

# Guía de Operaciones DRA

## Tabla de Contenidos

1. [Enrutamiento de Diámetro Estándar](#)
  2. [Configuración Base del DRA](#)
  3. [Tablas de Referencia](#)
    - [IDs de Aplicación 3GPP Comunes](#)
    - [Códigos AVP Comunes](#)
  4. [Módulo de Enrutamiento Avanzado](#)
  5. [Módulo de Transformación Avanzada](#)
  6. [Procesamiento de Reglas](#)
  7. [Módulo de Métricas Extendidas](#)
  8. [Solución de Problemas](#)
- 

## Visión General de la Arquitectura DRA

---

## Enrutamiento de Diámetro Estándar

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

### Enrutamiento de Solicitudes

El DRA enruta los 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 basado en el realm
  - El DRA selecciona un par conectado que publicita soporte para el realm objetivo

- Se aplica balanceo de carga cuando múltiples pares coinciden con el realm
  - El enrutamiento basado en realm permite flexibilidad entre múltiples hosts
3. **Application-Id** - Los pares se filtran por las aplicaciones de Diámetro soportadas
- Solo se consideran los pares que publicitan soporte para el Application-Id 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 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 hop-by-hop:** El DRA utiliza el Identificador Hop-by-Hop 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 de 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 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 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 <a href="#">Identidad de Diámetro</a> del DRA (nombre de dominio completamente calificado) |
| realm        | String  | El <a href="#">realm de Diámetro</a> del DRA  |
| product_name | String  | Nombre del producto publicitado en mensajes CER/CEA   |
| vendor_id    | Integer | Vendor-ID según se define en <a href="#">RFC 6733 Sección 5.3.3</a> (10415 = 3GPP)            |

## Configuraciones de Red

| Parámetro       | Tipo    | Descripción   |
|-----------------|---------|---|
| listen_ip       | String  | Dirección IP en la que el DRA escucha conexiones entrantes                                |
| 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  | Realm de Diámetro del par  |
| ip                  | String  | Dirección IP del par para la conexión  |
| 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 a 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 es alcanzable

### 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 y minúsculas)
- **Intercambio de Capacidades:** Al conectarse, los pares intercambian aplicaciones soportadas a través de mensajes CER/CEA
- **Soporte de Aplicaciones:** El DRA publicita 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, `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 de firewall coincidan con las configuraciones de `listen_ip` y `listen_port`
  - Validar certificados de pares al usar TLS
- 

## Tablas de Referencia

### IDs de Aplicación 3GPP Comunes

| Application-<br>Id | Interfaz     | Descripción   |
|--------------------|--------------|---|
| 16777251           | S6a/S6d      | Autenticación y datos de suscripción de MME/SGSN a HSS  |
| 16777252           | S13/<br>S13' | Verificación de identidad de equipo de MME a EIR        |
| 16777238           | Gx           | Control de políticas y carga 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                      |
| 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- | DiameterIdentity | Realm objetivo                             |

| <b>Código</b> | <b>Nombre AVP</b>   | <b>Tipo</b>      | <b>Uso</b>                                   |
|---------------|---------------------|------------------|--|
|               | Realm               |                  |  |
| 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 de Diámetro, no de los AVP:

| <b>Código</b> | <b>Nombre de Comando</b> | <b>Descripción</b>  |
|---------------|--------------------------|---|
| 257           | CER/CEA                  | Solicitud/Respuesta de Intercambio de Capacidades         |
| 258           | RAR/RAA                  | Solicitud/Respuesta de Reautenticación                    |
| 274           | ASR/ASA                  | Solicitud/Respuesta de Terminación de Sesión              |
| 275           | STR/STA                  | Solicitud/Respuesta de Terminación de Sesión              |
| 280           | DWR/DWA                  | Solicitud/Respuesta de Vigilancia de Dispositivo          |
| 282           | DPR/DPA                  | Solicitud/Respuesta de Desconexión de Par                 |
| 316           | ULR/ULA                  | Solicitud/Respuesta de Actualización de Ubicación (S6a)   |
| 317           | CLR/CLA                  | Solicitud/Respuesta de Cancelación de Ubicación (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 - 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_de_regla>
    match: <alcance_de_coincidencia>
    filters: [<lista_de_filtros>]
    route:
      peers: [<lista_de_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 <a href="#">Filtros Disponibles</a> )  |
| route.peers | Lista de nombres de host de pares de destino (deben ser pares de Diámetro preconfigurados en su configuración DRA), O use el destino especial :destination_host para enrutar basado en el <a href="#">AVP Destination-Host (293)</a> |

**Importante:** Los pares especificados en route.peers deben ser:

- Definidos en la configuración de pares de Diámetro 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)

## Filtros Disponibles

### Filtros Estándar

Disponibles en [Enrutamiento Avanzado](#) y [Transformación Avanzada](#):

- **:application\_id** - Coincidir con el 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 con el código de comando de Diámetro
  - Valor único: { :command\_code, 318 } (solicitud AIR)
  - Múltiples valores: { :command\_code, [ 317, 318 ] } (ULR o AIR)

- **:avp** - Coincidir con el 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 con 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 con 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 con 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 con 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 compatibles** y no coincidirán
- Los valores binarios complejos **no son compatibles**
- Usar formato: `{:avp, {code, value}}`
- **Operadores de Lista:** Compatibles 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 a través del Par

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 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 Origin-Realm 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

## Módulo de Transformación Avanzada

El módulo de Transformación Avanzada permite la modificación dinámica de los AVPs de mensajes de Diámetro 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_de_regla>
    match: <alcance_de_coincidencia>
    filters: [<lista_de_filtros>]
    transform:
      action: <acción_de_transformación>
      avps: [<modificaciones_de_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 <a href="#">Lógica de Filtros</a> |
| filters          | Lista de condiciones de filtro (ver <a href="#">Filtros Disponibles</a> )   |
| transform.action | Tipo de transformación (:edit, :remove, o :overwrite)   |
| transform.avps   | Lista de modificaciones de AVP a aplicar (ver <a href="#">referencia de código AVP</a> )                                      |

## 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 hay reglas que coincidan, el paquete se pasa a través de forma transparente sin ninguna transformación.

## Sintaxis de Modificación de AVP

### Modificación estándar:

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

### **Eliminar AVPs:**

- `{:avp, {<code>, :any}}` - Eliminar AVP por ID (elimina independientemente del valor actual)
- Nota: Se admite la eliminación basada en `avp_id`; no se admite la eliminación basada en contenidos de AVP

### **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 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 de IMSI, reescribe el Destination-Realm 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 realms de red apropiados basados en patrones de IMSI. La primera regla que coincide 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 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, "9999990000000001"}}'
- '{:avp, {264, :any}}' # Origin-Host debe existir con
cualquier valor
  transform:
    action: ":edit"
    avps:
      - '{:avp, {1, "9999990000000002"}}'

```

### Cómo funciona:

- Coincide solo con paquetes de **solicitud** S6a AIR (no paquetes de respuesta)
- Verifica que User-Name (AVP 1) sea igual a "9999990000000001"
- Verifica que Origin-Host (AVP 264) existe con cualquier valor
- Reescribe User-Name a "9999990000000002"
- 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
resposta
    - '{: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], ["0800"],
                [{:s6a_AMBR, 4200000000, 4200000000, [], [], []}],,
                [], [], [], [], [], [], [], [], [], [], [],
                [], []}],,
                [1], [], []}
            ]}, []
        }}'
```

## Cómo funciona:

- Coincide con paquetes de respuesta S6a de Actualización de Ubicación (ULA) desde un Realm 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 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 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 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 Operadores:** Dirigir tráfico a redes de operadores correctas basadas en rangos de IMSI

## **Procesamiento de Reglas**

Se 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 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 AND filtro2 AND filtro3` deben ser verdaderos.

### **match: :any (Lógica OR)**

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

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

## Mejores Prácticas

1. **Especificidad:** Ordenar reglas de las más específicas a las más generales
  2. **Rendimiento:** Colocar las coincidencias más comunes primero para reducir la sobrecarga de procesamiento
  3. **Pruebas:** Validar patrones regex antes de la implementación
  4. **Documentación:** Usar valores descriptivos de `rule_name` para claridad operativa
  5. **Monitoreo:** Rastrear 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 analítica 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   |
|---|---|
| <code>enabled</code>                          | Establecer en <code>true</code> para activar el módulo de métricas extendidas |
| <code>attach_attempt_reporting_enabled</code> | Habilitar el seguimiento e informe de intentos de conexión LTE (S6a AIR/AIA)  |

### Métricas Disponibles

#### Seguimiento de Intentos de Conexión

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

**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 un 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 297)
  - Emite la métrica con IMSI, origin host 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
- **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 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 (Application-Id 16777251, Command-Code 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 de forma independiente de los módulos de enrutamiento y transformación
- 

## Solución de Problemas

### Regla No Coincide

- Verifique que todas las condiciones de filtro sean correctas
- Compruebe que los códigos AVP coincidan con su aplicación de Diámetro (ver [referencia de código AVP](#))
- Pruebe patrones regex de forma independiente (ver [Patrones de Expresión Regular](#))
- Asegúrese de que el tipo de mensaje coincida con el alcance de match (ver [Lógica de Filtros](#))
- Revise [Filtros Disponibles](#) para asegurarse de que está utilizando el tipo de filtro correcto para su módulo

### Enrutamiento Inesperado

- Revise el orden de las reglas - [la primera coincidencia gana](#)
- Verifique que los nombres de los pares sean correctos y alcanzables
- Compruebe si hay reglas en conflicto con filtros superpuestos
- Confirme el comportamiento de [Enrutamiento de Diámetro Estándar](#) cuando no coinciden reglas

### Transformación No Aplicada

- Confirme que los códigos AVP sean correctos para su caso de uso (ver [referencia de código AVP](#))
- Para la acción :edit: Verifique que el AVP exista en el mensaje (edit no creará nuevos AVPs)
- Verifique que los filtros coincidan con el flujo de mensaje previsto
- Verifique el filtro de tipo de paquete si se utiliza (:request vs :answer)
- Asegúrese de que la acción sea compatible con el tipo de paquete (la :edit solo funciona en solicitudes - ver [Acciones de Transformación](#))

- Revise [Procesamiento de Reglas](#) para el orden de ejecución