución:

1. Confirme si se debe permitir el par. Si es así, agréguelo a la lista de `peers` con su `host`, `realm`, `ip`, `port` y `transport` exactos.

2. Confirme que el Origin-Host publicitado por el par coincida exactamente con el `host` configurado. El DRA trata una discrepancia como un par desconocido.

3. Mantenga `allow_undefined_peers_to_connect: false` en producción y agregue pares explícitamente. Habilítelo solo en entornos controlados donde cualquier par pueda conectarse.

4. Mantenga `log_unauthorized_peer_connection_attempts: true` para que el DRA registre intentos rechazados para revisión de seguridad.

5. Investigue las etiquetas `peer_ip` y `origin_host` reclamadas en `diameter_peer_unauthorized_connection_count_total`. Una tasa sostenida de una fuente desconocida justifica una investigación de seguridad.

Parámetro	Tipo	Requerido
`allow_undefined_peers_to_connect`	Booleano	No
`log_unauthorized_peer_connection_attempts`	Booleano	No

Problemas de Asociación SCTP

Síntomas: Un par SCTP no logra establecer una asociación. Alternativamente, una asociación establecida se cae o falla inesperadamente. Los pares TCP en el mismo host se conectan normalmente.

Causas posibles:

- El módulo SCTP del kernel no está cargado en el host.
- Las reglas del firewall permiten TCP pero no SCTP en las direcciones configuradas.
- Una o más direcciones en el conjunto de multihoming no son ruteables desde el par.
- La IP principal no está en la lista de `additional_ips` del par.

- Solo un punto final está configurado para multihoming, por lo que el DRA no logra redundancia completa.
- Los parámetros SCTP del kernel rigen el tiempo de conmutación por error de la ruta, y difieren de las expectativas.

Resolución:

1. Confirme que el módulo del kernel SCTP esté cargado (el paquete `lksctp-tools` en Linux).
2. Confirme que las reglas del firewall permitan tráfico SCTP en cada IP configurada, no solo TCP.
3. Confirme que todas las direcciones IP en el conjunto de multihoming sean ruteables en ambas direcciones entre el DRA y el par.
4. Para un par SCTP, incluya la IP principal en la lista de `additional_ips`. Para pares TCP, el DRA utiliza solo `ip`, porque TCP no admite multihoming.
5. Configure ambos extremos para multihoming para lograr redundancia completa de la ruta.
6. Si el tiempo de conmutación por error no es el esperado, revise los parámetros SCTP del kernel. El tiempo de conmutación por error de la ruta depende de ellos, no de la configuración del DRA.
7. Consulte Multihoming SCTP para la configuración completa del transporte y sus limitaciones.

Entrega de Mensajes

Mensajes No Ruteados / 3002 UNABLE_TO_DELIVER

Síntomas: Las solicitudes llegan al DRA, pero el DRA les responde con el Código de Resultado `3002` (DIAMETER_UNABLE_TO_DELIVER). Alternativamente, el DRA no las reenvía al destino esperado en absoluto.

Causas posibles:

- Ningún par conectado coincide con el Destination-Host o Destination-Realm de la solicitud.
- El par objetivo está desconectado, o está dentro de su ventana de asentamiento y aún no es ruteable.
- El par objetivo no publicita soporte para la aplicación en la solicitud.
- Una regla de ruteo avanzada seleccionó un par que no es accesible.
- Ningún par sirve el realm solicitado. Esto produce `3003` (DIAMETER_REALM_NOT_SERVED), no `3002`.

Resolución:

1. Confirme que al menos un par que coincida con el Destination-Host o Destination-Realm esté conectado. `diameter_peer_status` para ese par debe leer `1`.

2. Verifique si el par objetivo está dentro de su ventana de asentamiento. Un par que se ha reconectado recientemente no es deliberadamente ruteable hasta que transcurre `peer_settle_time_ms`. Consulte Fluctuación del Par.

3. Confirme que el par publicite la aplicación en la solicitud durante CER/CEA. El DRA excluye a los pares que no publicitan la aplicación de la selección.

4. Si una regla de ruteo avanzada apunta a un par específico, confirme que ese par esté conectado. El DRA omite a los pares desconectados. Consulte Ruteo Avanzado.

5. Distinga `3002` de `3003`. `3002` significa que el DRA no pudo seleccionar un par accesible. `3003` significa que ningún par sirve el realm solicitado. Corrija la configuración del realm si observa `3003`.

6. Realice un seguimiento de la tasa de respuestas `3002` para detectar fallos sistémicos de entrega:

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

Solicitudes No Respondidas (Tiempos de Espera)

Síntomas: El DRA reenvía solicitudes a un par, pero no se devuelve ninguna respuesta dentro del tiempo de espera. `diameter_peer_unanswered_request_count_total` se incrementa, y `diameter_peer_last_response_delay` aumenta.

Causas posibles:

- El par recibió la solicitud pero no respondió dentro de `request_timeout` (por defecto `5000` ms).
- El par está sobrecargado o lento.
- Un camino de red está degradado. Esto es relevante para el multihoming SCTP durante la conmutación por error.
- El par procesó la solicitud, pero la respuesta se perdió en tránsito.

Resolución:

1. Identifique los pares lentos o silenciosos:

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

2. Compare con `diameter_peer_last_response_delay` para el mismo par. Esto muestra si es lento en lugar de silencioso.

3. Revise la carga y salud del par. Los tiempos de espera sostenidos generalmente indican un problema en la parte inferior, no un problema del DRA.

4. Para pares SCTP, verifique si hay eventos de conmutación por error de ruta que coincidan con los tiempos de espera. Consulte Multihoming SCTP.

5. Ajuste `request_timeout` solo si el tiempo de procesamiento legítimo del par excede el valor predeterminado. Un valor más alto también extiende el TTL de Métricas Extendidas (`request_timeout` + 5 segundos).

Parámetro	Tipo	Requerido	Predeterminado	Desc
`request_timeout`	Entero	No	`5000`	Milisegu esperar respuest de que e trate el solicitud como ag

UNKNOWN_SESSION_ID en Gx / Gy con Estado

Síntomas: Las sesiones con estado (control de políticas Gx, carga en línea Gy) fallan con el Código de Resultado `5002` (DIAMETER_UNKNOWN_SESSION_ID). Típicamente, el par rechaza una actualización (CCR-U) o terminación (CCR-T) aunque la inicial (CCR-I) haya tenido éxito.

Causas posibles:

- El DRA enruta una solicitud a mitad de sesión a un par diferente al que mantiene el estado de la sesión.
- El par que mantiene la sesión se reinició y perdió su tabla de sesiones.
- El DRA enruta a un par que se reconectó antes de restaurar su estado. Esta es una interacción de ventana de asentamiento.
- El balanceo de carga distribuyó solicitudes para un Session-Id entre múltiples pares.

Resolución:

1. Confirme que todas las solicitudes con el mismo Session-Id lleguen al mismo par con estado. Para Gx y Gy, la afinidad de sesión es esencial. Los mensajes a mitad de sesión deben aterrizar en el par que respondió CCR-I. Prefiera el enrutamiento basado en `Destination-Host`, o una regla avanzada que asigne una familia de Session-Id a un par, sobre el balanceo de carga basado en realm. Consulte Ruteo Estándar y Ruteo Avanzado.

2. Revise el `peer_selection_algorithm`. `:random` distribuye entre pares coincidentes y puede romper la afinidad de sesión para aplicaciones con estado. `:failover` envía al primer par en la lista. Para interfaces con estado, rutee en `Destination-Host`. La afinidad es entonces explícita y no depende del balanceo de carga.

3. Verifique si el par objetivo se reinició alrededor del momento en que comenzaron las respuestas `5002`. Un par que perdió su tabla de sesiones rechaza sesiones en vuelo. El cliente debe restablecer esas sesiones.

4. Si un par se reconectó recientemente, confirme que `peer_settle_time_ms` sea lo suficientemente grande como para que el DRA rutee tráfico solo después de que el par esté listo. Consulte Fluctuación del Par.

5. Realice un seguimiento de la tasa de respuestas `5002` para correlacionar con eventos de pares:

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

Referencia: Códigos de Resultado

Comunes

Código	Nombre	Significado	Referencia
2001	DIAMETER_SUCCESS	El DRA completó la solicitud con éxito.	RFC 6733 §7.1.1
3002	DIAMETER_UNABLE_TO_DELIVER	El DRA no pudo seleccionar un par accesible para entregar la solicitud.	RFC 6733 §7.1.3
3003	DIAMETER_REALM_NOT_SERVED	Ningún par sirve el realm solicitado.	RFC 6733 §7.1.3
3004	DIAMETER_TOO_BUSY	El par seleccionado está sobrecargado.	RFC 6733 §7.1.3
5002	DIAMETER_UNKNOWN_SESSION_ID	El par no reconoce el Session-Id. Esta es una sesión con estado perdida o mal ruteada.	RFC 6733 §7.1.5

Código	Nombre	Significado	Referencia
5012	DIAMETER_UNABLE_TO_COMPLY	El DRA no puede procesar la solicitud tal como se presenta.	RFC 6733 §7.1.5

Documentación Relacionada

- Métricas - Métricas de Prometheus y consultas de ejemplo mencionadas anteriormente.
- Ruteo Estándar - Ruteo prioritario RFC 6733, selección de pares y ruteo de respuestas.
- Ruteo Avanzado - Ruteo basado en reglas, filtros, transformaciones y orientación de pares.
- Multihoming SCTP - Transporte SCTP, multihoming y conmutación por error de ruta.