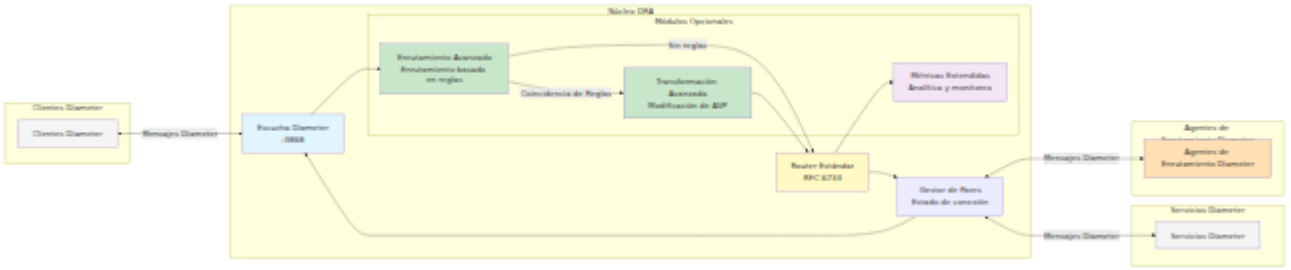
Guía de Operaciones DRA

Tabla de Contenidos

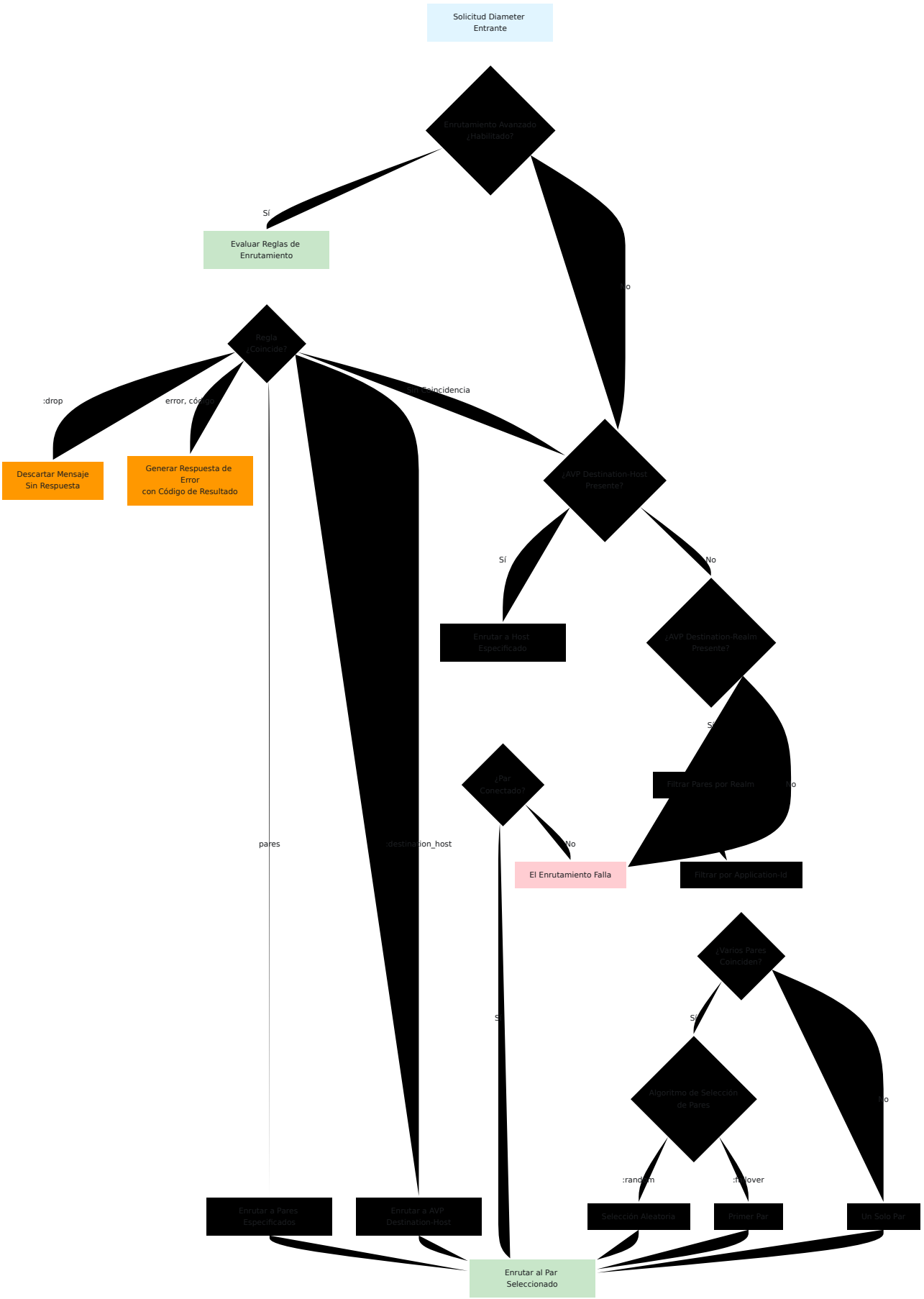
1. Enrutamiento Diameter Estándar
2. Configuración Base del DRA
3. Multihoming SCTP
4. Tablas de Referencia
    - IDs de Aplicación 3GPP Comunes
    - Códigos AVP Comunes
5. Módulo de Enrutamiento Avanzado
6. Módulo de Transformación Avanzada
7. Procesamiento de Reglas
8. Módulo de Métricas Extendidas
9. Módulos de Seguridad y Control
10. Métricas de Prometheus
    - Métricas de Diameter Core
    - Métricas del Módulo de Enrutamiento Avanzado
11. Solución de Problemas

Visión General de la Arquitectura DRA



Enrutamiento Diameter Estándar

Sin los módulos Enrutamiento Avanzado o Transformación Avanzada, el DRA realiza el enrutamiento Diameter estándar basado en el Protocolo Base Diameter (RFC 6733):


Solicitud Diameter Entrante
Enrutamiento Avanzado ¿Habilitado?
Sí
No
Evaluar Reglas de Enrutamiento
Regla ¿Coincide?
:drop
error, código
Descartar Mensaje Sin Respuesta
Generar Respuesta de Error con Código de Resultado
Coincidencia
¿AVP Destination-Host Presente?
Sí
No
Enrutar a Host Especificado
¿AVP Destination-Realm Presente?
Filtrar Pares por Realm
Filtrar por Application-Id
¿Par Conectado?
No
El Enrutamiento Falla
pares
:destination_host
¿Varios Pares Coinciden?
Algoritmo de Selección de Pares
:random
:failover
Selección Aleatoria
Primer Par
Un Solo Par
Enrutar a Pares Especificados
Enrutar a AVP Destination-Host
Enrutar al Par Seleccionado

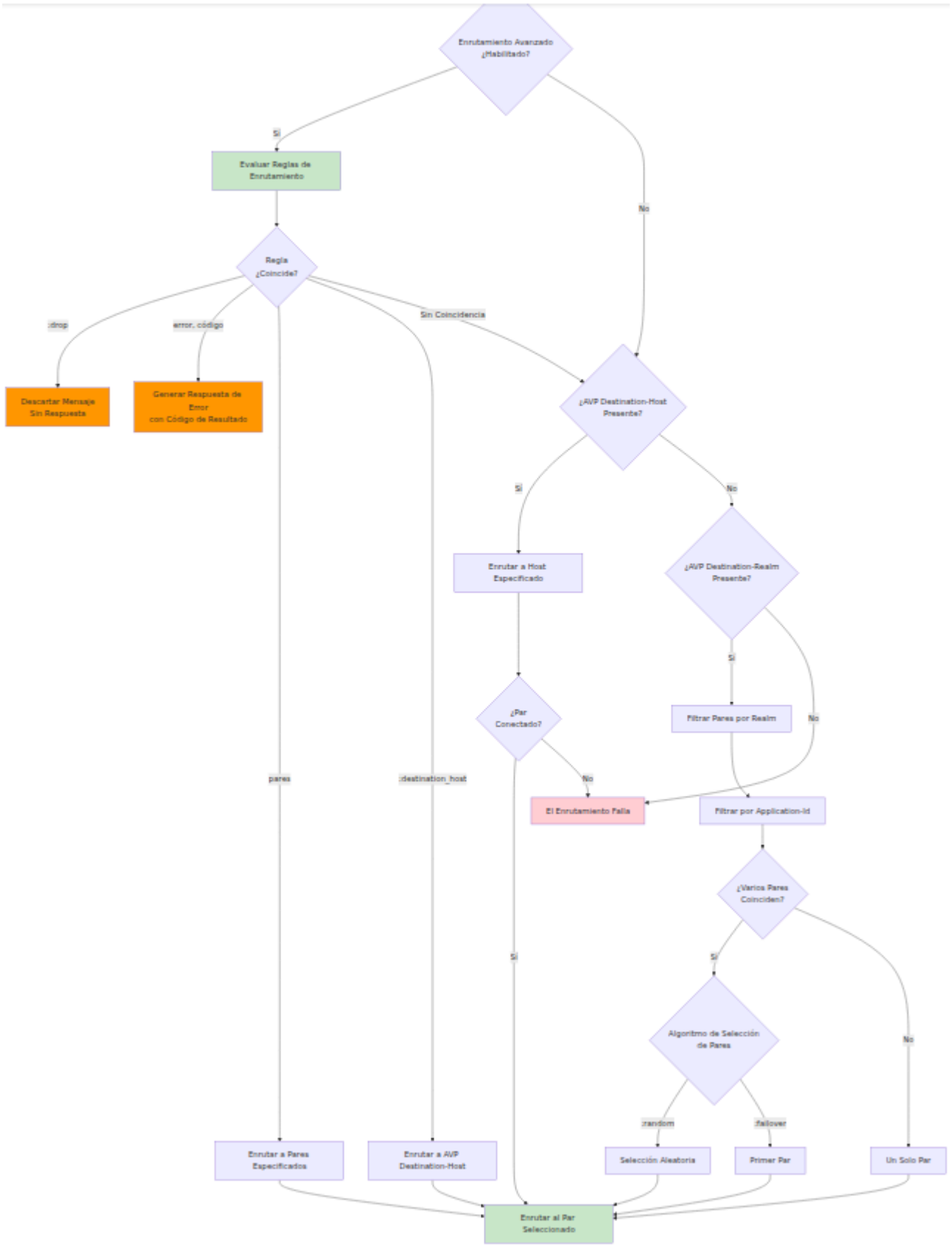
Enrutamiento de Solicitudes

El DRA enruta mensajes de solicitud utilizando un mecanismo basado en prioridades definido en RFC 6733 Sección 6.1:

1. **AVP Destination-Host (293)** - Si está presente, el DRA enruta directamente al par especificado
    - Este es el mecanismo de enrutamiento de mayor prioridad
    - Si el par no está conectado, el enrutamiento falla
    - Proporciona control de enrutamiento explícito a nivel de host
2. **AVP Destination-Realm (283)** - Si Destination-Host está ausente, enruta según el realm
    - El DRA selecciona un par conectado que anuncia soporte para el realm objetivo
    - Se aplica balanceo de carga cuando varios pares coinciden con el realm
    - El enrutamiento basado en realm permite flexibilidad entre múltiples hosts
3. **Application-Id** - Los pares se filtran por aplicaciones Diameter soportadas
    - Solo se consideran los pares que anuncian soporte para el Application-Id del mensaje
    - Basado en el Intercambio de Capacidades (CER/CEA) durante el establecimiento de conexión de pares
    - Ver IDs de Aplicación 3GPP Comunes para referencia

Enrutamiento de Respuestas

Los paquetes de respuesta utilizan un mecanismo de enrutamiento fundamentalmente diferente al de las solicitudes:



- **Enrutamiento basado en sesión**: Los paquetes de respuesta siempre siguen el camino inverso de la solicitud

- **Preservación del ID de extremo a extremo**: El Identificador de Extremo a Extremo permanece sin cambios a través de todos los saltos
- **Enrutamiento de salto a salto**: El DRA utiliza el Identificador de Salto a Salto para mantener el estado de enrutamiento (cambia en cada salto)
- **Sin evaluación de reglas**: El DRA no evalúa reglas de enrutamiento ni contenidos de AVP para respuestas
- **Correlación con estado**: Las tablas de enrutamiento internas rastrean qué par envió cada solicitud

Por qué las respuestas no son enrutadas por módulos avanzados:

- El enrutamiento de respuestas es determinista y debe regresar al par de origen
- El protocolo Diameter requiere que las respuestas sigan el camino de solicitud establecido
- Las decisiones de enrutamiento para respuestas se toman en función del contexto de la solicitud original, no del contenido de la respuesta
- Esto asegura una gestión adecuada de sesiones y previene bucles de enrutamiento

Ver RFC 6733 Sección 6.2 para detalles sobre el enrutamiento de mensajes de respuesta.

Selección de Pares

Cuando varios pares coinciden con los criterios de enrutamiento, el `peer_selection_algorithm` configurado determina la selección:

- `:random` - Selecciona aleatoriamente de los pares disponibles (predeterminado)
- `:failover` - Siempre selecciona el primer par en la lista (basado en prioridad)
- Los pares deben estar en **estado conectado** para ser seleccionados
- Los pares desconectados o inactivos se excluyen automáticamente

Limitaciones del Enrutamiento Estándar

- No hay reglas de enrutamiento personalizadas basadas en valores de AVP (por ejemplo, patrones de IMSI)
- No hay traducción de realm ni modificación de AVP
- No se puede enrutar según el par de origen
- Control limitado sobre la distribución del tráfico

Los módulos Enrutamiento Avanzado y Transformación Avanzada extienden este comportamiento estándar con capacidades de enrutamiento basado en reglas y manipulación de paquetes.

Configuración Base del DRA

El DRA requiere una configuración base que defina su identidad, configuraciones de red y conexiones de pares. Esta configuración establece la base para todas las operaciones de enrutamiento.

Estructura de Configuración

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

Parámetros de Identidad del DRA

Parámetro	Tipo	Descripción
`host`	String	La Identidad Diameter del DRA (nombre de dominio completamente calificado)
`realm`	String	El realm Diameter del DRA
`product_name`	String	Nombre del producto anunciado en mensajes CER/CEA
`vendor_id`	Integer	Vendor-ID según se define en RFC 6733 Sección 5.3.3 (10415 = 3GPP)

Configuraciones de Red

Parámetro	Tipo	Descripción
`listen_ip`	String o Lista	Dirección(es) IP en las que el DRA escucha. Para multihoming SCTP, use una lista de cadenas IP (ver Multihoming SCTP)
`listen_port`	Integer	Puerto TCP/SCTP para conexiones Diameter (estándar: 3868)
`service_name`	Atom	Identificador de servicio interno de Erlang
`request_timeout`	Integer	Tiempo de espera en milisegundos para pares de solicitud/respuesta (predeterminado: 5000)

Configuraciones de Selección de Pares

Parámetro	Tipo	Descripción
`peer_selection_algorithm`	Atom	Algoritmo de balanceo de carga: `:random` (selección aleatoria) o `:failover` (prioridad del primer par)
`allow_undefined_peers_to_connect`	Boolean	Permitir conexiones de pares no configurados (predeterminado `false`)
`log_unauthorized_peer_connection_attempts`	Boolean	Registrar intentos de conexión de pares no autorizados

Configuración de Pares

Cada par en la lista `peers` define una conexión Diameter:

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

Parámetros de Par

Parámetro	Tipo	Descripción
`host`	String	Identidad Diameter del par (FQDN) - debe coincidir exactamente para el enrutamiento
`realm`	String	Realm Diameter del par
`ip`	String	Dirección IP principal del par para la conexión (requerido)
`ips`	Lista	Lista de direcciones IP para multihoming SCTP (opcional, ver Multihoming SCTP)
`port`	Integer	Puerto Diameter del par (típicamente 3868)
`transport`	Atom	Protocolo de transporte: `:diameter_tcp` o `:diameter_sctp`
`tls`	Boolean	Habilitar cifrado TLS (si `true`, típicamente usar puerto 3869)
`initiate_connection`	Boolean	`true`: DRA se conecta al par, `false`: DRA espera a que el par se conecte

Modos de Conexión

Iniciar Conexión (`initiate_connection: true`)

- DRA actúa como cliente Diameter
- DRA inicia conexión TCP/SCTP al par
- Usado para conectarse a HSS, PCRF u otros sistemas backend
- DRA volverá a intentar conexiones si el par no es accesible

Aceptar Conexión (`initiate_connection: false`)

- DRA actúa como servidor Diameter
- DRA espera a que el par se conecte
- Usado para conexiones MME, SGSN, P-GW
- El par debe estar en la configuración o `allow_undefined_peers_to_connect: true`

Ejemplo de Configuración

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

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

Notas Importantes

- **Coincidencia de Nombres de Host**: Los nombres de host de los pares en las reglas de Enrutamiento Avanzado deben coincidir exactamente con el valor `host` configurado aquí (sensible a mayúsculas y minúsculas)
- **Intercambio de Capacidades**: Al conectarse, los pares intercambian aplicaciones soportadas a través de mensajes CER/CEA
- **Soporte de Aplicaciones**: El DRA anuncia todas las aplicaciones 3GPP soportadas (ver IDs de Aplicación 3GPP Comunes)
- **Vendor-ID 10415**: Valor estándar para aplicaciones 3GPP
- **Tiempo de Espera de Solicitud**: Afecta el TTL de Métricas Extendidas (tiempo de espera + 5 segundos)
- **Selección de Pares**: Cuando varios pares coinciden con los criterios de enrutamiento, `peer_selection_algorithm` determina cuál es elegido

Consideraciones de Seguridad

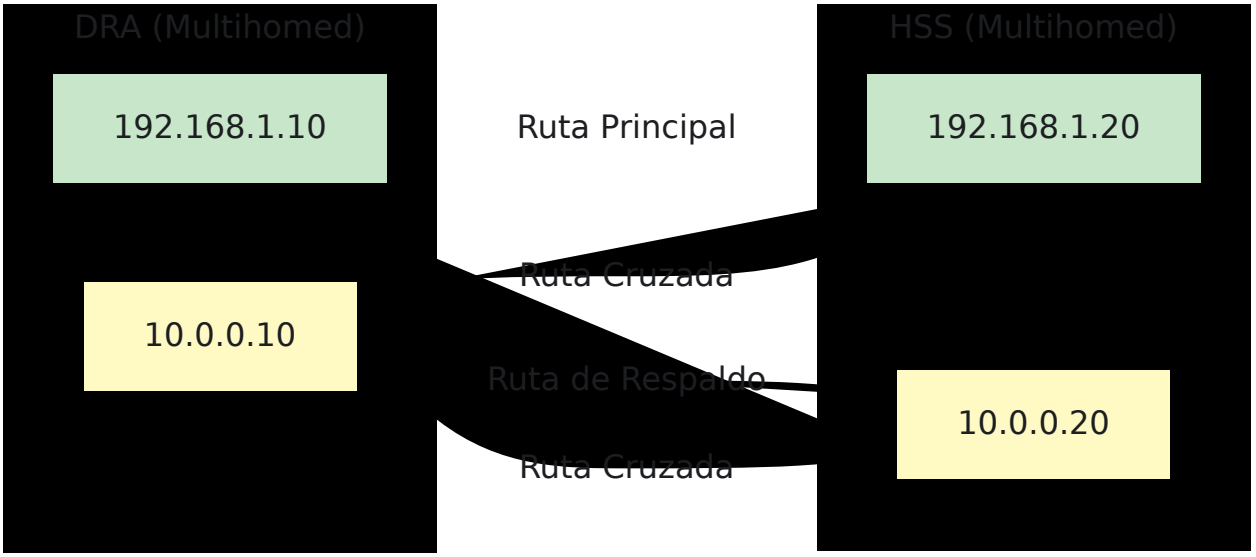
- Establecer `allow_undefined_peers_to_connect: false` en producción
- Habilitar `log_unauthorized_peer_connection_attempts: true` para monitoreo de seguridad
- Asegurarse de que las reglas del firewall coincidan con las configuraciones de `listen_ip` y `listen_port`
- Validar certificados de pares al usar TLS

Multihoming SCTP

El multihoming SCTP proporciona redundancia de red al permitir que los puntos finales se vinculen a múltiples direcciones IP. Si la ruta de red principal falla,

SCTP cambia automáticamente a una ruta alternativa sin interrumpir la sesión Diameter.

Cómo Funciona



- Los latidos SCTP monitorean todas las rutas de red
- El cambio automático ocurre si la ruta principal se vuelve inalcanzable
- No hay interrupción de la sesión Diameter durante el cambio de ruta
- El kernel maneja la selección de ruta automáticamente

Configuración

Direcciones de Escucha del DRA

Configure múltiples direcciones IP locales para que el DRA se vincule:

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

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

  listen_port: 3868,
  ...
}
```

Notas:

- El transporte TCP utiliza solo la primera IP de la lista
- El transporte SCTP se vincula a todas las IPs especificadas
- El formato de cadena IP única sigue siendo totalmente compatible

Configuración de Pares

Configure múltiples direcciones IP remotas para conexiones de pares:

```
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 multihoming
    port: 3868,
    transport: :diameter_sctp,
    tls: false,
    initiate_connection: true
  }
]
```

Notas:

- El campo `ip` es requerido para compatibilidad hacia atrás
- El campo `ips` es opcional; si se omite, solo se usa `ip`
- Para SCTP, incluya la IP principal en la lista `ips`
- Para TCP, solo se usa `ip` (TCP no soporta multihoming)

Ejemplo Completo

```
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,
        tls: false,
        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,
        tls: false,
        initiate_connection: false
      }
    ]
  }
```

Requisitos

- El módulo del kernel SCTP debe estar cargado (paquete `lksctp-tools` en Linux)
- Todas las direcciones IP deben ser enrutables desde/hacia el par
- Las reglas del firewall deben permitir tráfico SCTP en todas las IPs configuradas
- Ambos puntos finales deben estar configurados para multihoming para una redundancia completa

Limitaciones

- El transporte TCP no soporta multihoming (solo utiliza la IP principal)
- TLS sobre multihoming SCTP puede tener limitaciones de compatibilidad
- El tiempo de conmutación de ruta depende de los parámetros SCTP del kernel

---

Tablas de Referencia

IDs de Aplicación 3GPP Comunes

Application-Id	Interfaz	Descripción
16777251	S6a/S6d	MME/SGSN a HSS autenticación y datos de suscripción
16777252	S13/S13'	MME a EIR verificación de identidad del equipo
16777238	Gx	PCEF a PCRF control de políticas y cobro
16777267	S9	PCRF de origen a PCRF visitado política de roaming
16777272	Sy	PCRF a OCS vinculación de sesión
16777216	Cx	I-CSCF/S-CSCF a HSS registro IMS
16777217	Sh	AS a HSS datos de usuario IMS
16777236	SLg	MME/SGSN a GMLC servicios de localización
16777291	SLh	GMLC a HSS información de suscriptor de localización
16777302	S6m	MTC-IWF a HSS/HLR para dispositivos M2M
16777308	S6c	SMS-SC/IP-SM-GW a HSS enrutamiento de SMS
16777343	S6t	SCEF a HSS eventos de monitoreo

Application-Id	Interfaz	Descripción
16777334	Rx	AF a PCRF autorización de medios

Códigos AVP Comunes

Código	Nombre AVP	Tipo	Uso
1	User-Name	UTF8String	Identificador de suscriptor (IMSI en 3GPP)
264	Origin-Host	DiameterIdentity	Nombre de host del par de origen
268	Result-Code	Unsigned32	Código de resultado estándar
283	Destination-Realm	DiameterIdentity	Realm objetivo
293	Destination-Host	DiameterIdentity	Host objetivo (opcional)
296	Origin-Realm	DiameterIdentity	Realm de origen
297	Experimental-Result	Grouped	Código de resultado específico del proveedor

Códigos de Comando Comunes

Los códigos de comando son parte del encabezado del mensaje Diameter, no de los AVP:

Código	Nombre de Comando	Descripción
257	CER/CEA	Solicitud/Respuesta de Intercambio de Capacidades
258	RAR/RAA	Solicitud/Respuesta de Reautenticación
274	ASR/ASA	Solicitud/Respuesta de Abort-Session
275	STR/STA	Solicitud/Respuesta de Terminación de Sesión
280	DWR/DWA	Solicitud/Respuesta de Dispositivo-Vigilante
282	DPR/DPA	Solicitud/Respuesta de Desconectar-Par
316	ULR/ULA	Solicitud/Respuesta de Actualización de Ubicación (S6a)
317	CLR/CLA	Solicitud/Respuesta de Cancelar Ubicación (S6a)
318	AIR/AIA	Solicitud/Respuesta de Información de Autenticación (S6a)
321	PUR/PUA	Solicitud/Respuesta de Purga-UE (S6a)

Módulo de Enrutamiento Avanzado

El módulo de Enrutamiento Avanzado proporciona capacidades de enrutamiento de mensajes flexibles y basadas en reglas con soporte para condiciones de coincidencia complejas.

Importante: Este módulo evalúa **solo paquetes de solicitud Diameter entrantes** (no paquetes de respuesta). Los paquetes de respuesta siguen el enrutamiento de sesión establecido de regreso al par de origen - ver Enrutamiento de Respuestas para detalles.

Configuración

Habilite el módulo y defina reglas de enrutamiento en su configuración:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: <identificador_regla>
      match: <alcance_coincidencia>
      filters: [<lista_filtros>]
      route:
        peers: [<lista_pares>]
```

Parámetros

Parámetro	Descripción
`enabled`	Establecer en `True` para activar el módulo
`rule_name`	Identificador único para la regla de enrutamiento
`match`	Cómo se combinan los filtros: `:all` (lógica AND - todos los filtros deben coincidir), `:any` (lógica OR - al menos un filtro debe coincidir), `:none` (lógica NOR - ningún filtro puede coincidir)
`filters`	Lista de condiciones de filtro (ver Filtros Disponibles)
`route`	Acción de enrutamiento (ver Acciones de Enrutamiento a continuación)

Acciones de Enrutamiento

El parámetro `route` admite múltiples acciones:

Enrutar a Pares

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

Enruta a los nombres de host de pares especificados. Los pares deben ser:

- Definidos en la configuración de pares Diameter del DRA
- El nombre de host exacto como se configuró (sensible a mayúsculas y minúsculas)
- Actualmente conectados para que el enrutamiento tenga éxito (los pares desconectados se omiten)

Enrutar a AVP Destination-Host

```
route: :destination_host
```

Enruta al par especificado en el AVP Destination-Host (293) del mensaje. Si falta el AVP Destination-Host, el enrutamiento vuelve al comportamiento normal.

Descartar Tráfico

```
route: :drop
```

Descarta silenciosamente el mensaje sin enviar ninguna respuesta. Úselo para:

- Filtrado de tráfico y blackholing
- Bloqueo de solicitudes no deseadas
- Limitación de tasa al descartar tráfico excesivo

Comportamiento:

- El mensaje se descarta en el DRA (no se reenvía)
- No se envía mensaje de respuesta al par solicitante
- Implementa el comportamiento de Erlang Diameter `:discard`
- Métrica: `diameter_advanced_routing_drop_count_total` (ver Métricas de Prometheus)

Generar Respuesta de Error

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

Genera una respuesta de error Diameter con el Código de Resultado especificado y la envía de regreso al par solicitante. Códigos de resultado comunes:

- `3002` - DIAMETER_UNABLE_TO_DELIVER (enrutamiento no disponible)
- `3003` - DIAMETER_REALM_NOT_SERVED (realm no soportado)
- `3004` - DIAMETER_TOO_BUSY (protección contra sobrecarga, limitación de tasa)
- `5012` - DIAMETER_UNABLE_TO_COMPLY (rechazo general)

Comportamiento:

- El DRA genera respuesta de error con el Código de Resultado especificado
- La respuesta incluye Origin-Host, Origin-Realm, Session-Id (autopoblado por Diameter)
- El mensaje NO se reenvía a ningún par
- Implementa Erlang Diameter `{:protocol_error, code}` (equivalente a `{:answer_message, code}`)
- Métrica: `diameter_advanced_routing_error_count_total` (ver Métricas de Prometheus)

Filtros Disponibles

Filtros Estándar

Disponibles en Enrutamiento Avanzado y Transformación Avanzada:

- `:application_id` - Coincidir ID de aplicación Diameter (ver Referencia de ID de Aplicación)
  - Valor único: `{:application_id, 16777251}` (S6a/S6d)
  - Varios valores: `{:application_id, [16777251, 16777252]}` (S6a o S6b)
- `:command_code` - Coincidir código de comando Diameter
  - Valor único: `{:command_code, 318}` (solicitud AIR)
  - Varios valores: `{:command_code, [317, 318]}` (ULR o AIR)
- `:avp` - Coincidir valor de AVP (ver Referencia de código AVP)
  - Coincidencia exacta: `{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}`
  - Coincidencia regex: `{:avp, {1, ~r"999001.*"}}`
  - Varios patrones: `{:avp, {1, ["505057001313606", ~r"999001.*", ~r"505057.*"]}}`
  - Cualquier valor (verificación de presencia): `{:avp, {264, :any}}`

Filtro Específico de Enrutamiento

Solo disponible en Enrutamiento Avanzado:

- `:via_peer` - Coincidir el par desde el cual se recibió la solicitud
  - Un solo par: `{:via_peer, "omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org"}`
  - Varios pares: `{:via_peer, ["omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org", "omnitouch-lab-dra02.epc.mnc001.mcc001.3gppnetwork.org"]}`
  - Cualquier par: `{:via_peer, :any}`

Filtros Específicos de Transformación

Solo disponibles en Transformación Avanzada:

- `:to_peer` - Coincidir en el par de destino predeterminado (solo paquetes de solicitud)

    - Un solo par: `{:to_peer, "dra01.omnitouch.com.au"}`
    - Varios pares: `{:to_peer, ["dra01.omnitouch.com.au", "dra02.omnitouch.com.au"]}`

- **`:from_peer`** - Coincidir el par que envió la respuesta (solo paquetes de respuesta)
    - Un solo par: `{:from_peer, "hss-01.example.com"}`
    - Varios pares: `{:from_peer, ["hss-01.example.com", "hss-02.example.com"]}`

- **`:packet_type`** - Coincidir la dirección del paquete
    - Solicitud: `{:packet_type, :request}`
    - Respuesta: `{:packet_type, :answer}`

Notas Importantes sobre Filtros

- **Filtros AVP**: Recomendados solo para AVPs simples (User-Name, Origin-Host, Destination-Realm, etc.)
    - Los AVPs agrupados **no son soportados** y no coincidirán
    - Valores binarios complejos **no son soportados**
    - Usar formato: `{:avp, {code, value}}`

- **Operadores de Lista**: Soportados para todos los valores de filtro excepto `:packet_type`
    - Cuando se usa una lista, aplica lógica **OR** dentro de la lista
    - Ejemplo: `{:command_code, [317, 318]}` coincide con el código de comando 317 **O** 318

- **Valores Especiales**:
    - `:any` - Coincide con cualquier valor (verifica la presencia de AVP)
    - Ejemplo: `{:avp, {264, :any}}` coincide si el AVP Origin-Host existe con cualquier valor

Ejemplos de Enrutamiento

Ejemplo 1: Enrutamiento por Par de Origen

Enrutar mensajes según el DRA desde el cual llegaron:

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

Cómo funciona: Enruta el tráfico S6a que llega a través de pares DRA específicos a nodos HSS locales.

Ejemplo 2: Roaming Entrante con Coincidencia de Patrones

Enrutar tráfico de roaming basado en patrones de IMSI:

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

Cómo funciona: Enruta mensajes S6a desde un Realm de Origen específico con patrones de IMSI coincidentes a pares DRA designados.

Ejemplo 3: Enrutamiento Dinámico con :destination_host

Enrutar al valor del AVP Destination-Host en el mensaje:

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

Cómo funciona:

- Cuando los filtros coinciden, enruta al par especificado en el AVP Destination-Host (293)
- Si falta el AVP Destination-Host, la coincidencia se considera un fallo y vuelve al enrutamiento normal
- Útil para honrar el enrutamiento cuando el remitente especifica el destino exacto

Ejemplo 4: Descartar Tráfico No Deseado

Descartar tráfico de rangos de IMSI específicos:

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

Cómo funciona:

- Coincide con mensajes S6a con IMSI que comienzan con 999999
- Silenciosamente descarta el mensaje sin enviar ninguna respuesta
- Útil para filtrar tráfico de prueba o bloquear rangos de suscriptores específicos
- Ver Métricas de Prometheus para monitorear el tráfico descartado

Ejemplo 5: Limitación de Tasa con Respuestas de Error

Devolver DIAMETER_TOO_BUSY para patrones de tráfico específicos:

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

Cómo funciona:

- Coincide con tráfico S6a de un par sobrecargado específico
- Devuelve la respuesta de error DIAMETER_TOO_BUSY (3004)
- El par solicitante recibe un error y debe retroceder
- Útil para protección contra sobrecarga y limitación de tasa
- Ver Métricas de Prometheus para monitorear respuestas de error

Ejemplo 6: Respuestas de Error Condicionales por Comando

Bloquear tipos de comando específicos con códigos de error apropiados:

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

Cómo funciona:

- Coincide con mensajes de solicitud de Purga-UE S6a
- Devuelve el error DIAMETER_UNABLE_TO_COMPLY (5012)
- Bloquea operaciones específicas sin descartar tráfico silenciosamente
- Útil para deshabilitar selectivamente ciertos comandos Diameter

---

Módulo de Transformación Avanzada

El módulo de Transformación Avanzada permite la modificación dinámica de los AVP de mensajes Diameter según criterios de coincidencia. Ver Procesamiento de Reglas para detalles sobre cómo se evalúan las reglas.

Configuración

Habilite el módulo y defina reglas de transformación:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: <identificador_regla>
      match: <alcance_coincidencia>
      filters: [<lista_filtros>]
      transform:
        action: <acción_transformación>
        avps: [<modificaciones_avp>]
```

Parámetros

Parámetro	Descripción
`enabled`	Establecer en `True` para activar el módulo
`rule_name`	Identificador único para la regla de transformación
`match`	Cómo se combinan los filtros: `:all` (lógica AND), `:any` (lógica OR), `:none` (lógica NOR) - ver Lógica de Filtros
`filters`	Lista de condiciones de filtro (ver Filtros Disponibles)
`transform.action`	Tipo de transformación (`:edit`, `:remove`, o `:overwrite`)
`transform.avps`	Lista de modificaciones de AVP a aplicar (ver Referencia de código AVP)

Acciones de Transformación

Paquetes de Solicitud (Solicitudes Diameter)

- **`:edit`** - Modificar valores de AVP existentes
  - Solo modifica AVPs que existen en el mensaje

    - Si el AVP no existe, no se realiza ningún cambio
- `:remove` - Eliminar AVPs del mensaje
- `:overwrite` - Reemplazar estructuras completas de AVP
    - Requiere el parámetro `dictionary` que especifica el diccionario Diameter (por ejemplo, `:diameter_gen_3gpp_s6a`)

Paquetes de Respuesta (Respuestas Diameter)

- `:remove` - Eliminar AVPs del mensaje
- `:overwrite` - Reemplazar estructuras completas de AVP
    - Requiere el parámetro `dictionary`

Importante: Si no coinciden reglas, el paquete se pasa a través de forma transparente sin transformaciones.

Sintaxis de Modificación de AVP

Modificación estándar:

- `{:avp, {<código>, <nuevo_valor>}}` - Establecer AVP en nuevo valor

Eliminando AVPs:

- `{:avp, {<código>, :any}}` - Eliminar AVP por ID (elimina independientemente del valor actual)
- Nota: Eliminar basado en avp_id es soportado; eliminar basado en contenidos de AVP no es soportado

Sobrescribir con diccionario:

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

Ejemplos de Transformación

Ejemplo 1: Reescritura de Realm de Destino Basada en el Par

Reescribir Destination-Realm según a dónde se está enrutando el mensaje:

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

Cómo funciona: Cuando se enrutan solicitudes S6a a pares DRA específicos y coinciden con el patrón de IMSI, reescribe el Destination-Realm para la red del Operador X.

Ejemplo 2: Enrutamiento de Varios Transportistas con Transformaciones

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

Cómo funciona: Dirige diferentes rangos de suscriptores IMSI a los realms de red apropiados según los patrones de IMSI. La primera regla que coincida gana (ver Orden de Ejecución).

Ejemplo 3: Reescritura de Realm para MVNO

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

Cómo funciona: Transforma el Destination-Realm para un suscriptor MVNO específico a su red central alojada.

Ejemplo 4: Transformación Solo de Solicitud con Filtro de Tipo de Paquete

Transformar solo paquetes de solicitud (no respuestas):

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

Cómo funciona:

- Coincide solo con paquetes de **solicitud** S6a (no paquetes de respuesta)
- Verifica que User-Name (AVP 1) sea igual a "999999000000001"
- Verifica que Origin-Host (AVP 264) exista con cualquier valor
- Reescribe User-Name a "999999000000002"
- Si el AVP no existe, no se realiza ningún cambio

Ejemplo 5: Eliminar AVP

Eliminar un AVP específico de los mensajes:

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

Cómo funciona: Elimina el AVP User-Name (código 1) de todos los mensajes S6a, independientemente de su valor actual.

Ejemplo 6: Sobrescribir AVP Agrupado en Paquetes de Respuesta

Modificar AVPs agrupados complejos en paquetes de respuesta utilizando la acción `:overwrite` con soporte de diccionario:

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

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

Cómo funciona:

- Coincide con paquetes de respuesta S6a Update Location (ULA) de un Realm de Origen específico
- Utiliza la acción `:overwrite` para reemplazar todo el AVP agrupado APN-Configuration-Profile
- **Requiere el parámetro `dictionary`** para codificar correctamente estructuras de AVP agrupadas complejas
- Agrega tres configuraciones de APN: "internet" (contexto 1), "ims" (contexto 2) y "sos" (contexto 3)
- Cada APN incluye perfiles de QoS, límites de ancho de banda (AMBR) y configuraciones de tipo PDN
- La transformación asegura que el APN de servicios de emergencia (SOS) esté provisionado para todos los suscriptores de este realm

Cuándo usar `:overwrite` con diccionario:

- Modificar AVPs agrupados con estructuras anidadas (como APN-Configuration-Profile)
- Agregar o reestructurar datos de suscripción complejos de 3GPP
- Cuando la acción `:edit` no puede manejar la complejidad del AVP
- El diccionario debe coincidir con la aplicación Diameter (`:diameter_gen_3gpp_s6a` para S6a, etc.)

Notas importantes:

- `:overwrite` reemplaza todo el AVP, no solo campos individuales
- La estructura del AVP debe coincidir exactamente con la definición del diccionario
- Una estructura incorrecta causará fallos de codificación y paquetes descartados

- Esta es una característica avanzada - valide exhaustivamente en el entorno de prueba primero

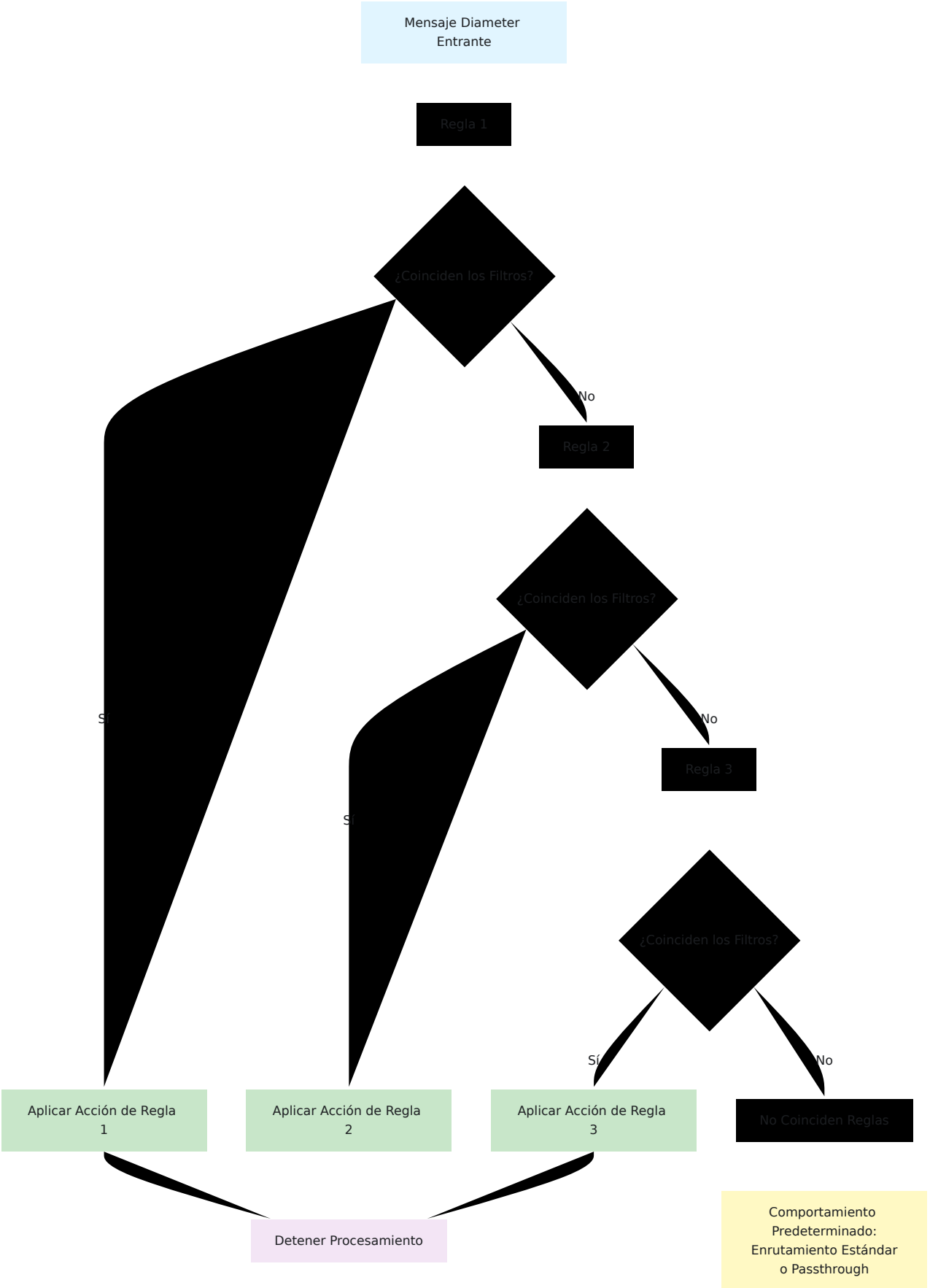
Casos de Uso

- **Soporte para MVNO**: Enrutar tráfico de operadores virtuales a redes centrales alojadas
- **Migración de Red**: Redirigir gradualmente suscriptores a nueva infraestructura
- **Traducción de Realm**: Convertir entre diferentes esquemas de nombres para socios de roaming
- **Multi-tenencia**: Aislar poblaciones de suscriptores por realm
- **Enrutamiento de Transportistas**: Dirigir tráfico a redes de transportistas correctas según rangos de IMSI

Procesamiento de Reglas

Aplica a los módulos Enrutamiento Avanzado y Transformación Avanzada.

Orden de Ejecución



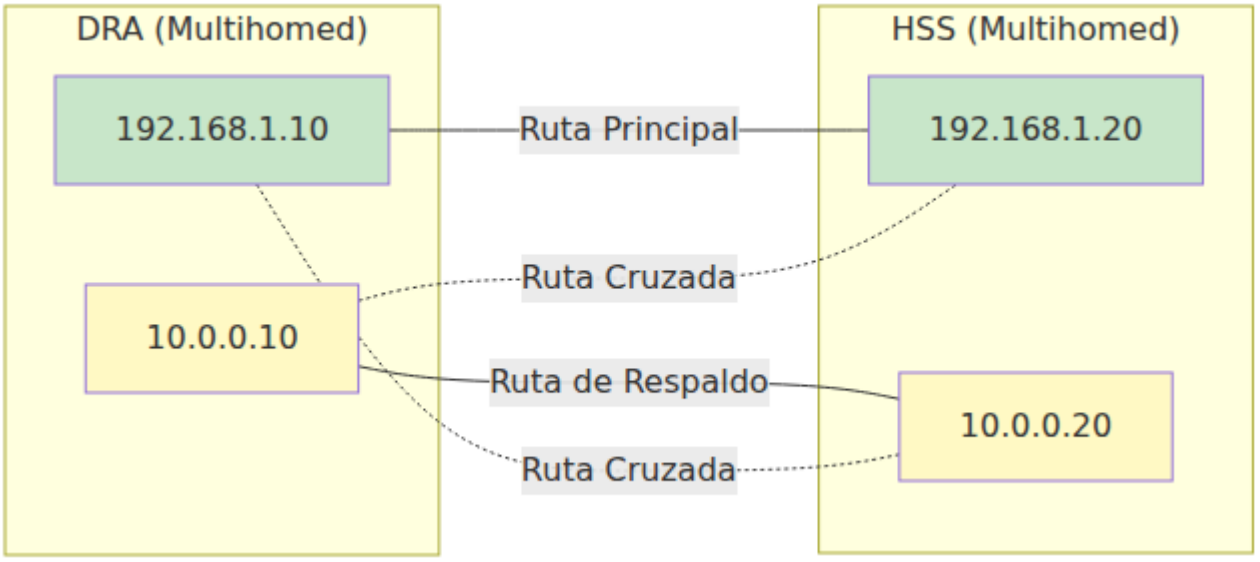
1. Las reglas se evalúan **en orden de arriba hacia abajo** según se definen en la configuración
2. Los filtros dentro de una regla se evalúan según el parámetro `match` (`:all`, `:any`, o `:none`)
3. **La primera regla que coincide gana** - las reglas subsiguientes no se evalúan
4. Si no coinciden reglas, se utiliza el comportamiento de enrutamiento/passthrough predeterminado

Lógica de Filtros

El parámetro `match` determina cómo se combinan los filtros:

match: :all (Lógica AND)

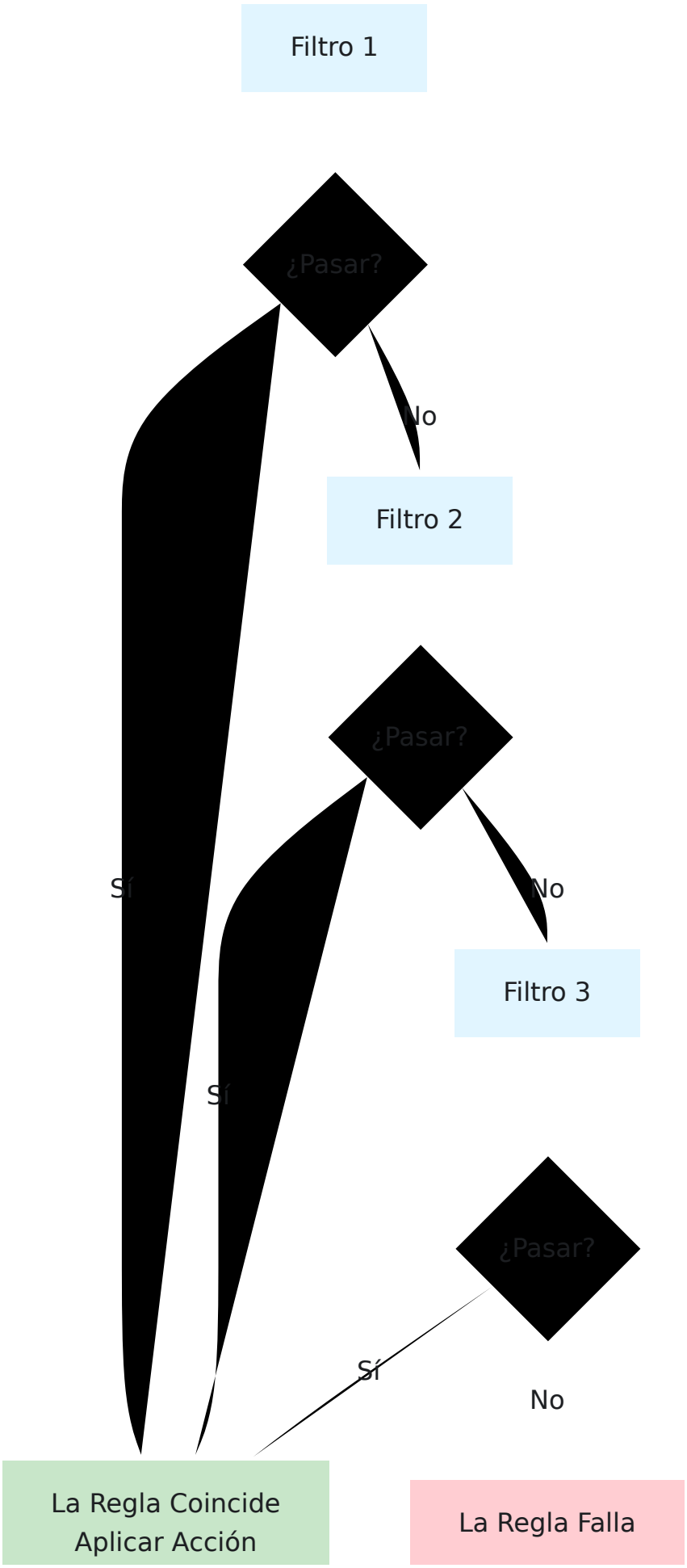
Todos los filtros deben coincidir para que la regla tenga éxito.



Ejemplo: Con 3 filtros, `filtro1 Y filtro2 Y filtro3` deben ser todos verdaderos.

match: :any (Lógica OR)

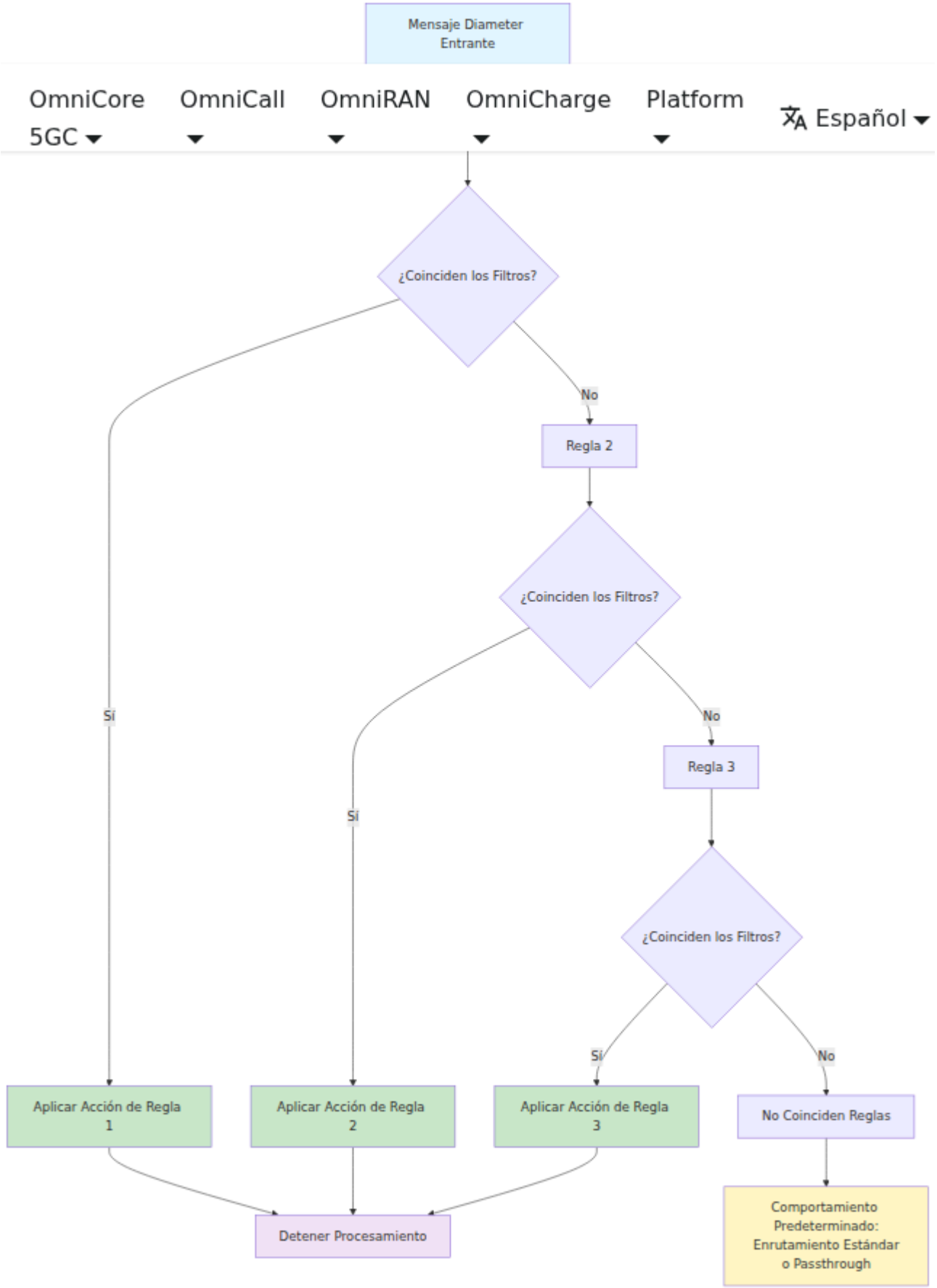
Al menos un filtro debe coincidir para que la regla tenga éxito.


Filtro 1
¿Pasar?
No
Filtro 2
¿Pasar?
Sí
No
Filtro 3
Sí
¿Pasar?
Sí
No
La Regla Coincide
Aplicar Acción
La Regla Falla

Ejemplo: Con 3 filtros, `filtro1 O filtro2 O filtro3` (cualquiera pasa).

match: :none (Lógica NOR)

Ningún filtro puede coincidir para que la regla tenga éxito (coincidencia inversa).



Ejemplo: Con 3 filtros, `NO filtro1 Y NO filtro2 Y NO filtro3` (todos deben fallar).

Notas Adicionales:

Al usar operadores de lista dentro de un valor de filtro (por ejemplo, `{:avp, {1, ["value1", "value2"]}}`), los valores utilizan lógica **OR** (cualquiera puede coincidir).

Patrones de Expresión Regular

Use la sintaxis `~r"pattern"` para coincidencias regex:

- `~r"999001.*"` - Coincide con IMSI que comienza con 999001
- `~r"^310[0-9]{3}.*"` - Coincide con IMSI con patrones MNC específicos
- `~r".*test$"` - Coincide con valores que terminan en "test"

Mejores Prácticas

1. **Especificidad**: Ordene las reglas de lo más específico a lo más general
2. **Rendimiento**: Coloque las coincidencias más comunes primero para reducir la sobrecarga de procesamiento
3. **Pruebas**: Valide patrones regex antes de la implementación
4. **Documentación**: Use valores descriptivos de `rule_name` para claridad operativa
5. **Monitoreo**: Realice un seguimiento de las tasas de coincidencia de reglas para verificar el comportamiento esperado

---

Módulo de Métricas Extendidas

El módulo de Métricas Extendidas proporciona capacidades avanzadas de telemetría y análisis para analizar patrones de tráfico Diameter más allá de las métricas estándar.

Configuración

Habilite el módulo y configure tipos de métricas específicas:

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

Parámetros

Parámetro	Descripción
`enabled`	Establecer en `true` para activar el módulo de métricas extendidas
`attach_attempt_reporting_enabled`	Habilitar seguimiento e informes de intentos de conexión LTE (S6a AIR/AIA)

Métricas Disponibles

Seguimiento de Intentos de Conexión

Rastrea intentos de conexión de suscriptores LTE monitoreando pares de mensajes de Solicitud de Información de Autenticación (AIR) y Respuesta (AIA):

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

Intente de nuevo

Medición: `attach_attempt_count`

Campos:

- `imsi` - El IMSI del suscriptor (del AVP User-Name - ver códigos AVP)

Etiquetas:

- `origin_host` - El par que originó la solicitud de conexión
- `result_code` - El código de resultado Diameter de la respuesta HSS

Cómo funciona:

1. Cuando se recibe una AIR (código de comando 318, aplicación S6a 16777251 - ver IDs de Aplicación), el módulo extrae:
   - ID de Extremo a Extremo para correlación de solicitud/respuesta
   - IMSI (AVP User-Name código 1)
   - Origin-Host (AVP código 264)
2. Los metadatos de la solicitud se almacenan en ETS con TTL
3. Cuando se recibe la AIA coincidente, el módulo:
   - Correlaciona usando ID de Extremo a Extremo
   - Extrae el código de resultado (AVP 268 o AVP de resultado experimental AVP 297)
   - Emite la métrica con IMSI, origin host y código de resultado

Casos de Uso

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

Integración

Las métricas extendidas se exportan a través de la integración de InfluxDB:

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

Los códigos de resultado son códigos estándar de Diameter:

- `2001` - Éxito (DIAMETER_SUCCESS)
- `5001` - Fallo de autenticación (DIAMETER_AUTHENTICATION_REJECTED)
- `5004` - AVP Diameter no soportado
- Ver RFC 6733 para la lista completa de códigos de resultado

Notas Importantes

- Las métricas de intentos de conexión solo rastrean pares de AIR/AIA de S6a (ID de Aplicación 16777251, Código de Comando 318)
- Los metadatos de la solicitud expiran según el tiempo de espera de solicitud configurado + 5 segundos
- El procesamiento de métricas es asíncrono (proceso generado) para evitar bloquear el flujo de mensajes
- El módulo opera independientemente de los módulos de enrutamiento y transformación

---

Módulos de Seguridad y Control

El DRA se envía con tres módulos independientes y autoconfigurables para seguridad de interconexión y control de roaming. Cada uno tiene su propia guía de operaciones dedicada:

- **Seguridad Diameter** — Protección alineada con GSMA FS.19/FS.21: el Firewall Diameter (verificaciones de formato de bajo nivel más filtrado de Categoría 1/2/3), Ocultación de Topología (eliminación de Route-Record y reescritura de Origen), Limitación de Tasa por par y Sanitización de AVP.

- **Control de Roaming (SoR)** — Dirige a suscriptores de roaming hacia VPLMNs preferidos al rechazar Solicitudes de Actualización de Ubicación que provienen de redes no preferidas.
- **Búsqueda de Suscriptores (SLF)** — Aprende dinámicamente las vinculaciones de suscriptores a nodos de servicio a partir del tráfico Diameter y enruta solicitudes SLg/SLh al nodo de servicio correcto.

Métricas de Prometheus

El DRA expone métricas completas de Prometheus para monitorear tráfico Diameter, salud de pares y operaciones de módulos. Todas las métricas están disponibles en el endpoint `/metrics`.

Métricas Core Diameter

Estado de Pares

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

- `origin_host` - Identidad Diameter del par
- `ip` - Dirección IP del par

Ejemplo:

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

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

Conteo de Mensajes

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

- `origin_host` - Identidad Diameter del par
- `received_from` - Par desde el cual se recibió el mensaje
- `application_id` - ID de Aplicación Diameter (ver Referencia de ID de Aplicación)
- `cmd_code` - Código de Comando Diameter (ver Códigos de Comando Comunes)
- `application_name` - Nombre de aplicación legible por humanos (por ejemplo, "3GPP_S6a")
- `cmd_name` - Nombre de comando legible por humanos (por ejemplo, "AIR")
- `direction` - "request" o "response"

Ejemplo:

```
# Tasa de solicitud AIR S6a desde 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]))
```

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**:

- `origin_host` - Solicitante original
- `routed_to` - Par que envió la respuesta
- `application_id` - ID 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

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:

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

Retraso de Respuesta

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

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

Ejemplo:

```
# Tiempo promedio de respuesta desde HSS
avg(diameter_peer_last_response_delay{routed_to="hss01.example.com"})

# Tiempo de respuesta P95 para S6a
histogram_quantile(0.95,
  rate(diameter_peer_last_response_delay{application_name="3GPP_S6a"}
[5m])
)
```

Solicitudes No Respondidas

Métrica: `diameter_peer_unanswered_request_count_total` **Tipo**: Counter **Descripción**: Solicitudes enviadas pero no respondidas dentro del período de tiempo de espera **Etiquetas**:

- `origin_host` - Solicitante original
- `routed_to` - Par que no respondió

Dirección de Roaming (SoR)

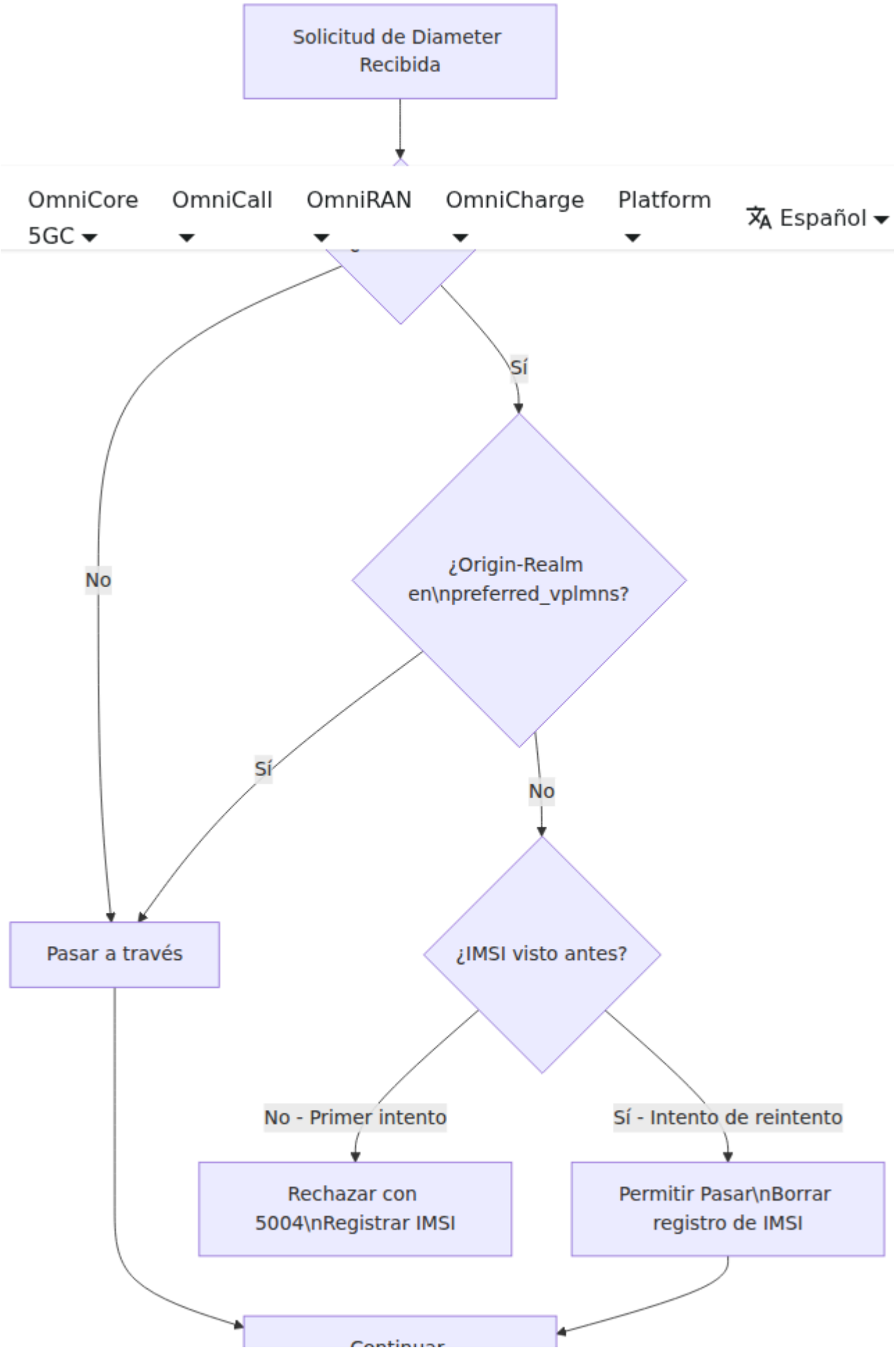
La Dirección de Roaming (SoR) permite al HPLMN influir en qué VPLMN se conecta un suscriptor en roaming. Cuando un suscriptor intenta registrarse a través de un VPLMN no preferido, el DRA intercepta la solicitud de actualización de ubicación S6a (ULR) y la rechaza con **DIAMETER_ERROR_ROAMING_NOT_ALLOWED (5004)**. Esto provoca que el UE se desasocie y se vuelva a asociar, idealmente seleccionando un VPLMN preferido.

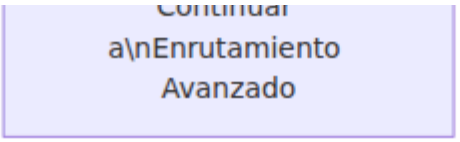
Si la cobertura preferida no está disponible y el suscriptor regresa al mismo VPLMN no preferido, el módulo permite la solicitud en el siguiente intento.

SoR está definido en 3GPP TS 29.272 Sección 5.2.1.1 y 3GPP TS 23.122.

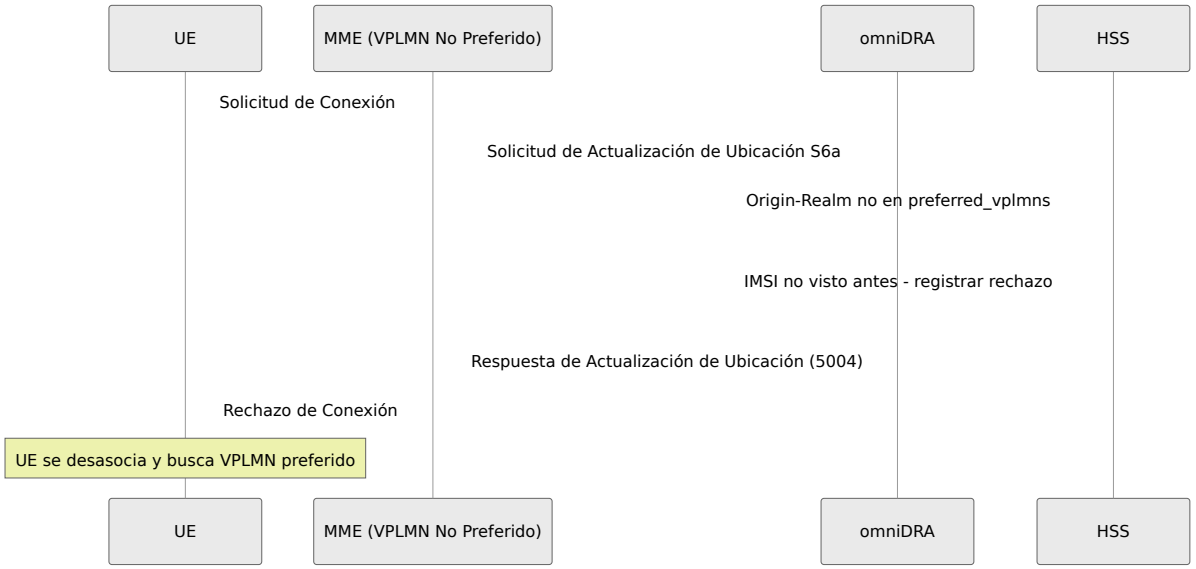
Cómo Funciona

El módulo opera dentro de la tubería de procesamiento de solicitudes del DRA, ejecutándose **antes** del Enrutamiento Avanzado. Solo se evalúan las solicitudes de actualización de ubicación S6a (Application-Id `16777251`, Command-Code `316`). Todos los demás mensajes de Diameter pasan sin modificaciones.

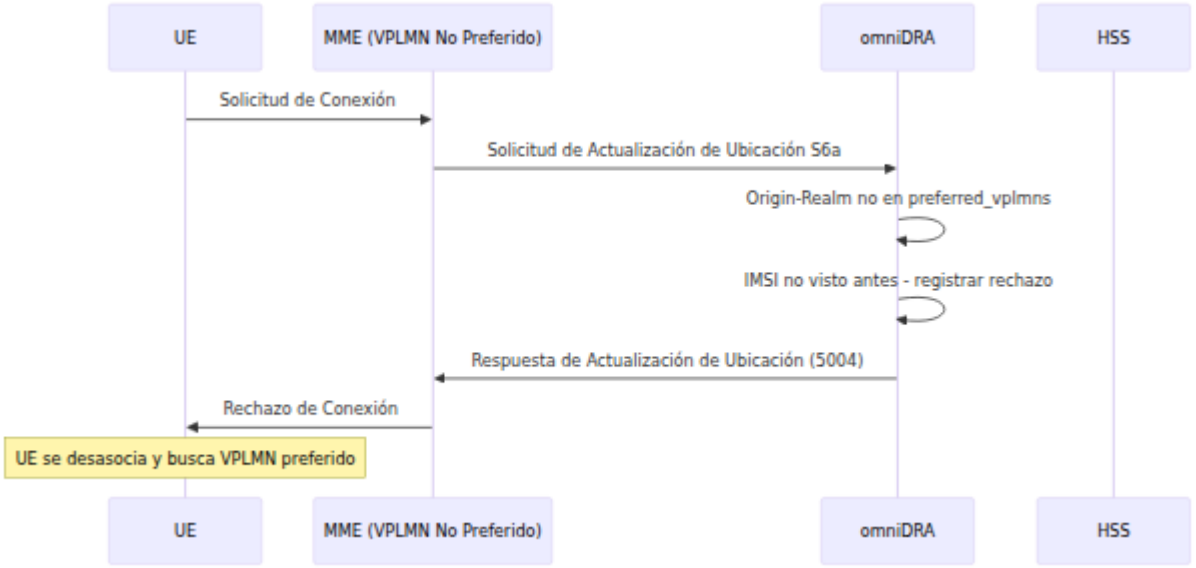

Solicitud de Diameter Recibida
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
Sí
No
¿Origin-Realm en\npreferred_vplmns?
Sí
No
Pasar a través
¿IMSI visto antes?
No - Primer intento
Sí - Intento de reintento
Rechazar con 5004\nRegistrar IMSI
Permitir Pasar\nBorrar registro de IMSI



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, el

Enrutamiento Avanzado nunca se alcanza para ese mensaje.

Seguimiento de Suscriptores

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

Cuando un IMSI rastreado ha alcanzado el umbral de `max_rejections` y envía otra ULR, la solicitud se permite y el registro de seguimiento se elimina.

Si no llega ninguna ULR adicional dentro de la ventana TTL, el registro expira y se elimina durante el siguiente ciclo de limpieza. Una ULR posterior del mismo IMSI se tratará como un primer intento y será rechazada nuevamente.

Configuración

El módulo se configura 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
  preferred_vplmns: [
    "epc.mnc001.mcc001.3gppnetwork.org",
    "epc.mnc002.mcc001.3gppnetwork.org"
  ]
}
```

Parámetros

Parámetro	Tipo	Requerido	Predeterminado	
`enabled`	Booleano	Sí	`false`	Ha Di las
`max_rejections`	Entero	No	`1`	Nú su va se
`rejection_ttl_seconds`	Entero	No	`300`	Tie re su es co int
`rejection_result_code`	Entero	No	`5004`	Có en co DI se
`preferred_vplmns`	Lista	No	`[]`	Lis ide pr co ev

Ejemplos de Configuración

Rechazar Una Vez, Luego Permitir

El comportamiento predeterminado. Los suscriptores que llegan a un VPLMN no preferido son rechazados una vez. Si regresan, la ULR se reenvía 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 que llega de un MME en `epc.mnc003.mcc310.3gppnetwork.org` (no en la lista preferida) es rechazada con el Código de Resultado 5004. El UE se desasocia e intenta encontrar una red preferida. Si regresa al mismo VPLMN no preferido dentro de 300 segundos, la ULR se reenvía normalmente.

Caso de uso: Dirección de roaming estándar donde el operador tiene acuerdos de roaming preferido con redes asociadas específicas.

Múltiples Rechazos Antes de Permitir

Para escenarios donde el operador desea 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 suscriptor es rechazado tres veces antes de ser permitido en el cuarto intento. El TTL se extiende a 600 segundos para acomodar los ciclos de reintento adicionales.

Caso de uso: Áreas urbanas densas donde existe cobertura preferida pero puede requerir múltiples intentos de conexión para adquirir.

Métricas

Rechazos

Métrica: `diameter.roaming_steering.reject.count` **Tipo**: Contador **Descripción**: Incrementado cada vez que una ULR es rechazada por el módulo SoR. **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

Permitido (VPLMN Preferido)

Métrica: `diameter.roaming_steering.allow.count` **Tipo**: Contador **Descripción**: Incrementado cuando una ULR es permitida. **Etiquetas**:

- `origin_realm` - Origin-Realm del VPLMN de origen
- `reason` - Por qué se 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**: Incrementado cuando el procesador devuelve un mensaje de respuesta de error debido a 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: Los suscriptores en VPLMNs no preferidos son rechazados repetidamente y nunca se conectan con éxito.

Causas posibles:

- `rejection_ttl_seconds` es demasiado corto, lo que provoca que el registro de seguimiento expire antes de que el suscriptor vuelva a intentar
- `max_rejections` está configurado más alto que el número de reintentos que realiza el UE antes de rendirse

Resolución:

1. Aumentar `rejection_ttl_seconds` para acomodar el tiempo de reintento del UE
2. Verificar que `max_rejections` esté configurado a un valor que el UE pueda alcanzar realísticamente dentro de su ciclo de reintento

Suscriptores en VPLMNs Preferidos Siendo Rechazados

Síntomas: Las ULR de redes asociadas preferidas están siendo rechazadas con 5004.

Causas posibles:

- El Origin-Realm del VPLMN preferido no está listado en `preferred_vplmns`
- La cadena de Origin-Realm no coincide exactamente (sensible a mayúsculas y minúsculas)

Resolución:

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

Módulo No Efectivo

Síntomas: Las ULR de VPLMNs no preferidos se están reenviando sin rechazo.

Causas posibles:

- `enabled` está configurado como `false`
- El proceso del módulo no está en ejecución (verificar supervisor)

Resolución:

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

Referencia

Códigos de Diameter

Código	Nombre	Descripción	Ref
5004	DIAMETER_ERROR_ROAMING_NOT_ALLOWED	Suscriptor no permitido para hacer roaming en este VPLMN	3GI 29. Sec 7.4

Comandos S6a

Command-Code	Nombre	Descripción	Referencia
316	Solicitud/Respuesta de Actualización de Ubicación (ULR/ULA)	Enviado por el MME al HSS durante la conexión para actualizar el nodo de servicio del suscriptor	3GPP TS 29.272 Sección 7.2.3

IDs de Aplicación

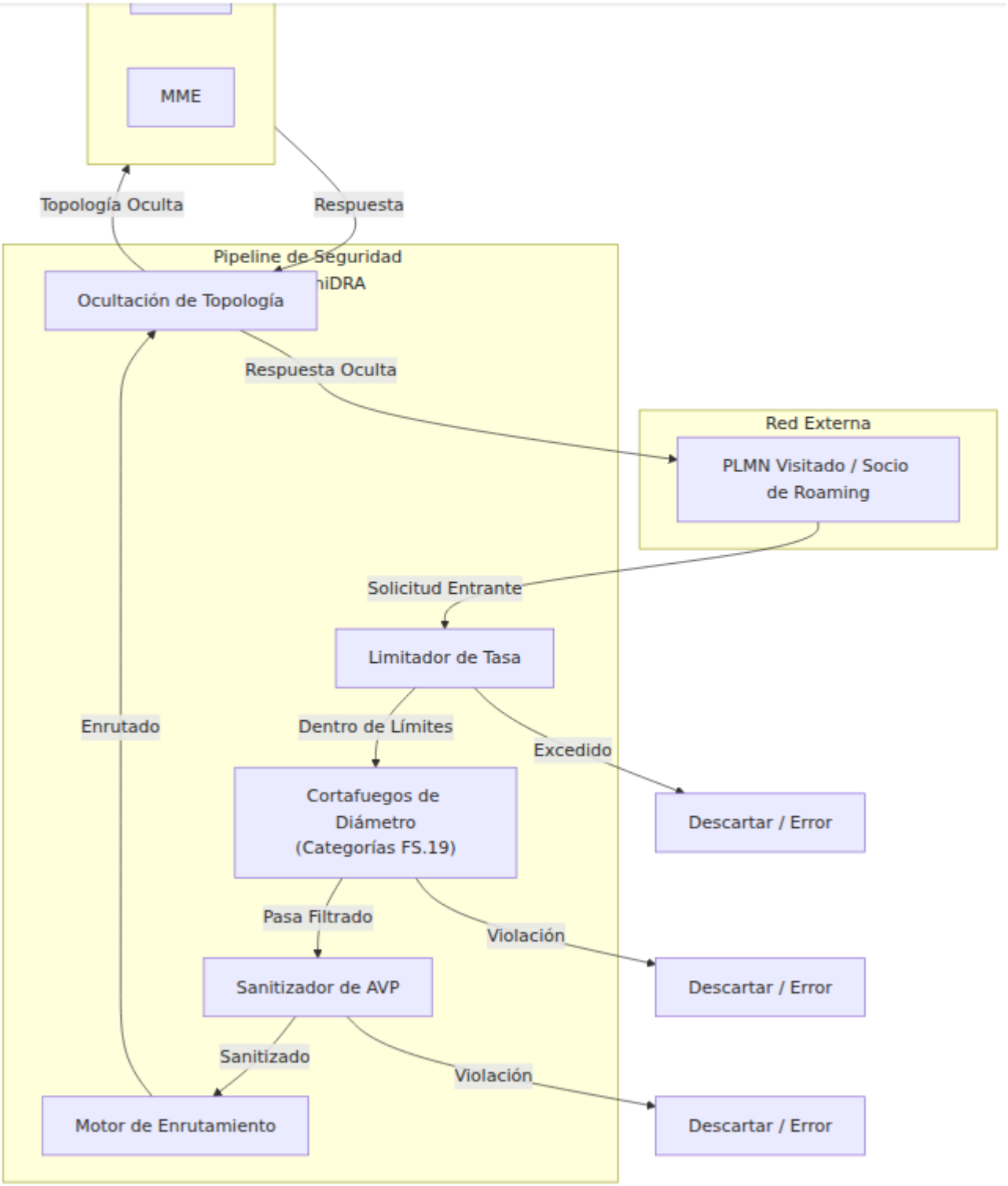
ID	Interfaz	Descripción	Referencia
16777251	S6a/S6d	Autenticación y gestión de suscripciones de MME/SGSN a HSS	3GPP TS 29.272

Seguridad de Diámetro

OmniDRA proporciona un conjunto integral de módulos de seguridad de Diámetro alineados con **GSMA FS.19** (Seguridad de Interconexión de Diámetro, 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 Diámetro.

Cada módulo es configurable de forma independiente, puede ser habilitado o deshabilitado sin afectar a los demás, y emite sus propios eventos de telemetría para monitoreo y alertas.

Visión General de la Arquitectura



Orden de Procesamiento

Las solicitudes entrantes de Diámetro pasan a través de los módulos de seguridad en el siguiente orden. Un mensaje rechazado en cualquier etapa nunca se reenvía a etapas posteriores.

Orden	Módulo	Dirección	Propósito
1	Limitador de Tasa	Entrante	Protección contra inundaciones volumétricas
2	Cortafuegos de Diámetro	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 Diámetro
5	Ocultación de Topología	Saliente	Ocultación de la topología de la red

Acciones

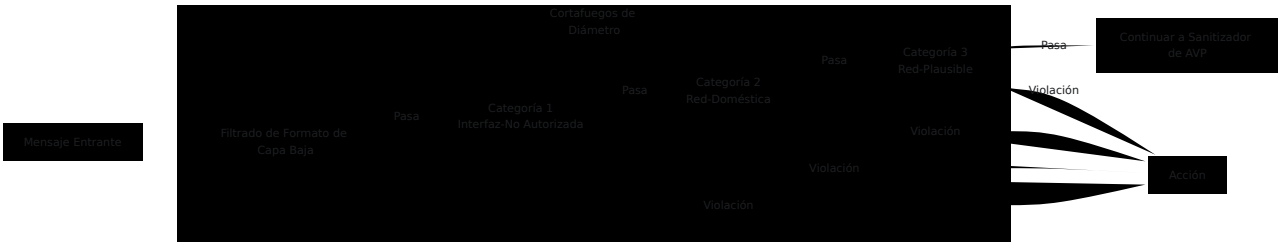
Todos los módulos de seguridad admiten acciones configurables cuando se detecta una violación. Las acciones se configuran por módulo (y por categoría para el Cortafuegos de Diámetro).

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

---

Cortafuegos de Diámetro

El Cortafuegos de Diámetro implementa las cuatro capas de filtrado definidas en **GSMA FS.19 Sección 3.3** y la categorización de paquetes independiente del protocolo de **FS.21 Sección 7**. Los mensajes se evalúan a través de cada capa en orden; una violación en cualquier capa detiene el procesamiento adicional.



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 sin necesidad de entender la semántica de aplicación de capas superiores. Esta capa captura mensajes malformados 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	Valida que el Session-Id (AVP 263) sea el primer AVP en el mensaje, según RFC 6733 Sección 8.8.
Destination-Host en Respuestas	Sección 3.3.4	Rechaza mensajes de respuesta que contengan un AVP Destination-Host, lo cual es una violación de protocolo que puede indicar evasión de filtro.
Validación de AVP Obligatorios	Sección 3.3.5	Valida que los AVPs requeridos estén presentes para cada código de comando (por ejemplo, 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 Interfaz-No Autorizada

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

El filtrado de Categoría 1 asegura que solo se acepten IDs de Aplicación de Diámetro y Códigos de Comando autorizados en cada interfaz de interconexión. Esto previene el acceso externo a interfaces solo internas (por ejemplo, bloqueando comandos Sh a través de una interfaz S6a) y hace cumplir que los socios de roaming solo envíen tipos de mensajes cubiertos por sus acuerdos de roaming.

El enfoque de lista blanca sigue la recomendación de FS.19: **bloquear todos los mensajes de Diámetro excepto aquellos explícitamente requeridos para una interfaz dada**.

Las listas blancas pueden configurarse globalmente (aplicando a todos los pares) o por par, permitiendo 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	Solicitud/Respuesta de Actualización de Ubicación	MME → HSS	3GPP TS 29.272 §7.2.3
317	Solicitud/Respuesta de Cancelación de Ubicación	HSS → MME	3GPP TS 29.272 §7.2.7
318	Solicitud/Respuesta de Información de Autenticación	MME → HSS	3GPP TS 29.272 §7.2.5
319	Solicitud/Respuesta de Inserción de Datos de Suscriptor	HSS → MME	3GPP TS 29.272 §7.2.9
320	Solicitud/Respuesta de Eliminación de Datos de Suscriptor	HSS → MME	3GPP TS 29.272 §7.2.11
321	Solicitud/Respuesta de Purga de UE	MME → HSS	3GPP TS 29.272 §7.2.13
323	Solicitud/Respuesta de Notificación	MME → HSS	3GPP TS 29.272 §7.2.15

IDs de Aplicación Comunes para interconexión 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-Doméstica

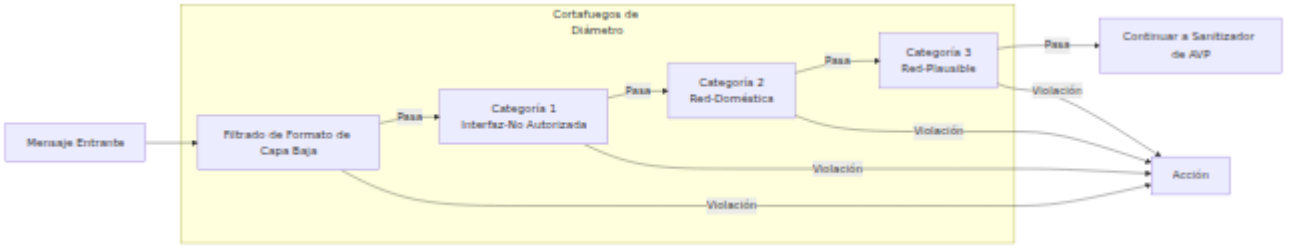
Referencia GSMA: FS.19 Sección 3.3.6, FS.21 Sección 7.2.2

El filtrado de Categoría 2 protege a los suscriptores domésticos de ser atacados por mensajes que llegan desde la interconexión. Los mensajes en códigos de comando protegidos se inspeccionan para identificar la identidad del suscriptor (IMSI en el AVP User-Name, MSISDN en el AVP 701). Si la identidad coincide con un prefijo de suscriptor doméstico, el mensaje es rechazado; el tráfico legítimo para suscriptores domésticos debe originarse desde dentro de la red doméstica, no desde pares externos.

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

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

category_2: %{
  enabled: true,
  action: {:error, 3002},
  # Códigos de comando 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 rastreando la última red vista para cada suscriptor (IMSI). Cuando llega una Solicitud de Actualización de Ubicación desde una red visitada diferente a la anterior, el tiempo transcurrido se compara con un umbral configurable. Un cambio de dominio que ocurre más rápido de lo físicamente posible indica un posible ataque (por ejemplo, actualizaciones de ubicación falsificadas).

Este módulo mantiene un seguimiento estatal por IMSI utilizando una tabla ETS con limpieza automática de entradas obsoletas (las entradas más antiguas de 24 horas se eliminan 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 cambios de dominio que ocurren dentro de una ventana de tiempo mínima 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

El Cortafuegos de Diámetro se habilita a nivel superior, con cada capa de filtrado configurada de forma independiente. Los fragmentos de configuración mostrados anteriormente dentro de cada sección de categoría se combinan 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	Descripción
`enabled`	Booleano	Sí	`false`	Habilitar o deshabilitar todo el módulo de Cortafuegos de Diámetro. Cuando `false`, todos los mensajes pasan sin inspección.
`home_imsi_prefixes`	Lista	No	`[]`	Lista de cadenas de prefijos de IMSI que identifican a los suscriptores domésticos. Utilizado por el filtrado de Categoría 2. Ejemplo: `["31338", "31339"]`.
`home_msisdn_prefixes`	Lista	No	`[]`	Lista de cadenas de prefijos de MSISDN que

Parámetro	Tipo	Requerido	Por Defecto	Descripción
				identifican a los suscriptores domésticos. Utilizado por el filtrado de Categoría 2. Ejemplo: `["+1313"]`.

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 violación de Categoría 1 responde con DIAMETER_APPLICATION
`whitelists`	Mapa	Sí	-	Mapa de nombre de hos `:all`) a combinaciones de Aplicación / Código d clave `:all` proporciona por defecto. Las entrada pares tienen prioridad.

Parámetros de Categoría 2

Parámetro	Tipo	Requerido	Por Defecto	Descrip
`enabled`	Booleano	No	`true`	Habilitar deshabili filtrado d Categorí
`action`	Acción	No	`{:error, 3002}`	Acción a tomar cu un suscri doméstic atacado desde la intercone
`protected_command_codes`	Lista	No	`[]`	Códigos comando activan verificaci de identi de suscri doméstic Típicame los códig de coma S6a que 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]`	Se detectó y bloqueó una violación de formato de capa baja
`[:diameter, :firewall, :category_1, :block]`	Se detectó y bloqueó una violación de Categoría 1
`[:diameter, :firewall, :category_2, :block]`	Se detectó y bloqueó una violación de Categoría 2
`[:diameter, :firewall, :category_3, :block]`	Se detectó y bloqueó una violación de Categoría 3
`[:diameter, :firewall, :pass, :count]`	El mensaje pasó todas las verificaciones del cortafuegos

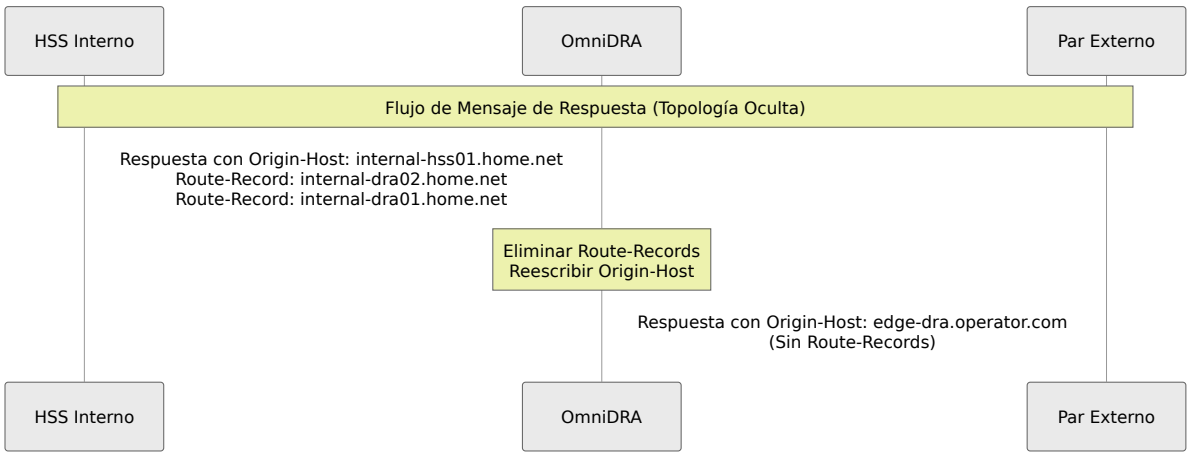
Todos los eventos incluyen metadatos: `origin_host`, `application_id`, `command_code`, `application_name`, `command_name` y `reason`.

Ocultación de Topología

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

El módulo de Ocultación de Topología previene que la topología interna de la red sea expuesta a pares externos. Cuando los mensajes de Diámetro atraviesan múltiples nodos internos, acumulan AVPs de Route-Record y llevan valores de Origin-Host/Origin-Realm que revelan nombres de host internos, estructura de red y conteos de nodos. Esta información puede ser utilizada por atacantes para mapear la red interna para ataques dirigidos.

La ocultación de topología opera en la **ruta saliente** — después de que se han tomado decisiones de enrutamiento, antes de que los mensajes sean reenviados al par de destino.



Características

Característica	Referencia FS.19	Descripción
Eliminación de Route-Record	Sección 2.4	Elimina todos los AVPs de Route-Record (282) que revelan rutas 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 los 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 la ocultación de topología de manera selectiva — solo a pares externos, o a todos los pares

```
module_topology_hiding: %{
  enabled: true,
  # Eliminar AVPs de 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 Ocultación de Topología.
`strip_route_records`	Booleano	No	`true`	Eliminar todo AVPs de Rout Record (códig 282) de los mensajes ant de reenviarlo pares externo
`rewrite_origin_host`	Mapa	No	Ver abajo	Controla la reescritura de Origin-Host e mensajes de respuesta.
`rewrite_origin_realm`	Mapa	No	Ver abajo	Controla la reescritura de Origin-Realm los mensajes respuesta.
`external_peers`	`:all` o Lista	No	`:all`	Pares considerados externos. La ocultación de topología sol aplica a mens

Parámetro	Tipo	Requerido	Por Defecto	Descripci
				destinados a estos pares. Utilice `:all` implementac de intercone Utilice una lis de nombres host para una aplicación selectiva.

Parámetros de Reescritura (aplican tanto a `rewrite_origin_host` como a `rewrite_origin_realm`):

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

Eventos de Telemetría

Evento	Descripción
`[:diameter, :topology_hiding, :route_record, :stripped]`	Se eliminaron AVPs de Route-Record. La medición incluye `count` de AVPs eliminados.
`[:diameter, :topology_hiding, :origin_host, :rewritten]`	AVP Origin-Host reescrito
`[:diameter, :topology_hiding, :origin_realm, :rewritten]`	AVP Origin-Realm reescrito

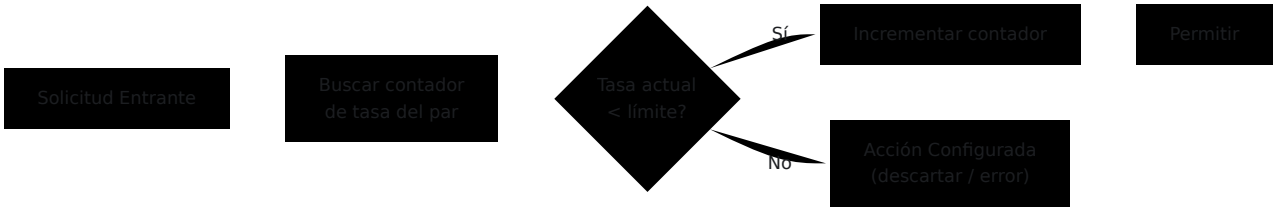
Limitador de Tasa

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

El Limitador de Tasa aplica límites de tasa de mensajes por par para proteger contra ataques volumétricos y inundaciones de mensajes. Opera como la primera verificación de seguridad en el pipeline — antes de cualquier análisis de mensaje o inspección de contenido — para eliminar carga excesiva lo antes posible.

Los límites de tasa se rastrean por par utilizando 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},
  # Anulaciones 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`
`default_max_requests_per_second`	Entero	No	`1000`
`default_action`	Acción	No	`{:error, 3004}`
`peer_limits`	Mapa	No	`%{}`

Parámetros de Anulación por Par:

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

Eventos de Telemetría

Evento	Descripción
`[:diameter, :rate_limiter, :throttled]`	Un mensaje fue limitado por tasa. 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 AVPs de Diámetro en la frontera de interconexión. Aborda las recomendaciones de la Sección 3.6 de FS.21 para

manejar 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 Anidamiento de AVP Agrupados	FS.21 Sección 3.6	Aplica una profundidad máxima de anidamiento para AVPs agrupados. Previene ataques de desbordamiento de pila utilizando 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 una vio profun anidam elimina vende silenci AVPs o bloque
`allowed_vendor_ids`	Lista	No	`[0, 10415, 13019, 5535]`	Lista d vende permit interco AVPs d vende elimina
`strip_unknown_vendor_avps`	Booleano	No	`true`	Habilit elimina de ven no esta de `allowe`

Parámetro	Tipo	Requerido	Por Defecto	De
`max_avp_nesting_depth`	Entero	No	`10`	Profun permit anidam agrupa mensa exceda sujetos configu

IDs de Vendedores Comunes:

ID de Vendedor	Organización	Notas
0	IETF	AVPs de Diámetro estándar 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 CDMA)

Eventos de Telemetría

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

---

Recomendaciones de Implementación

Orden de Habilitación Recomendado

Al implementar los módulos de seguridad por primera vez, habilítalos de manera incremental para evitar interrumpir el tráfico en vivo:

1. **Limitador de Tasa** — Comienza con límites generosos y monitorea patrones de tráfico. Ajusta los límites una vez que se entiendan las tasas base.
2. **Sanitizador de AVP** — Bajo riesgo de falsos positivos. Habilita la eliminación y las verificaciones de anidamiento.
3. **Cortafuegos de Diámetro (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 Diámetro (Capa Baja)** — Habilita las verificaciones de conformidad de protocolo.
5. **Cortafuegos de Diámetro (Categoría 2)** — Configura prefijos de IMSI/MSISDN domésticos.

6. **Ocultación de Topología** — Habilita primero la eliminación de Route-Record, luego la reescritura de Origin-Host.
7. **Cortafuegos de Diámetro (Categoría 3)** — Habilita con acción `:log_only` para observar patrones de ubicación antes de hacer cumplir.

Defensa en Profundidad

Estos módulos implementan el principio de **defensa en profundidad** descrito en 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 Diámetro (todas las categorías)
Abuso de Interfaz	Categoría 1	Sanitizador de AVP
Objetivo de Suscriptor Doméstico	Categoría 2	Categoría 1 (restricción de interfaz)
Suplantación de Ubicación	Categoría 3	Categoría 2 (verificación de suscriptor doméstico)
Descubrimiento de Topología	Ocultación 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
Ocultación 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 (Independiente del Protocolo)	Anexo A	Sección 7

Función de Búsqueda de Suscriptores (SLF)

La Función de Búsqueda de Suscriptores (SLF) aprende dinámicamente qué elemento de red está sirviendo a cada suscriptor al observar el tráfico de señalización de Diameter que pasa a través del DRA. Utiliza estos enlaces aprendidos para enrutar solicitudes posteriores — como consultas de servicio de ubicación — directamente al nodo de servicio correcto sin requerir reglas de enrutamiento estáticas.

Esto es particularmente útil para solicitudes de servicio de ubicación SLg/SLh, donde el GMLC necesita alcanzar el MME de servicio para un suscriptor dado, pero no tiene conocimiento previo de cuál es ese MME.

El concepto de SLF se describe en 3GPP TS 29.172 y 3GPP TS 29.173 para servicios de ubicación, con el registro de suscriptores definido en 3GPP TS 29.272.

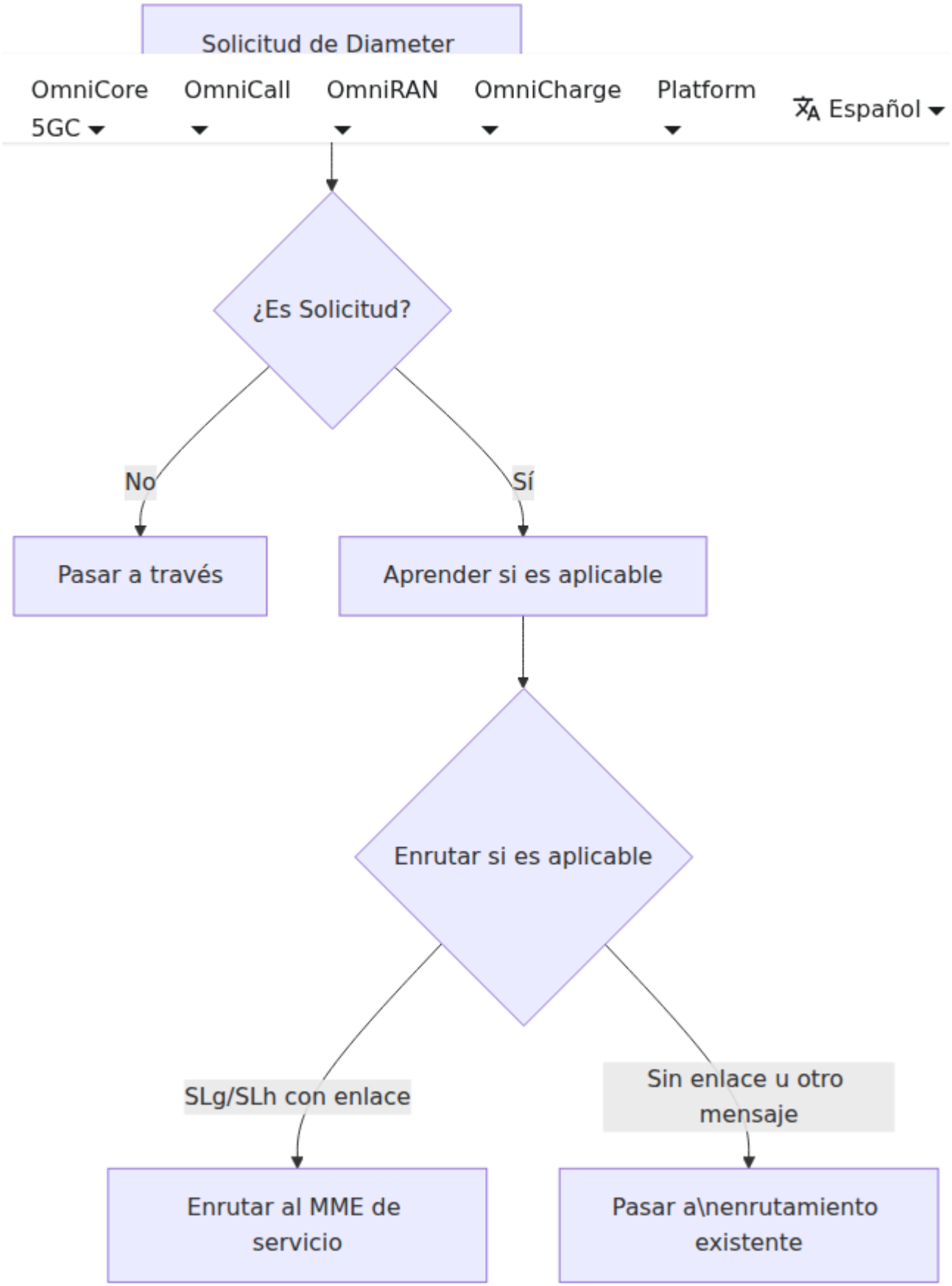
Cómo Funciona

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

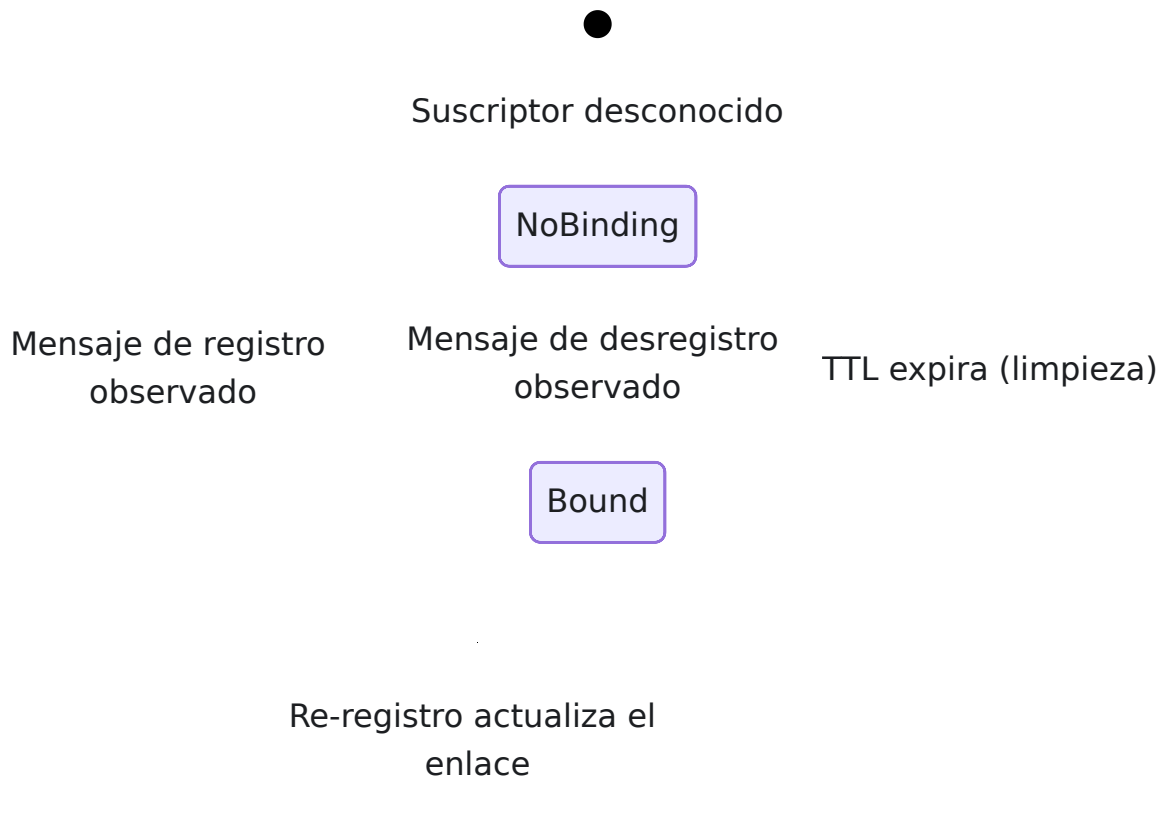
Durante el **aprendizaje**, el módulo observa pasivamente los mensajes de registro y desregistro que fluyen a través del DRA. Cuando un suscriptor se registra (por ejemplo, a través de una solicitud de actualización de ubicación S6a), el módulo registra qué elemento de red está ahora sirviendo a ese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

Durante el **enrutamiento**, cuando llega una solicitud que necesita 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 pasa a la lógica de enrutamiento existente (incluido el Enrutamiento Avanzado).

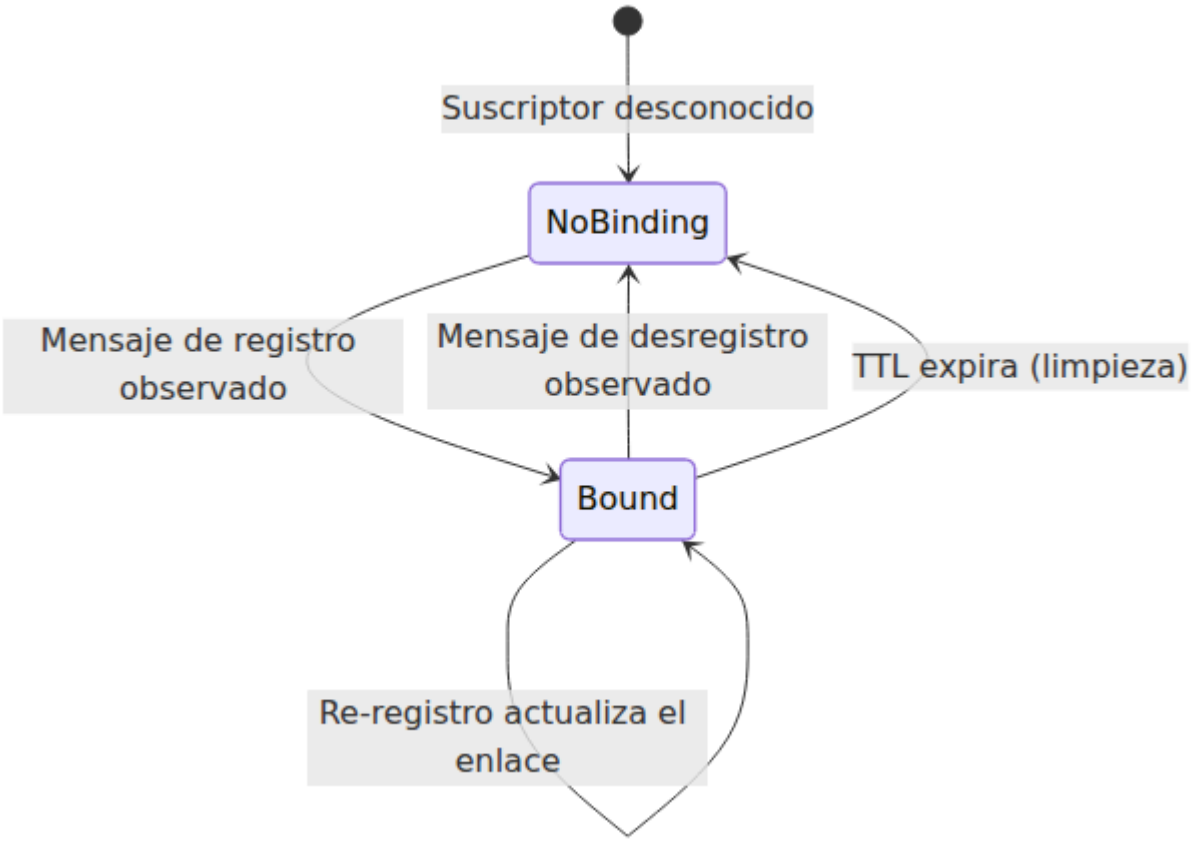

Solicitud de Diameter
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
¿Es Solicitud?
No
Sí
Pasar a través
Aprender si es aplicable
Enrutar si es aplicable
SLg/SLh con enlace
Sin enlace u otro mensaje
Enrutar al MME de servicio
Pasar a\nenrutamiento existente

Ciclo de Vida del Enlace



Aprendizaje: Mensajes que Crean Enlaces

El módulo aprende enlaces de suscriptor a nodo de servicio de los siguientes mensajes de 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 Id 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 Eliminan Enlaces

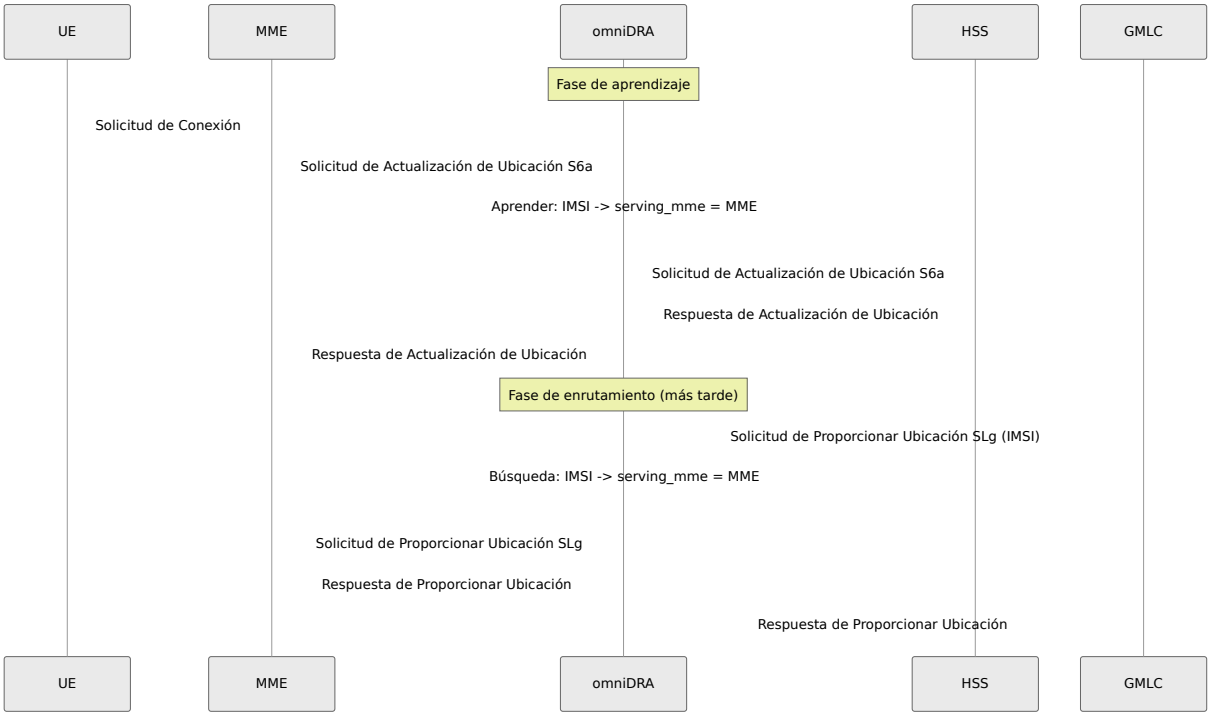
Interfaz	Mensaje	Código de Comando	Enlace Eliminado
S6a	Solicitud de Cancelación de Ubicación (CLR)	317	`serving_mme`
S6a	Solicitud de Purga de UE (PUR)	321	`serving_mme`
Gx	Solicitud de Control de Crédito de Terminación (CCR-T)	272 (CC-Request-Type=3)	`serving_pgw`
Cx	Solicitud de Asignación de Servidor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

Cuando se elimina el último enlace para un suscriptor, todo el registro del suscriptor se elimina 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`

Flujo de Ejemplo: Solicitud de Servicio de Ubicación



Posición en la Canalización

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

Configuración

El módulo se configura bajo `module_subscriber_lookup` en `config/runtime.exs`.

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

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

Parámetros

Parámetro	Tipo	Requerido	Predeterminado	Des
`enabled`	Booleano	Sí	`false`	Habi desh módu Búsq Susc Cuar toda solici pasa modi y no apre enla
`binding_ttl_seconds`	Entero	No	`86400`	Tiem segu de qu enla susc expir elimi la lim pred es 24 limpi ejecu mita inter TTL, míni segu

Ejemplos de Configuración

Implementación Estándar

Adecuado 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. Las solicitudes SLg y SLh se enrutan automáticamente al MME de servicio correcto. Los enlaces se actualizan cada vez que se observa un nuevo mensaje de registro para el mismo suscriptor.

Caso de uso: Redes con requisitos de servicio de ubicación (LCS) donde el GMLC necesita alcanzar el MME de servicio sin mapeos estáticos de suscriptor a MME.

Entorno de Alta Rotación

Para redes con movilidad frecuente de suscriptores donde los enlaces obsoletos deben expirar rápidamente.

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

Cómo funciona: Los enlaces expiran después de 1 hora. Esto es apropiado cuando los suscriptores se mueven frecuentemente entre MMEs y 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**: Incrementado cada vez que se 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**: Incrementado cada vez que se elimina explícitamente un enlace de suscriptor a través de un mensaje de desregistro. **Etiquetas**:

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

Solicitudes Enrutadas

Métrica: `diameter.subscriber_lookup.route.count` **Tipo**: Contador **Descripción**: Incrementado cada vez que una solicitud SLg/SLh se enruta con éxito utilizando un enlace aprendido. **Etiquetas**:

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

Solución de Problemas

Solicitudes de Ubicación No Enrutadas al MME de Servicio

Síntomas: Las solicitudes de Proporcionar Ubicación SLg o las solicitudes de Información de Enrutamiento LCS SLh no se están enrutando al MME esperado, pasando a la enrutación predeterminada 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 (TTL excedido)
- El par MME de servicio no está conectado a este DRA
- `enabled` está configurado como `false`

Resolución:

1. Verifique que el módulo esté habilitado en la configuración
2. Asegúrese de que el tráfico ULR S6a para el suscriptor fluya a través de este DRA (los enlaces solo se aprenden del tráfico observado)
3. Verifique que `binding_ttl_seconds` sea lo suficientemente largo para cubrir el intervalo entre el registro y la solicitud de ubicación
4. Confirme que el MME de servicio esté 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 (verifique el supervisor)
- Los mensajes no contienen un IMSI válido en el AVP esperado

Resolución:

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

Enlaces Obsoletos Causando Solicitudes Mal Enrutadas

Síntomas: Las solicitudes de ubicación se enrutan a un MME que ya no está sirviendo 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. Reduzca `binding_ttl_seconds` para que coincida con el intervalo de re-registro esperado del suscriptor
2. Asegúrese de 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 cargos 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 ubicación 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 cargos 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 sirve 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 sirve el registro IMS del suscriptor (reservado para enrutamiento futuro)