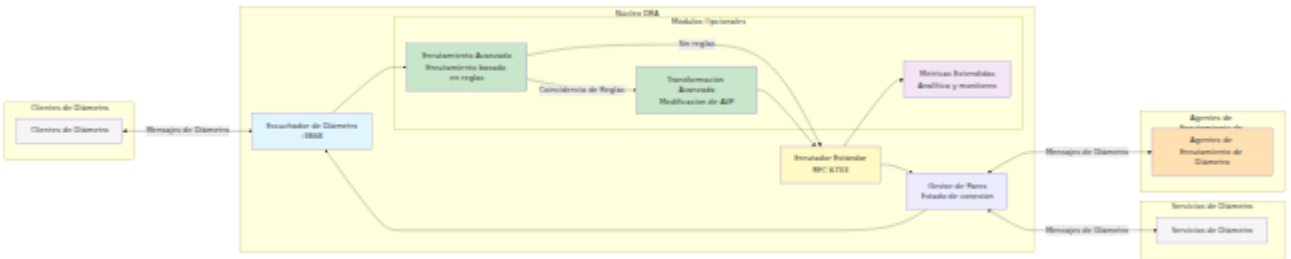
Guía de Operaciones DRA

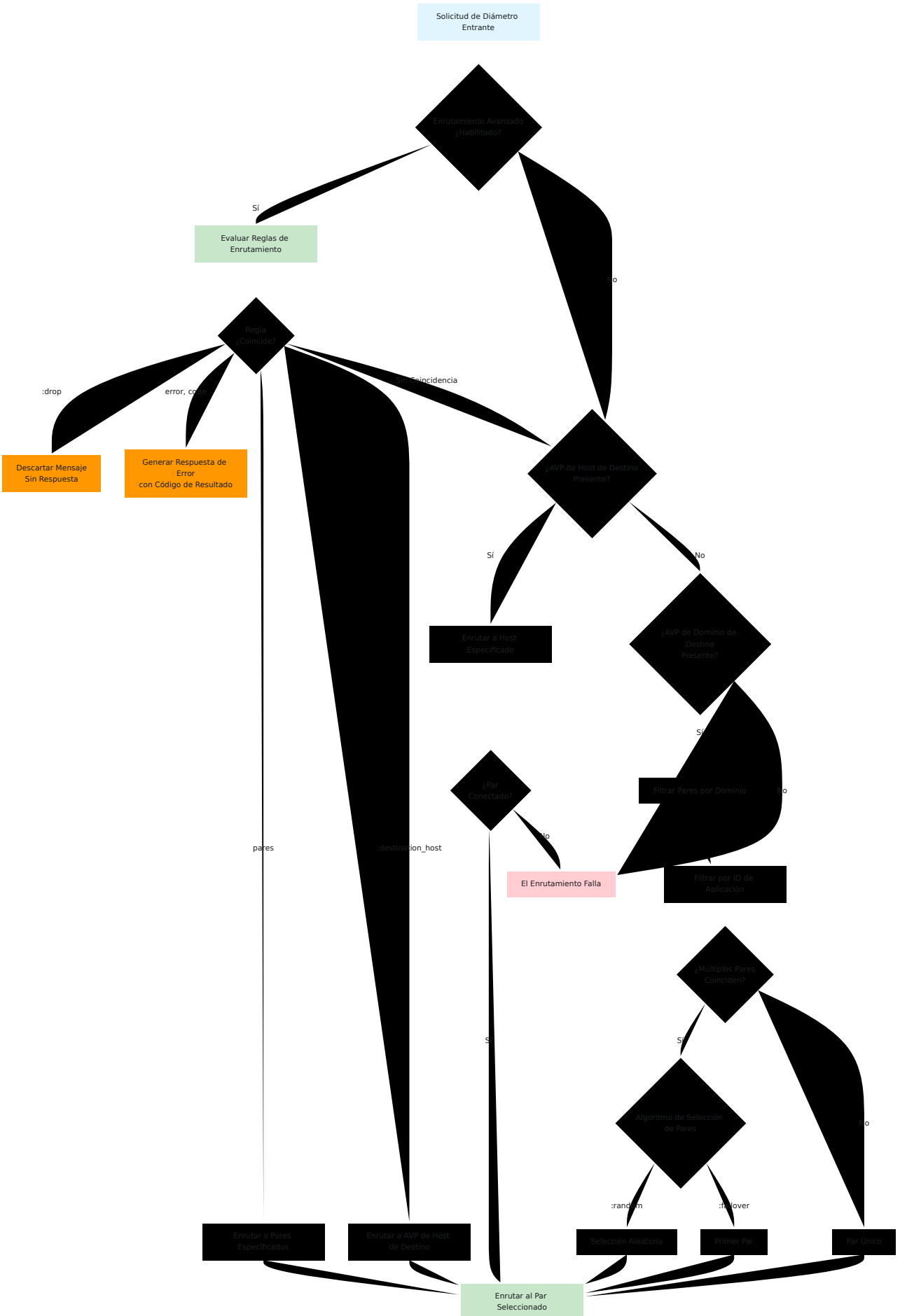
Tabla de Contenidos

Descripción General de la Arquitectura DRA



Enrutamiento de Diámetro Estándar

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


Solicitud de Diámetro Entrante
Enrutamiento Avanzado ¿Habilitado?
Sí
Evaluar Reglas de Enrutamiento
Regla ¿Coincide?
:drop
Descartar Mensaje Sin Respuesta
Generar Respuesta de Error con Código de Resultado
¿AVP de Host de Destino Presente?
Sí
Enrutar a Host Especificado
No
¿AVP de Dominio de Destino Presente?
Filtrar Pares por Dominio
¿Par Conectado?
El Enrutamiento Falla
Filtrar por ID de Aplicación
pares
:destination_host
¿Múltiples Pares Coinciden?
Algoritmo de Selección de Pares
:random
Selección Aleatoria
Primer Par
Par Único
Enrutar a Pares Especificados
Enrutar a AVP de Host de Destino
Enrutar al Par Seleccionado

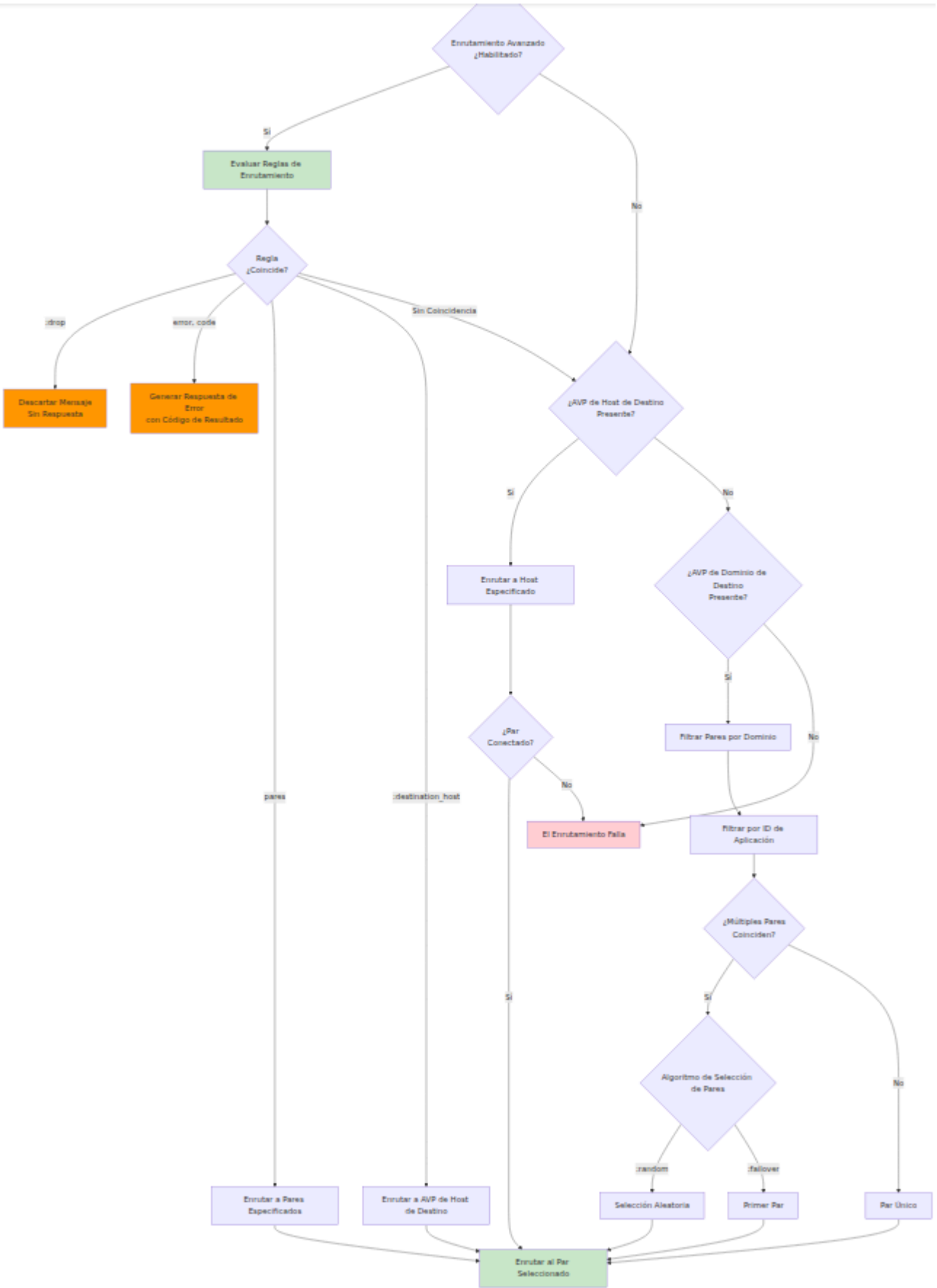
Enrutamiento de Solicitudes

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

1. **AVP de Host de Destino (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 de Dominio de Destino (283)** - Si el Host de Destino está ausente, enruta basado en el dominio
    - El DRA selecciona un par conectado que anuncia soporte para el dominio objetivo
    - Se aplica balanceo de carga cuando múltiples pares coinciden con el dominio
    - El enrutamiento basado en el dominio permite flexibilidad entre múltiples hosts
3. **ID de Aplicación** - Los pares se filtran por aplicaciones de Diámetro soportadas
    - Solo se consideran los pares que anuncian soporte para el ID de Aplicación del mensaje
    - Basado en el Intercambio de Capacidades (CER/CEA) durante el establecimiento de conexión del par
    - 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 Avanzado ¿Habilitado?
Si
No
Evaluar Reglas de Enrutamiento
Regla ¿Coincide?
:drop
error, code
Sin Coincidencia
Descartar Mensaje Sin Respuesta
Generar Respuesta de Error con Código de Resultado
¿AVP de Host de Destino Presente?
Si
No
Enrutar a Host Especificado
¿AVP de Dominio de Destino Presente?
¿Par Conectado?
Si
No
Filtrar Pares por Dominio
No
El Enrutamiento Falla
Filtrar por ID de Aplicación
pares
:destination_host
¿Múltiples Pares Coinciden?
Si
No
Algoritmo de Selección de Pares
:random
:failover
Enrutar a Pares Especificados
Enrutar a AVP de Host de Destino
Selección Aleatoria
Primer Par
Par Único
Enrutar al Par Seleccionado

- **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 Diámetro 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 múltiples pares coinciden con los criterios de enrutamiento, el `peer_selection_algorithm` configurado determina la selección:

- `:random` - Selecciona aleatoriamente entre 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 caídos se excluyen automáticamente

Limitaciones del Enrutamiento Estándar

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

Los módulos de 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 de Diámetro del DRA (nombre de dominio completamente calificado)
`realm`	String	El dominio de Diámetro 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 de Diámetro (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 de Diámetro:

```
%{
  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 de Diámetro del par (FQDN) - debe coincidir exactamente para el enrutamiento
`realm`	String	Dominio de Diámetro 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 de Diámetro 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 que el par se conecte

Modos de Conexión

Iniciar Conexión (`initiate_connection: true`)

- DRA actúa como cliente de Diámetro
- 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 está disponible

Aceptar Conexión (`initiate_connection: false`)

- DRA actúa como servidor de Diámetro
- 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)
- **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 múltiples pares coinciden con los criterios de enrutamiento, el `peer_selection_algorithm` determina cuál es elegido

Consideraciones de Seguridad

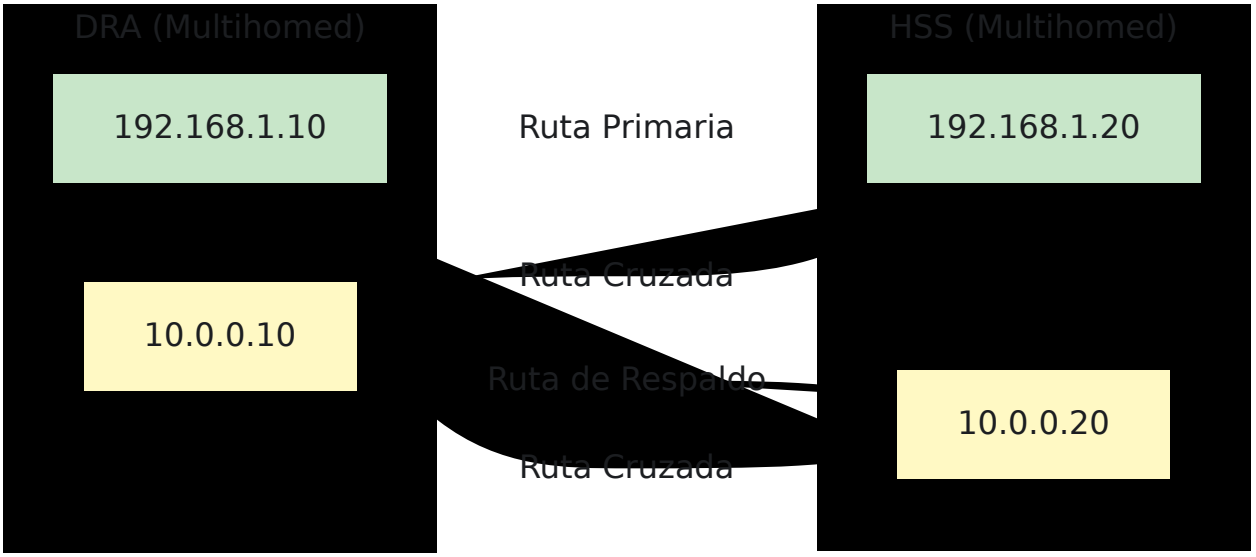
- Establecer `allow_undefined_peers_to_connect: false` en producción
- Habilitar `log_unauthorized_peer_connection_attempts: true` para monitoreo de seguridad
- Asegurar que las reglas de firewall coincidan con las configuraciones de `listen_ip` y `listen_port`
- Validar certificados de pares al usar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

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

Cómo Funciona



- Los latidos SCTP monitorean todas las rutas de red
- El failover automático ocurre si la ruta primaria se vuelve inalcanzable
- No hay interrupción de la sesión de Diámetro durante el cambio de ruta
- El kernel maneja la selección de rutas 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
(requerido)
    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 obligatorio 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 de 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 primaria)
- TLS sobre multihoming SCTP puede tener limitaciones de compatibilidad
- El tiempo de failover de ruta depende de los parámetros SCTP del kernel

---

Tablas de Referencia

IDs de Aplicación 3GPP Comunes

Application-Id	Interface	Descripción
16777251	S6a/S6d	Autenticación de MME/SGSN a HSS y datos de suscripción
16777252	S13/S13'	MME a verificación de identidad de equipo EIR
16777238	Gx	Control de políticas y cobros de PCEF a PCRF
16777267	S9	Política de roaming de PCRF local a PCRF visitado
16777272	Sy	Vinculación de sesión de PCRF a OCS
16777216	Cx	Registro IMS de I-CSCF/S-CSCF a HSS
16777217	Sh	Datos de usuario IMS de AS a HSS
16777236	SLg	Servicios de localización de MME/SGSN a GMLC
16777291	SLh	Información de suscriptor de localización de GMLC a HSS
16777302	S6m	MTC-IWF a HSS/HLR para dispositivos M2M
16777308	S6c	Enrutamiento de SMS de SMS-SC/IP-SM-GW a HSS
16777343	S6t	Eventos de monitoreo de SCEF a HSS

Application-Id	Interface	Descripción
16777334	Rx	Autorización de medios de AF a PCRF

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	Dominio objetivo
293	Destination-Host	DiameterIdentity	Host objetivo (opcional)
296	Origin-Realm	DiameterIdentity	Dominio 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 de Diámetro, 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 Desconexión de Par
316	ULR/ULA	Solicitud/Respuesta de Actualización de Localización (S6a)
317	CLR/CLA	Solicitud/Respuesta de Cancelación de Localización (S6a)
318	AIR/AIA	Solicitud/Respuesta de Información de Autenticación (S6a)
321	PUR/PUA	Solicitud/Respuesta de Purga de 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 de Diámetro entrantes** (no paquetes de respuesta). Los paquetes de respuesta siguen el enrutamiento de sesión establecido de regreso al par de origen - vea Enrutamiento de Respuestas para detalles.

Configuración

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

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: <rule_identifier>
      match: <match_scope>
      filters: [<filter_list>]
      route:
        peers: [<peer_list>]
```

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 Ruta a continuación)

Acciones de Ruta

El parámetro `route` admite múltiples acciones:

Enrutar a Pares

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

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

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

Enrutar a AVP de Host de Destino

```
route: :destination_host
```

Enruta al par especificado en el AVP de Host de Destino (293). Si falta el AVP de Host de Destino, el enrutamiento vuelve al comportamiento normal.

Descartar Tráfico

```
route: :drop
```

Descarta silenciosamente el mensaje sin enviar ninguna respuesta. Utilizar 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 descarte de Diámetro de Erlang
- Métrica: `diameter_advanced_routing_drop_count_total` (ver Métricas de Prometheus)

Generar Respuesta de Error

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

Genera una respuesta de error de Diámetro 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 (dominio no soportado)
- `3004` - DIAMETER_TOO_BUSY (protección contra sobrecarga, limitación de tasa)
- `5012` - DIAMETER_UNABLE_TO_COMPLY (rechazo general)

Comportamiento:

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

Filtros Disponibles

Filtros Estándar

Disponibles tanto en Enrutamiento Avanzado como en Transformación Avanzada:

- `:application_id` - Coincidir ID de aplicación de Diámetro (ver referencia de ID de Aplicación)
  - Valor único: `{:application_id, 16777251}` (S6a/S6d)
  - Múltiples valores: `{:application_id, [16777251, 16777252]}` (S6a o S6b)
- `:command_code` - Coincidir código de comando de Diámetro
  - Valor único: `{:command_code, 318}` (solicitud AIR)
  - Múltiples 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.*"}}`
  - Múltiples 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
  - Par único: `{:via_peer, "omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org"}`
  - Múltiples 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)

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

- **`:from_peer`** - Coincidir el par que envió la respuesta (solo paquetes de respuesta)

    - Par único: `{:from_peer, "hss-01.example.com"}`
    - Múltiples pares: `{:from_peer, ["hss-01.example.com", "hss-02.example.com"]}`

- **`:packet_type`** - Coincidir 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
    - Los 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 utiliza una lista, se 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 basados en qué DRA 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 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 el Origin-Realm específico con patrones IMSI coincidentes a los pares DRA designados.

Ejemplo 3: Enrutamiento Dinámico con :destination_host

Enrutar al valor del AVP de Host de Destino 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 IMSI
      route: :destination_host
```

Cómo funciona:

- Cuando los filtros coinciden, enruta al par especificado en el AVP de Host de Destino (293)
- Si falta el AVP de Host de Destino, 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 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 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 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 comandos 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
de UE)
      route: {:error, 5012}
```

Cómo funciona:

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

---

Módulo de Transformación Avanzada

El módulo de Transformación Avanzada permite la modificación dinámica de los AVP de mensajes de Diámetro en función de criterios de coincidencia. Vea Procesamiento de Reglas para detalles sobre cómo se evalúan las reglas.

Configuración

Habilite el módulo y defina las reglas de transformación:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: <rule_identifier>
      match: <match_scope>
      filters: [<filter_list>]
      transform:
        action: <transform_action>
        avps: [<avp_modifications>]
```

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 de Diámetro)

- **`: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 de Diámetro (por ejemplo, `:diameter_gen_3gpp_s6a`)

Paquetes de Respuesta (Respuestas de Diámetro)

- `: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, {<code>, <new_value>}}` - Establecer AVP en nuevo valor

Eliminando AVPs:

- `{:avp, {<code>, :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 Dominio de Destino Basada en el Par

Reescribir el Dominio de Destino basado en 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 las solicitudes S6a se enrutan a pares DRA específicos y coinciden con el patrón IMSI, reescribe el Dominio de Destino para la red del Operador X.

Ejemplo 2: Enrutamiento de Múltiples Operadores 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: Enruta diferentes rangos de suscriptores IMSI a los dominios de red apropiados basados en patrones IMSI. La primera regla coincidente gana (ver Orden de Ejecución).

Ejemplo 3: Reescritura de Dominio 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 Dominio de Destino para un suscriptor específico de MVNO 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 AIR (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 Localización)
        - '{:packet_type, :answer}'           # Solo paquetes de
respuesta
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'  #
Dominio 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 de Actualización de Localización (ULA) de un Dominio de Origen específico
- Utiliza la acción `:overwrite` para reemplazar todo el AVP agrupado de 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 dominio

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 de Diámetro (`: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 - validar a fondo en el entorno de prueba primero

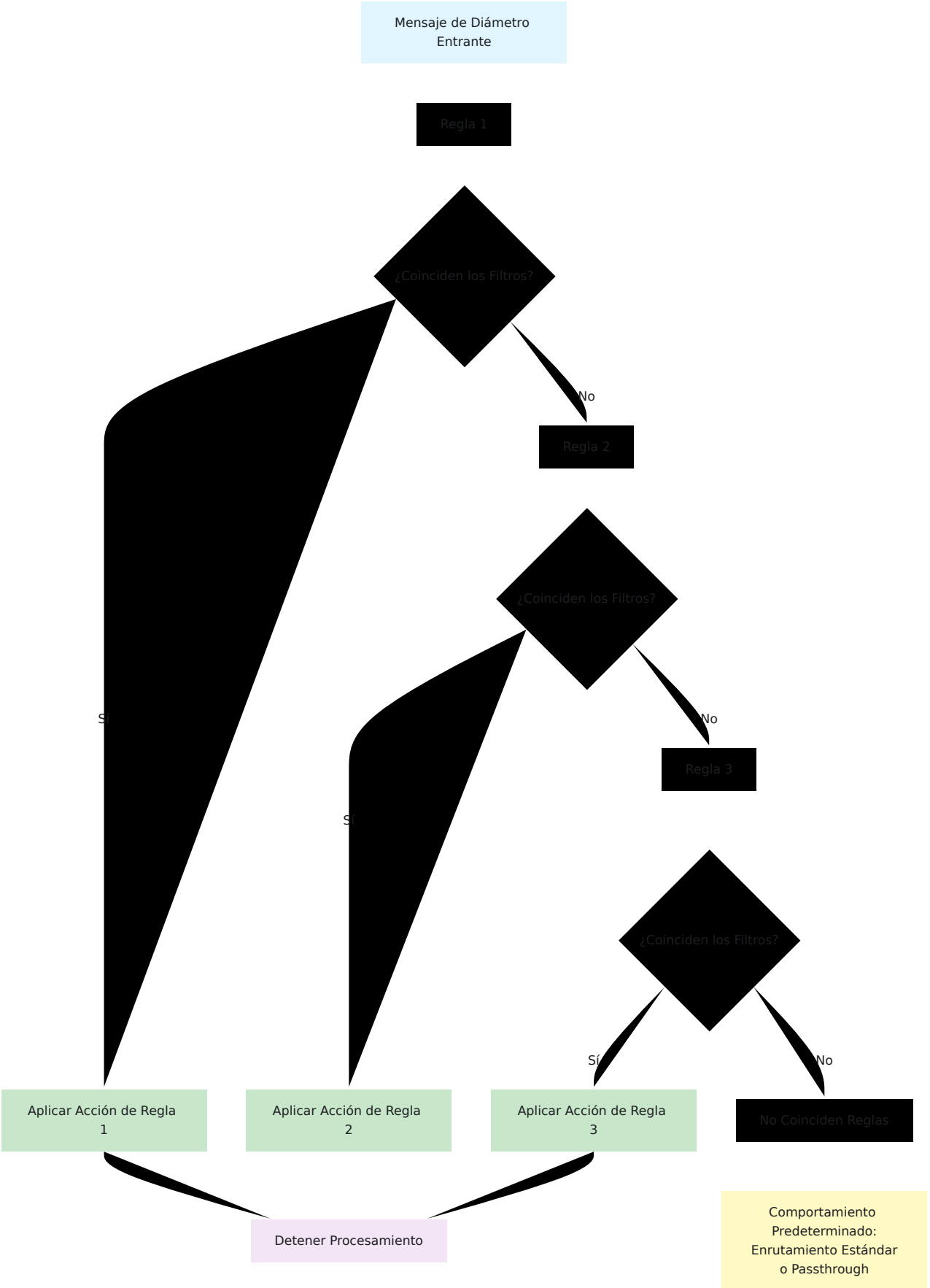
Casos de Uso

- **Soporte para MVNO**: Enrutar tráfico de operadores virtuales a redes centrales alojadas
- **Migración de Red**: Redirigir gradualmente a los suscriptores a nueva infraestructura
- **Traducción de Dominio**: Convertir entre diferentes esquemas de nombres para socios de roaming
- **Multi-tenencia**: Aislar poblaciones de suscriptores por dominio
- **Enrutamiento de Operadores**: Dirigir tráfico a redes de operadores correctas basadas en rangos IMSI

Procesamiento de Reglas

Aplica tanto a los módulos de Enrutamiento Avanzado como a Transformación Avanzada.

Orden de Ejecución



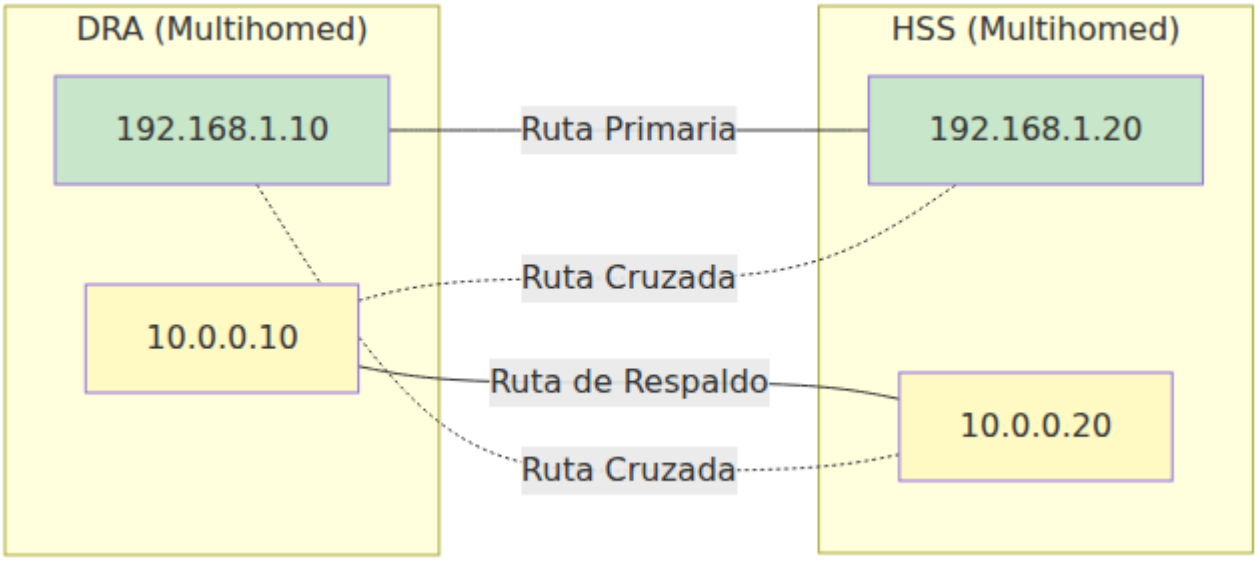
1. Las reglas se evalúan **en orden de arriba hacia abajo** según se definen en la configuración
2. Los filtros dentro de una regla se evalúan según el parámetro `match` (`:all`, `:any` o `:none`)
3. **La primera regla coincidente gana** - las reglas subsiguientes no se evalúan
4. Si no coinciden reglas,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

Lógica de Filtros

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

match: :all (Lógica AND)

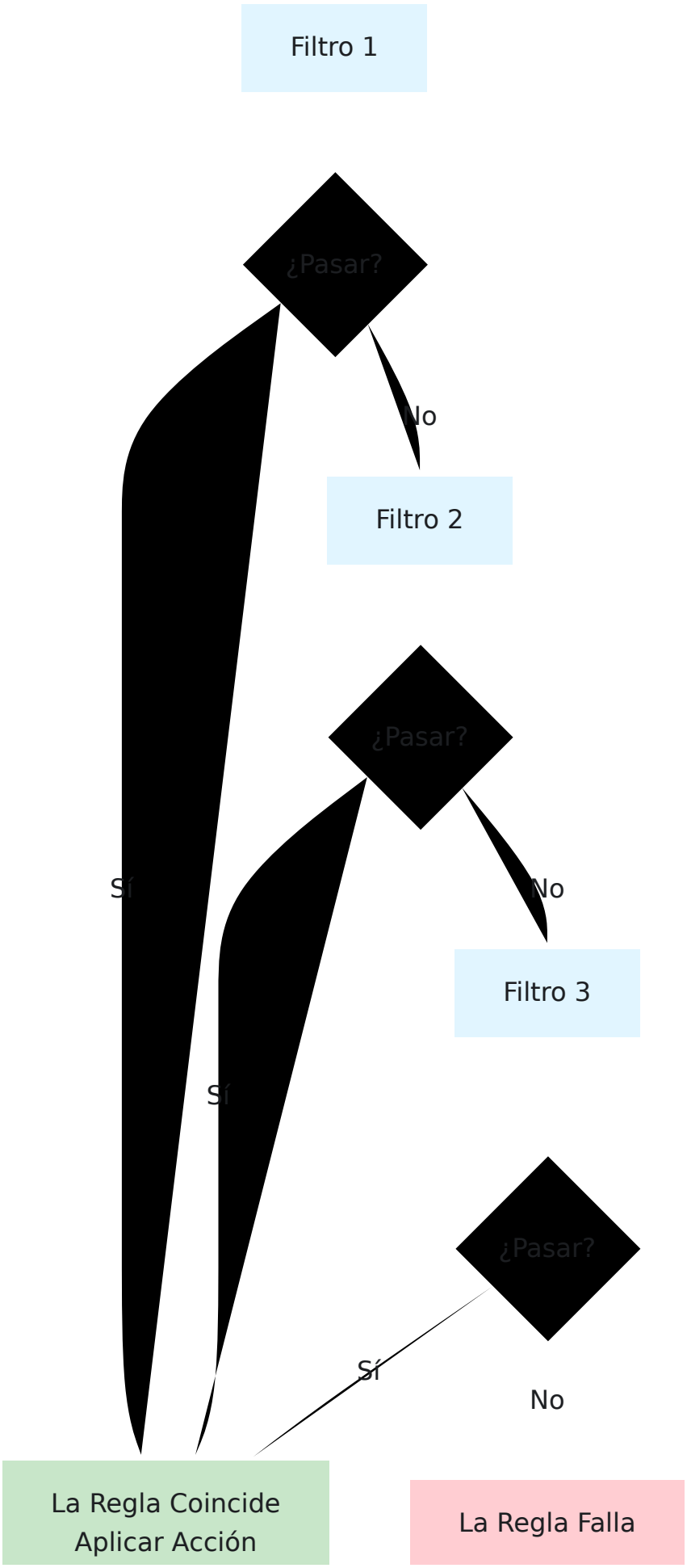
Todos los filtros deben coincidir para que la regla tenga éxito.



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

match: :any (Lógica OR)

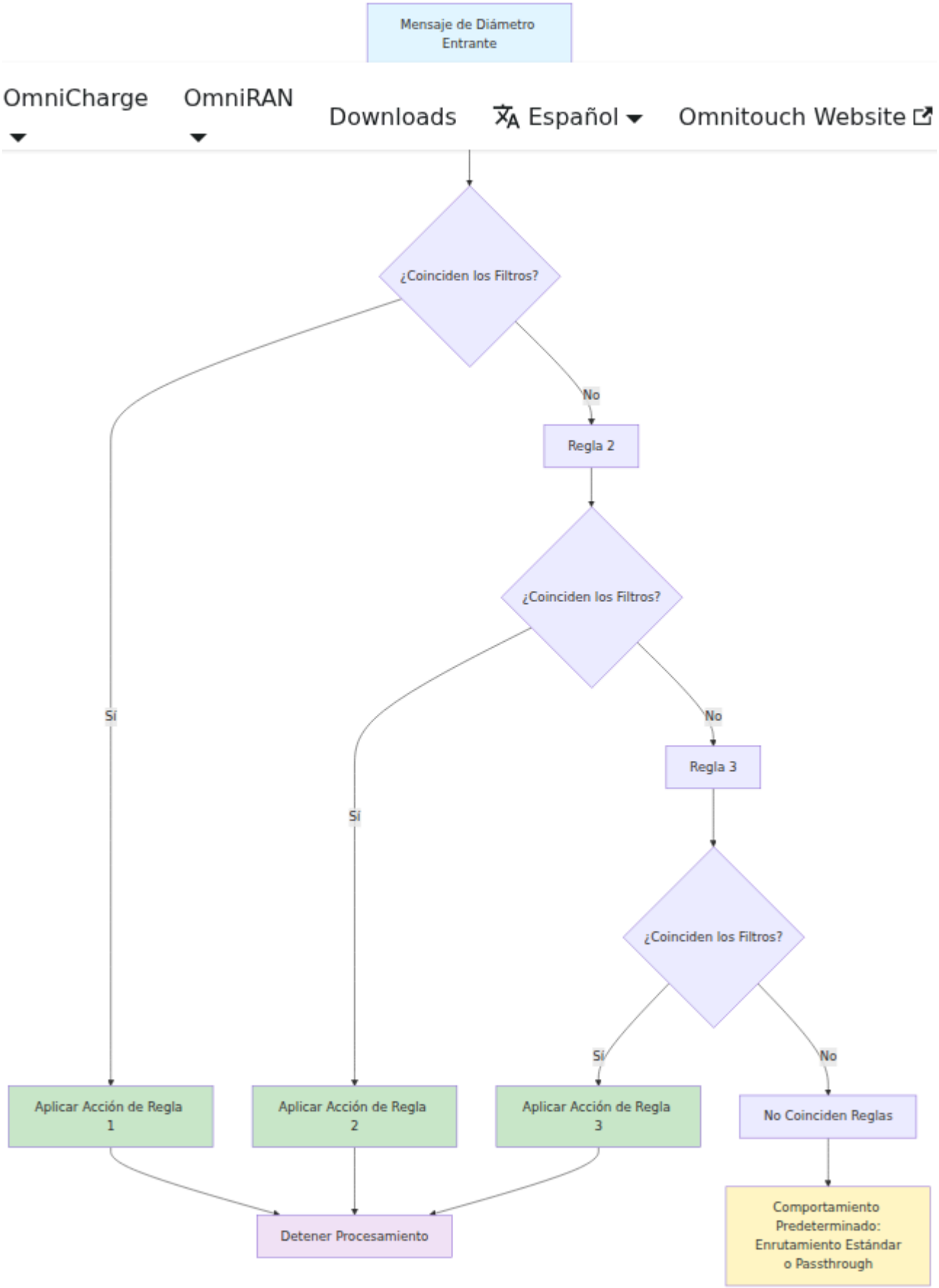
Al menos un filtro debe coincidir para que la regla tenga éxito.


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 OR filtro2 OR filtro3` (cualquiera pasa).

match: :none (Lógica NOR)

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



Ejemplo: Con 3 filtros, `NOT filtro1 AND NOT filtro2 AND NOT 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

Utilice 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**: Utilice valores descriptivos para `rule_name` para claridad operativa
5. **Monitoreo**: Rastree 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 de Diámetro 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 el seguimiento y la generación de informes de intentos de conexión LTE (S6a AIR/AIA)

Métricas Disponibles

Seguimiento de Intentos de Conexión

Rastrea los 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 de Diámetro de la respuesta del 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, host de origen y código de resultado

Casos de Uso

- **Análisis de Tasa de Éxito de Conexión** - Rastrear intentos de conexión exitosos vs fallidos por código de resultado
- **Resolución 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)
- **Análisis 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 Diámetro:

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

Notas Importantes

- Las métricas de intentos de conexión solo rastrean pares 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

---

Métricas de Prometheus

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

Métricas de Diámetro del Núcleo

Estado de Pares

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

- `origin_host` - Identidad de Diámetro 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 de Diámetro intercambiados con pares **Etiquetas**:

- `origin_host` - Identidad de Diámetro del par
- `received_from` - Par del cual se recibió el mensaje
- `application_id` - ID de Aplicación de Diámetro (ver referencia de ID de Aplicación)
- `cmd_code` - Código de Comando de Diámetro (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 solicitudes S6a AIR 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 de Diámetro por código de resultado **Etiquetas**:

- `origin_host` - Solicitante original
- `routed_to` - Par que envió la respuesta
- `application_id` - ID de Aplicación de Diámetro
- `cmd_code` - Código de Comando de Diámetro
- `application_name` - Nombre de la aplicación
- `cmd_name` - Nombre del comando
- `result_code` - Código de Resultado de Diámetro o Código de Resultado Experimental

Ejemplo:

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

# Tasa de errores 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"})

# P95 tiempo de respuesta 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
- `application_id` - ID de Aplicación de Diámetro
- `cmd_code` - Código de Comando de Diámetro
- `application_name` - Nombre de la aplicación
- `cmd_name` - Nombre del comando

Ejemplo:

```
# Tasa de solicitudes no respondidas
rate(diameter_peer_unanswered_request_count_total[5m])

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

Intent