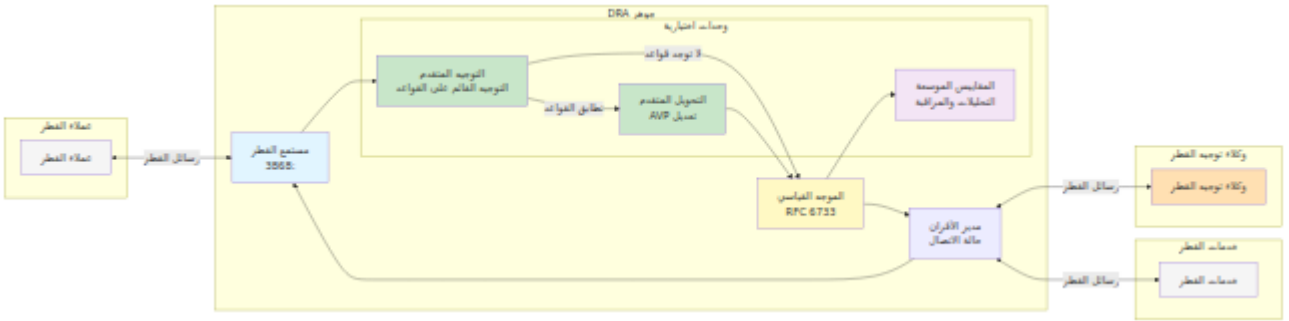
دليل عمليات DRA

جدول المحتويات

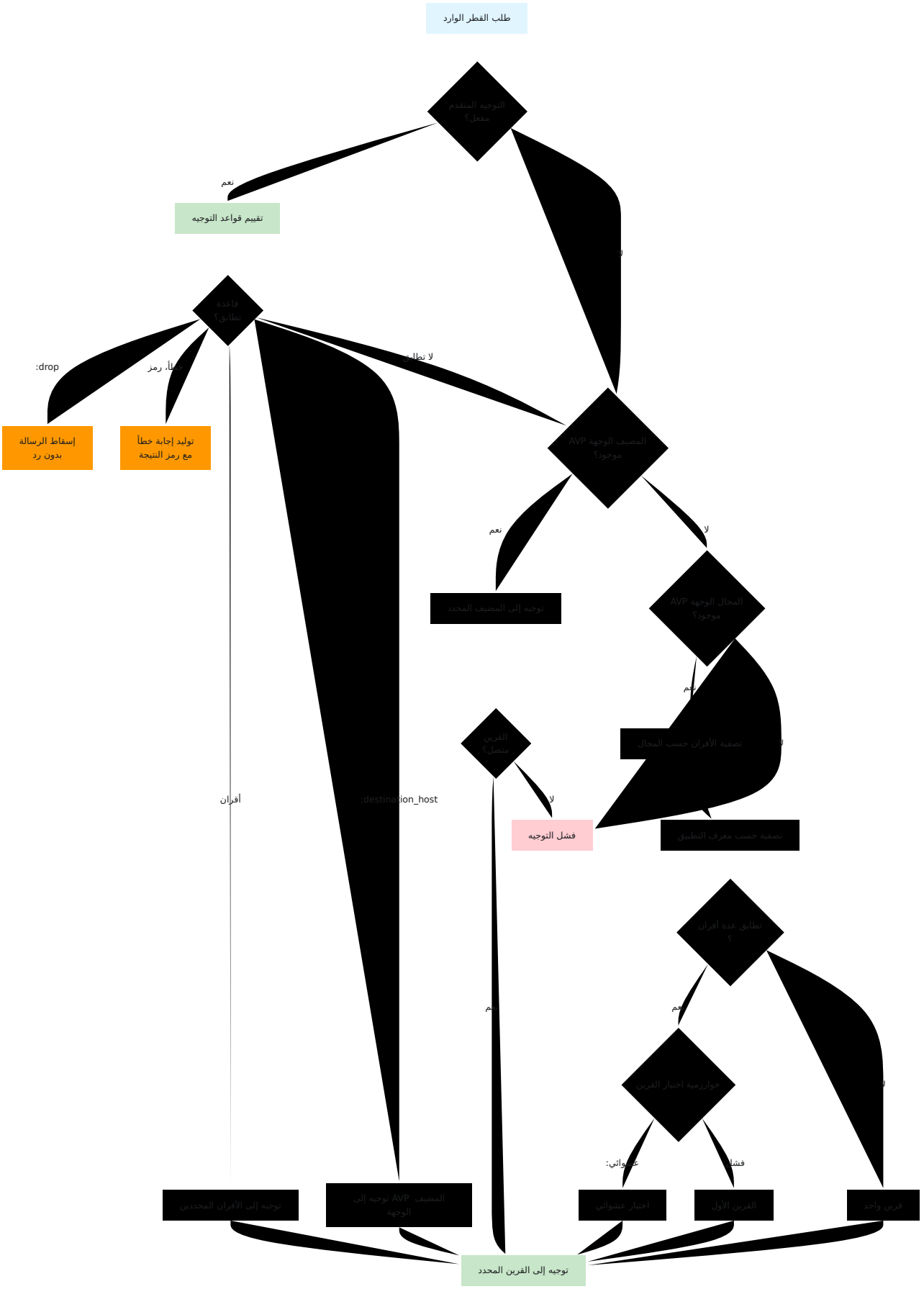
1. توجيه القطر القياسي
2. تكوين DRA الأساسي
3. تعدد الإرسال عبر SCTP
4. جداول المرجع
    - معرفات التطبيقات الشائعة 3GPP
    - رموز AVP الشائعة
5. وحدة التوجيه المتقدم
6. وحدة التحويل المتقدم
7. معالجة القواعد
8. وحدة المقاييس الموسعة
9. مقاييس بروميثيوس
    - مقاييس القطر الأساسية
    - مقاييس وحدة التوجيه المتقدم
10. استكشاف الأخطاء وإصلاحها

نظرة عامة على بنية DRA



توجيه القطر القياسي

بتنفيذ توجيه القطر القياسي بناءً على DRA بدون وحدات التوجيه المتقدم أو التحويل المتقدم، يقوم بروتوكول القطر الأساسي (RFC 6733):


طلب القطر الوارد
التوجيه المتقدم مفعّل؟
نعم
تقييم قواعد التوجيه
قاعدة تطابق؟
drop:
خطأ، رمز
لا تطابق
إسقاط الرسالة بدون رد
توليد إجابة خطأ مع رمز النتيجة
المضيف الوجهة AVP موجود؟
نعم
توجيه إلى المضيف المحدد
المجال الوجهة AVP موجود؟
نعم
تصفية الأقران حسب المجال
تصفية حسب معرف التطبيق
القرين متصل؟
لا
فشل التوجيه
أقران
destination_host:
تطابق عدة أقران ؟
نعم
خوارزمية اختيار القرين
عشوائي:
فشل
القرين الأول
قرين واحد
توجيه إلى الأقران المحددين
المضيف AVP توجيه إلى الوجهة
توجيه إلى القرين المحدد

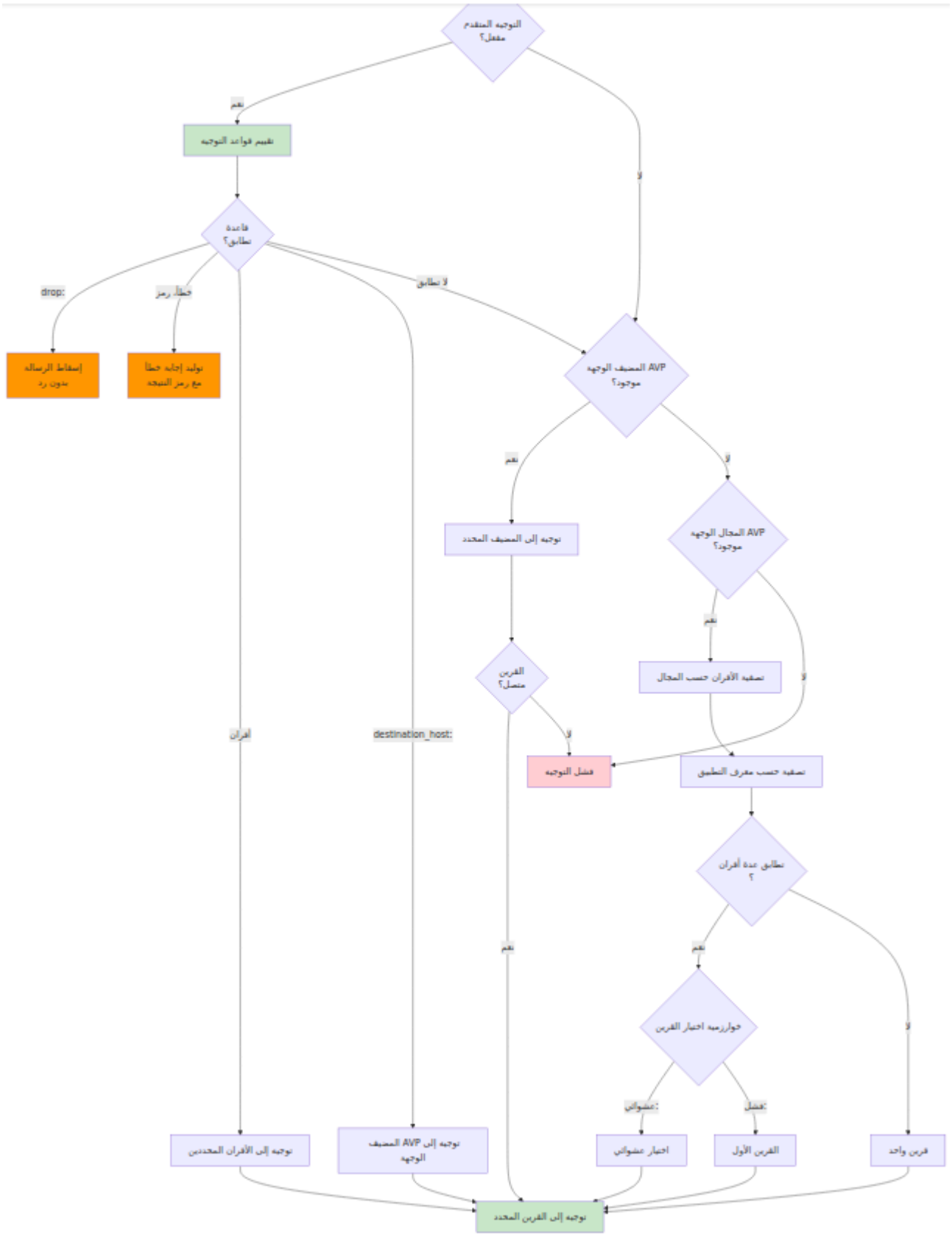
توجيه الطلبات

يقوم DRA بتوجيه رسائل الطلب باستخدام آلية قائمة على الأولويات كما هو محدد في RFC 6733 القسم 6.1:

1. **AVP المضيف الوجهة (293)** - إذا كان موجودًا، يقوم DRA بالتوجيه مباشرة إلى القرين المحدد
    - هذه هي آلية التوجيه ذات الأولوية القصوى
    - إذا لم يكن القرين متصلًا، يفشل التوجيه
    - يوفر تحكمًا صريحًا في التوجيه على مستوى المضيف
2. **AVP المجال الوجهة (283)** - إذا كان المضيف الوجهة غائبًا، يتم التوجيه بناءً AVP على المجال
    - يختار DRA قرينًا متصلًا يعلن عن دعمه للمجال المستهدف
    - يتم تطبيق موازنة الحمل عند تطابق عدة أقران مع المجال
    - يسمح التوجيه القائم على المجال بالمرونة عبر عدة مضيفين
3. **معرف التطبيق** - يتم تصفية الأقران حسب التطبيقات المدعومة من القطر
    - يتم اعتبار الأقران الذين يعلنون عن دعم معرف التطبيق للرسالة فقط
    - أثناء إنشاء اتصال القرين (CER/CEA) بناءً على تبادل القدرات
    - للرجوع إليها GPP انظر معرفات التطبيقات الشائعة 3

توجيه الإجابة

تستخدم حزم الإجابة آلية توجيه مختلفة تمامًا عن الطلبات:



- **التوجيه القائم على الجلسة**: تتبع حزم الإجابة دائمًا المسار العكسي للطلب

- **الحفاظ على معرف من طرف إلى طرف**: يبقى المعرف من طرف إلى طرف دون تغيير عبر جميع القفزات
- **التوجيه من طرف إلى طرف**: يستخدم DRA معرف القفز للحفاظ على حالة التوجيه (يتغير في كل قفزة)
- **لا تقييم للقواعد**: لا يقوم DRA بتقييم قواعد التوجيه أو محتويات AVP للإجابات
- **الارتباط القائم على الحالة**: تتبع الجداول الداخلية للتوجيه أي قرين أرسل كل طلب

لماذا لا يتم توجيه الإجابات بواسطة الوحدات المتقدمة:

- توجيه الإجابات حتمي ويجب أن يعود إلى القرين الأصلي
- يتطلب بروتوكول القطر أن تتبع الإجابات المسار الذي تم إنشاؤه للطلب
- يتم اتخاذ قرارات التوجيه للإجابات بناءً على سياق الطلب الأصلي، وليس محتوى الإجابة
- يضمن ذلك إدارة الجلسات بشكل صحيح ويمنع حلقات التوجيه

انظر RFC 6733 القسم 6.2 للحصول على تفاصيل توجيه رسائل الإجابة.

اختيار القرين

عندما يتطابق عدة أقران مع معايير التوجيه، تحدد خوارزمية `peer_selection_algorithm` الاختيار:

- **`عشوائي`**: يختار عشوائيًا من الأقران المتاحة (افتراضي)
- **`فشل`**: يختار دائمًا القرين الأول في القائمة (قائم على الأولوية)
- يجب أن تكون الأقران في **حالة متصلة** ليتم اختيارها
- يتم استبعاد الأقران المنفصلة أو المعطلة تلقائيًا

قيود التوجيه القياسي

- لا توجد قواعد توجيه مخصصة بناءً على قيم AVP (مثل أنماط IMSI)
- لا توجد ترجمة للمجال أو تعديل لـ AVP
- لا يمكن التوجيه بناءً على القرين الأصلي
- تحكم محدود في توزيع الحركة

تقوم وحدات التوجيه المتقدم و التحويل المتقدم بتمديد هذا السلوك القياسي مع قدرات التوجيه القائم على القواعد وتلاعب الحزم.

تكوين DRA الأساسي

يتطلب DRA تكوينًا أساسيًا يحدد هويته وإعدادات الشبكة واتصالات الأقران. يؤسس هذا التكوين الأساس لجميع عمليات التوجيه.

هيكل التكوين

```
%{
  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: [
    # تكوينات الأقران...
  ]
}
```

معلمات هوية DRA

المعلمة	النوع	الوصف
`host`	سلسلة	هوية DRA (اسم النطاق المؤهل بالكامل) القطرية
`realm`	سلسلة	مجال DRA القطري
`product_name`	سلسلة	اسم المنتج المعلن عنه في رسائل CER/CEA
`vendor_id`	عدد صحيح	معرف البائع كما هو محدد في القسم 5.3.3 RFC 6733 (3 = 10415GPP)

إعدادات الشبكة

المعلمة	النوع	الوصف
`listen_ip`	سلسلة أو قائمة	عنوان IP (عناوين) التي يستمع عليها DRA. لتعدد IP استخدم قائمة من سلاسل، الإرسال عبر SCTP (انظر SCTP تعدد الإرسال عبر)
`listen_port`	عدد صحيح	منفذ TCP/SCTP لاتصالات القطر (القياسي: 3868)
`service_name`	ذرة	معرف الخدمة الداخلية لـ Erlang
`request_timeout`	عد�� صحيح	مهلة بالمللي ثانية لزوج الطلب/الإجابة (الافتراضي: 5000)

إعدادات اختيار القرين

المعلمة	النوع	الوصف
`peer_selection_algorithm`	ذرة	خوارزمية موازنة الحمل: `:random` (اختيار عشوائي) أو `:failover` (أولوية القرين الأول)
`allow_undefined_peers_to_connect`	Boolean	السماح بالاتصالات من الأقران غير الموجودين في التكوين (الافتراضي: `false`)
`log_unauthorized_peer_connection_attempts`	Boolean	تسجيل محاولات الاتصال من الأقران غير المصرح لهم

تكوين القرين

يحدد كل قرين في قائمة `peers` اتصال القطر:

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

معلمات القرين

المعلمة	النوع	الوصف
`host`	سلسلة	هوية القرين (FQDN) - يجب أن تتطابق تمامًا للتوجيه القطرية
`realm`	سلسل��	مجال القرين القطرية
`ip`	سلسلة	عنوان IP (مطلوب) الأساسي للقرين للاتصال
`ips`	قائمة	قائمة بعناوين IP لتعدد الإرسال عبر SCTP (SCTP اختياري، انظر تعدد الإرسال عبر)
`port`	عدد صحيح	منفذ القطر للقرين (عادة 3868)
`transport`	ذرة	بروتوكول النقل: `:diameter_tcp` أو `:diameter_sctp`
`tls`	Boolean	استخدم `true`، إذا كانت) TLS تمكين تشفير عادةً المنفذ 3869)
`initiate_connection`	Boolean	`true`: يتصل DRA بالقرين، `false`: ينتظر اتصال القرين DRA

أوضاع الاتصال

بدء الاتصال (`initiate_connection: true`)

- يعمل DRA كعميل قطر
- يبدأ DRA اتصال TCP/SCTP بالقرين
- يستخدم HSS أو PCRF أو أنظمة خلفية أخرى للاتصال بـ
- سيعيد DRA محاولة الاتصالات إذا كان القرين غير قابل للوصول

قبول الاتصال (`initiate_connection: false`)

- يعمل DRA كخادم قطر

- ينتظر DRA اتصال القرين
- يستخدم MME و SGSN و P-GW لاتصالات
- يجب أن يكون القرين في التكوين أو `allow_undefined_peers_to_connect: true`

مثال على التكوين

```
%{
  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 - ينتظر MME للاتصال
    %{
      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 يبدأ الاتصال
    %{
      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 مع TLS - DRA يبدأ اتصال آمن
    %{
      host: "pcrf01.mvno.example.com",
      realm: "mvno.example.com",
      ip: "10.100.3.22",
      port: 3869,
      transport: :diameter_tcp,
      tls: true,
```

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

ملاحظات مهمة

- **مطابقة اسم المضيف**: يجب أن تتطابق أسماء مضيفي القرين في قواعد التوجيه المكونة هنا (حساسة لحالة الأحرف) `host` المتقدم تمامًا مع قيمة
- **تبادل القدرات**: عند الاتصال، يتبادل الأقران التطبيقات المدعومة عبر رسائل CER/CEA
- **دعم التطبيق**: يعلن DRA 3 عن جميع التطبيقات المدعومة من GPP (انظر معرفات التطبيقات الشائعة 3GPP)
- **معرف البائع 10415**: قيمة قياسية لتطبيقات 3GPP
- **مهلة الطلب**: تؤثر على TTL (مهلة + 5 ثوانٍ) المقاييس الموسعة
- **اختيار القرين**: عندما يتطابق عدة أقران مع معايير التوجيه، تحدد `peer_selection_algorithm` أيها يتم اختياره

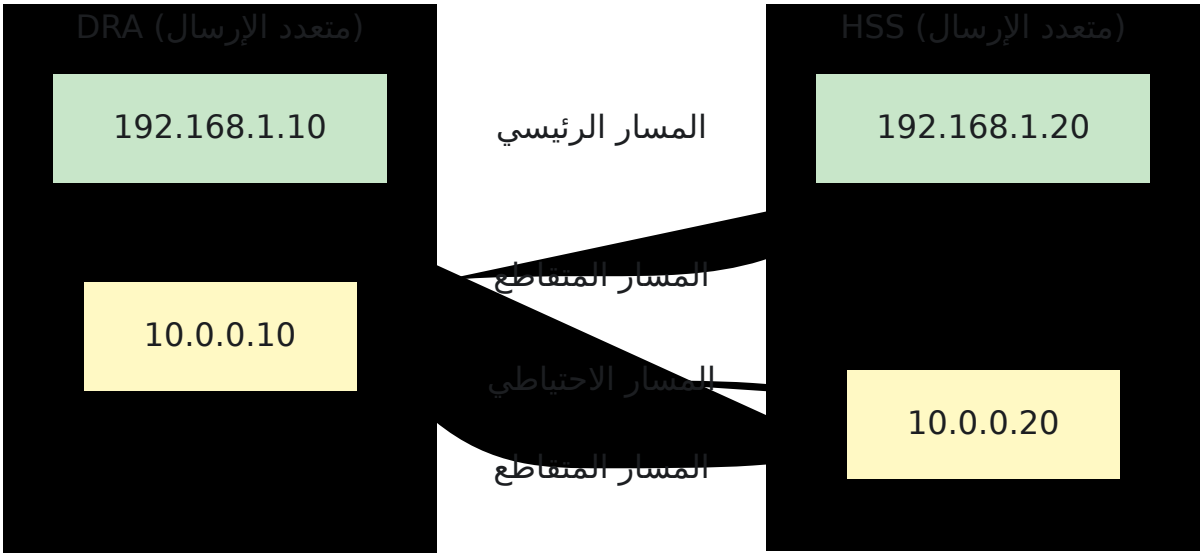
اعتبارات الأمان

- قم بتعيين `allow_undefined_peers_to_connect: false` في الإنتاج
- قم بتمكين `log_unauthorized_peer_connection_attempts: true` لمراقبة الأمان
- تأكد من أن قواعد جدار الحماية تتطابق مع إعدادات `listen_ip` و `listen_port`
- تحقق من شهادات الأقران عند استخدام TLS

SCTP تعدد الإرسال عبر

يوفر تعدد الإرسال عبر SCTP تكرار الشبكة من خلال السماح للنقاط النهائية بالارتباط بعدد من عناوين IP. يقوم SCTP تلقائيًا بالتبديل إلى مسار بديل دون إذا فشل المسار الشبكي الرئيسي، تعطيل جلسة القطر.

كيف يعمل



- جميع المسارات الشبكية SCTP تراقب نبضات
- يحدث التبديل التلقائي إذا أصبح المسار الرئيسي غير قابل للوصول
- لا يحدث أي تعطيل لجلسة القطر أثناء تبديل المسار
- يتعامل النواة مع اخت��ار المسار تلقائيًا

التكوين

عناوين الاستماع لـ DRA

قم بتكوين عدة عناوين IP محلية للقرين للارتباط بها:

```
%{
  # عنوان IP (متوافق مع الإصدارات السابقة) واحد
  listen_ip: "192.168.1.10",

  # عدة IPs عبر الإرسال لـتعدد SCTP
  listen_ip: ["192.168.1.10", "10.0.0.10"],

  listen_port: 3868,
  ...
}
```

ملاحظات:

- العنوان الأول فقط في القائمة TCP يستخدم النقل عبر
- المحددة IP بجميع عناوين SCTP يرتبط النقل عبر
- الواحدة مدعومًا بالكامل IP يظل تنسيق سلسلة

تكوين القرين

قم بتكوين عدة عناوين IP بعيدة لاتصالات الأقران:

```
peers: [
  %{
    host: "hss01.example.com",
    realm: "example.com",
    ip: "192.168.1.20",                          # IP (مطلوب) الأساسي
    additional_ips: ["192.168.1.20", "10.0.0.20"],        # جميع
IPs لـتعدد الإرسال
    port: 3868,
    transport: :diameter_sctp,
    tls: false,
    initiate_connection: true
  }
]
```

ملاحظات:

- حقل `ip` مطلوب للتوافق مع الإصدارات السابقة
- حقل `ips` اختياري؛ إذا تم حذفه، يتم استخدام `ip` فقط
- لتعدد الإرسال عبر SCTP، قم بتضمين IP الأساسي في قائمة `ips`
- بالنسبة لـ TCP، يتم استخدام `ip` فقط (تعدد الإرسال TCP لا يدعم)

مثال كامل

```
config :dra,
  diameter: %{
    service_name: :omnitouch_dra,
    listen_ip: ["192.168.1.10", "10.0.0.10"],  # DRA متعدد الإرسال
    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: [
      # اتصال HSS متعدد الإرسال
      %{
        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 (متوافق مع الإصدارات السابقة) ذو الإرسال الفردي
      %{
        host: "mme01.example.com",
        realm: "example.com",
        ip: "192.168.1.30",
        port: 3868,
        transport: :diameter_sctp,
        tls: false,
        initiate_connection: false
      }
    ]
  }
```

المتطلبات

- يجب تحميل وحدة النواة SCTP (حزمة `lksctp-tools` على Linux)
- يجب أن تكون جميع عناوين IP قابلة للتوجيه من/إلى القرين
- يجب أن تسمح قواعد جدار الحماية بحركة مرور SCTP على جميع IPs المكونة
- يجب تكوين كلا الطرفين لتعدد الإرسال للحصول على تكرار كامل

القيود

- لا يدعم النقل عبر TCP تعدد الإرسال (يستخدم فقط IP الأساسي)
- قد تكون هناك قيود على التوافق عند استخدام TLS عبر تعدد الإرسال مع SCTP
- يعتمد توقيت التبديل في المسار على معلمات SCTP في النواة

جداول المرجع

3 معرفات التطبيقات الشائعةGPP

معرف التطبيق	الواجهة	الوصف
16777251	S6a/S6d	مصادقة MME/SGSN وبيانات الاشتراك إلى HSS
16777252	S13/S13'	تحقق من هوية المعدات من MME إلى EIR
16777238	Gx	التحكم في السياسة والفوترة من PCEF إلى PCRF
16777267	S9	سياسة التجوال من PCRF المحلي إلى PCRF الزائر
16777272	Sy	ربط جلسة PCRF إلى OCS
16777216	Cx	تسجيل IMS من I-CSCF/S-CSCF إلى HSS
16777217	Sh	بيانات مستخدم IMS من AS إلى HSS
16777236	SLg	خدمات الموقع من MME/SGSN إلى GMLC
16777291	SLh	معلومات موقع المشترك من GMLC إلى HSS
16777302	S6m	MTC-IWF إلى HSS/HLR للأجهزة M2M
16777308	S6c	توجيه SMS من SMS-SC/IP-SM-GW إلى HSS
16777343	S6t	مراقبة الأحداث من SCEF إلى HSS
16777334	Rx	تفويض الوسائط من AF إلى PCRF

رموز AVP الشائعة

الرمز	اسم AVP	النوع	الاستخدام
1	اسم المستخدم	UTF8String	معرف المشترك (IMSI في 3GPP)
264	المضيف الأصلي	DiameterIdentity	اسم مضيف القرين الأصلي
268	رمز النتيجة	Unsigned32	رمز النتيجة القياسي
283	المجال الوجهة	DiameterIdentity	المجال المستهدف
293	المضيف الوجهة	DiameterIdentity	المضيف المستهدف (اختياري)
296	المجال الأصلي	DiameterIdentity	المجال المصدر
297	النتيجة التجريبية	Grouped	رمز النتيجة المحدد من البائع

رموز الأوامر الشائعة

رموز الأوامر هي جزء من رأس رسالة القطر، وليست AVPs:

الرمز	اسم الأمر	الوصف
257	CER/CEA	طلب/إجابة تبادل القدرات
258	RAR/RAA	طلب/إجابة إعادة المصادقة
274	ASR/ASA	طلب/إجابة إنهاء الجلسة
275	STR/STA	طلب/إجابة إنهاء الجلسة
280	DWR/DWA	طلب/إجابة مراقبة الجهاز
282	DPR/DPA	طلب/إجابة فصل القرين
316	ULR/ULA	طلب/إجابة تحديث الموقع (S6a)
317	CLR/CLA	طلب/إجابة إلغاء الموقع (S6a)
318	AIR/AIA	طلب/إجابة معلومات المصادقة (S6a)
321	PUR/PUA	طلب/إجابة تطهير UE (S6a)

وحدة التوجيه المتقدم

توفر وحدة التوجيه المتقدم قدرات توجيه رسائل مرنة قائمة على القواعد مع دعم لظروف مطابقة معقدة.

مهم: تقوم هذه الوحدة بتقييم **حزم طلب القطر الواردة فقط** (ليس حزم الإجابة). تتبع حزم الإجابة توج��ه الجلسة المحدد إلى القرين الأصلي - انظر توجيه الإجابة للحصول على التفاصيل.

التكوين

قم بتمكين الوحدة وحدد قواعد التوجيه في تكوينك:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: <معرف القاعدة>
      match: <نطاق المطابقة>
      filters: [<قائمة الفلاتر>]
      route:
        peers: [<قائمة الأقران>]
```

المعلمات

المعلمة	الوصف
`enabled`	لتفعيل الوحدة `True` تعيين إلى
`rule_name`	معرف فريد لقواعد التوجيه
`match`	كيفية دمج الفلاتر: `:all` (منطق AND - يجب أن تتطابق جميع الفلاتر)، `:any` (منطق OR - يجب أن يتطابق على الأقل فلتر واحد)، `:none` (منطق NOR - لا يمكن أن تتطابق أي فلاتر)
`filters`	قائمة شروط الفلترة (انظر الفلاتر المتاحة)
`route`	إجراء التوجيه (انظر إجراءات التوجيه أدناه)

إجراءات التوجيه

تدعم معلمة `route` إجراءات متعددة:

التوجيه إلى الأقران

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

توجيه إلى أسماء المضيفين المحددين. يجب أن تكون الأقران:

- محددة في تكوين قرين القطر الخاص بـ DRA
- الاسم المضيف بالضبط كما هو مكون (حساسة لحالة الأحرف)
- متصلة حاليًا حتى ينجح التوجيه (يتم تخطي الأقران المنفصلة)

التوجيه إلى AVP المضيف الوجهة

```
route: :destination_host
```

المضيف الوجهة مفقودًا، AVP المضيف الوجهة (293). إذا كان AVP توجيه إلى القرين المحدد في يعود التوجيه إلى السلوك العادي.

إسقاط الحركة

```
route: :drop
```

يسقط الرسالة بهدوء دون إرسال أي رد. استخدم لـ:

- تصفية الحركة وقطعها
- حظر الطلبات غير المرغوب فيها
- تحديد المعدل عن طريق إسقاط الحركة الزائدة

السلوك:

- يتم إسقاط الرسالة في DRA (لا يتم تمريرها)
- لا يتم إرسال رسالة إجابة إلى القرين المطلوب
- ينفذ سلوك Erlang Diameter `:discard`
- المقياس: `diameter_advanced_routing_drop_count_total` (انظر مقاييس بروميثيوس)

توليد استجابة خطأ

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

يولد إجابة خطأ قطرية مع رمز النتيجة المحدد ويرسلها مرة أخرى إلى القرين المطلوب. رموز النتيجة الشائعة:

- `3002` - DIAMETER_UNABLE_TO_DELIVER (التوجيه غير متاح)
- `3003` - DIAMETER_REALM_NOT_SERVED (المجال غير مدعوم)
- `3004` - DIAMETER_TOO_BUSY (حماية من الحمل الزائد، تحديد المعدل)
- `5012` - DIAMETER_UNABLE_TO_COMPLY (رفض عام)

السلوك:

- يقوم DRA بتوليد إجابة خطأ مع رمز النتيجة المحدد
- تتضمن الإجابة Origin-Host وOrigin-Realm وSession-Id (تملأ تلقائيًا بواسطة Diameter)
- لا يتم تمرير الرسالة إلى أي قرين
- ينفذ Erlang Diameter `{:protocol_error, code}` (معادل لـ `{:answer_message, code}`)
- المقياس: `diameter_advanced_routing_error_count_total` (انظر مقاييس بروميثيوس)

الفلاتر المتاحة

الفلاتر القياسية

متاحة في كل من التوجيه المتقدم و التحويل المتقدم:

- **`:application_id`** - تطابق معرف تطبيق القطر (انظر مرجع معرف التطبيق)
  - قيمة واحدة: `{:application_id, 16777251}` (S6a/S6d)
  - قيم متعددة: `{:application_id, [16777251, 16777252]}` (S6a أو S6b)
- **`:command_code`** - تطابق رمز أمر القطر
  - قيمة واحدة: `{:command_code, 318}` (طلب AIR)
  - قيم متعددة: `{:command_code, [317, 318]}` (ULR أو AIR)
- **`:avp`** - تطابق قيمة AVP (انظر مرجع رمز AVP)
  - تطابق دقيق: `{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}`
  - تطابق تعبير عادي: `{:avp, {1, ~r"999001.*"}}`

    - أنماط متعددة: `{:avp, {1, ["505057001313606", ~r"999001.*", ~r"505057.*"]}}`
    - أي قيمة (تحقق من الوجود): `{:avp, {264, :any}}`

فلتر خاص بالتوجيه

التوجيه المتقدم متاح فقط في:

- `:via_peer` - تطابق القرين الذي تم استلام الطلب منه
    - قرين واحد: `{:via_peer, "omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org"}`
    - عدة أقران: `{:via_peer, ["omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org", "omnitouch-lab-dra02.epc.mnc001.mcc001.3gppnetwork.org"]}`
    - أي قرين: `{:via_peer, :any}`

فلاتر خاصة بالتحويل

التحويل المتقدم متاحة فقط في:

- `:to_peer` - تطابق على القرين الوجهة المحددة مسبقًا (حزم الطلب فقط)
    - قرين واحد: `{:to_peer, "dra01.omnitouch.com.au"}`
    - عدة أقران: `{:to_peer, ["dra01.omnitouch.com.au", "dra02.omnitouch.com.au"]}`
- `:from_peer` - تطابق القرين الذي أرسل الإجابة (حزم الإجابة فقط)
    - قرين واحد: `{:from_peer, "hss-01.example.com"}`
    - عدة أقران: `{:from_peer, ["hss-01.example.com", "hss-02.example.com"]}`
- `:packet_type` - تطابق اتجاه الحزمة
    - طلب: `{:packet_type, :request}`
    - إجابة: `{:packet_type, :answer}`

ملاحظات مهمة حول الفلاتر

- **فلاتر AVP**: البسيطة (اسم المستخدم، المضيف الأصلي، AVPs موصى بها فقط لـ: المجال الوجهة، إلخ.)
  - لا تدعم AVPs **ولن تتطابق** المجمعة
  - لا تدعم القيم الثنائية المعقدة **ولن تتطابق**
  - استخدم التنسيق: `{:avp, {code, value}}`
- **عوامل التشغيل القائمة**: `:packet_type` مدعومة لجميع قيم الفلتر باستثناء
  - داخل القائمة **OR** عند استخدام قائمة، يتم تطبيق **منطق**
  - مثال: `{:command_code, [317, 318]}` تطابق رمز الأمر 317 **أو** 318
- **القيم الخاصة**:
  - `:any` - تطابق أي قيمة (يتحقق من وجود AVP)
  - مثال: `{:avp, {264, :any}}` تطابق إذا كان AVP المضيف الأصلي موجودًا بأي قيمة

أمثلة التوجيه

المثال 1: توجيه عبر القرين

توجيه الرسائل بناءً على أي DRA وصلت منها:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: مؤقت حتى التحويل S6a عبر إلى HSS المحلي
      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]
```

كيف يعمل: يوجه حركة S6a التي تصل عبر أقران DRA المحددة إلى عقد HSS المحلية.

المثال 2: التجوال الوارد مع مطابقة الأنماط

توجيه حركة التجوال بناءً على أنماط IMSI:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: التجوال الوارد S6a إلى 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]
```

كيف يعمل: يوجه رسائل S6a المطابقة إلى أقران IMSI من المجال الأصلي المحدد مع أنماط DRA المعينة.

المثال 3: توجيه ديناميكي مع destination_host:

توجيه إلى قيمة AVP المضيف الوجهة في الرسالة:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: التوجيه إلى المضيف الوجهة المحدد
      match: ":all"
      filters:
        - '{:avp, {1, [~r"90199.*"]}}'  # تطابق نمط IMSI
      route: :destination_host
```

كيف يعمل:

- المضيف الوجهة (293) AVP عندما تتطابق الفلاتر، يتم التوجيه إلى القرين المحدد في
- إذا كان AVP المضيف الوجهة مفقودًا، يعتبر التطابق فشلًا ويعود إلى التوجيه العادي
- مفيد لتوجيه الشرف عندما يحدد المرسل الوجهة الدقيقة

المثال 4: إسقاط الحركة غير المرغوب فيها

إسقاط الحركة من نطاقات IMSI المحددة:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: إسقاط المشتركين التجريبيين
      match: ":all"
      filters:
        - '{:application_id, 16777251}'  # S6a
        - '{:avp, {1, [~r"999999.*"]}}'  # نطاق IMSI التجريبي
      route: :drop
```

كيف يعمل:

- تطابق رسائل S6a مع IMSI 999999 التي تبدأ بـ
- يسقط الرسالة بهدوء دون إرسال أي رد
- مفيد لتصفية حركة الاختبار أو حظر نطاقات مشتركة محددة
- انظر مقاييس بروميثيوس لمراقبة الحركة المفقودة

المثال 5: تحديد المعدل مع استجابات الخطأ

إرجاع DIAMETER_TOO_BUSY لأنماط حركة معينة:

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

كيف يعمل:

- تطابق حركة S6a من القرين المحدد الذي يعاني من الحمل الزائد
- إرجاع استجابة خطأ DIAMETER_TOO_BUSY (3004)

- يتلقى القرين المطلوب خطأ ويجب أن يتراجع
- مفيد لحماية الحمل الزائد وتحديد المعدل
- انظر مقاييس بروميثيوس لمراقبة استجابات الخطأ

المثال 6: استجابات الخطأ الشرطية حسب الأمر

حظر أنواع أوامر معينة مع رموز الخطأ المناسبة:

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: حظر طلبات التطهير
      match: ":all"
      filters:
        - '{:application_id, 16777251}'  # S6a
        - '{:command_code, 321}'         # PUR (طلب تطهير UE)
      route: {:error, 5012}
```

كيف يعمل:

- تطابق رسائل S6a طلبات تطهير UE
- إرجاع DIAMETER_UNABLE_TO_COMPLY (5012) كخطأ
- حظر عمليات معينة دون إسقاط الحركة بهدوء
- مفيد لتعطيل أوامر القطر المحددة

وحدة التحويل المتقدم

تمكن وحدة التحويل المتقدم من تعديل ديناميكي لقيم AVP في رسائل القطر بناءً على معايير المطابقة. انظر معالجة القواعد للحصول على تفاصيل حول كيفية تقييم القواعد.

التكوين

قم بتمكين الوحدة وحدد قواعد التحويل:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: <معرف القاعدة>
      match: <نطاق المطابقة>
      filters: [<قائمة الفلاتر>]
      transform:
        action: <إجراء التحويل>
        avps: [<تعديلات AVP>]
```

المعلمات

المعلمة	الوصف
`enabled`	لتفعيل الوحدة `True` تعيين إلى
`rule_name`	معرف فريد لقواعد التحويل
`match`	كيفية دمج الفلاتر: `:all` (منطق AND)، `:any` (منطق OR)، `:none` (منطق NOR) - انظر منطق الفلتر
`filters`	قائمة شروط الفلترة (انظر الفلاتر المتاحة)
`transform.action`	نوع التحويل (`:edit`، `:remove`، أو `:overwrite`)
`transform.avps`	قائمة بتعديلات AVP التي يجب تطبيقها (انظر مرجع رمز AVP)

إجراءات التحويل

حزم الطلب (طلبات القطر)

- **`:edit`** - تعديل قيم AVP الموجودة
  - يعدل فقط AVPs التي توجد في الرسالة
  - إذا لم يكن AVP موجودًا، فلا يتم إجراء أي تغيير
- **`:remove`** - إزالة AVPs من الرسالة
- **`:overwrite`** - استبدال هياكل AVP بالكامل

    - يتطلب معلمة `dictionary` (مثل تحدد القاموس القطر `:diameter_gen_3gpp_s6a`)

حزم الإجابة (إجابات القطر)

- `:remove` - إزالة AVPs من الرسالة
- `:overwrite` - استبدال هياكل AVP بالكامل
    - يتطلب معلمة `dictionary`

مهم: إذا لم تتطابق أي قواعد، يتم تمرير الحزمة عبر الشفافية دون أي تحويلات.

بناء جملة تعديل AVP

تعديل قياسي:

- `{:avp, {<code>, <new_value>}}` - تعيين AVP إلى قيمة جديدة

إزالة AVPs:

- `{:avp, {<code>, :any}}` - إزالة AVP حسب ID (تتم الإزالة بغض النظر عن القيمة الحالية)
- ملاحظة: يتم دعم الإزالة بناءً على محتوياتavp_id ؛ لا يتم دعم الإزالة بناءً على AVP

استبدال باستخدام القاموس:

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

أمثلة التحويل

المثال 1: إعادة كتابة المجال الوجهة بناءً على القرين

إعادة كتابة المجال الوجهة بناءً على المكان الذي يتم توجيه الرسالة إليه:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: إعادة كتابة المجال الوجهة S6a لـ 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"}}'
```

كيف يعمل: عندما يتم توجيه طلبات S6a إلى أقران DRA محددة وتتطابق مع نمط IMSI، يتم إعادة كتابة المجال الوجهة لشبكة المشغل X.

المثال 2: توجيه متعدد الناقلين مع التحويلات

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: إعادة كتابة المجال الوجهة S6a لشريك التجوال
الأسترالي
      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"}}'
```

كيف يعمل: يوجه نطاقات مشتركي IMSI المختلفة إلى المجالات الشبكية المناسبة بناءً على أنماط IMSI. (انظر ترتيب التنفيذ) تفوز أول قاعدة مطابقة.

المثال 3: إعادة كتابة مجال MVNO

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: إعادة كتابة المجال الوجهة S6a لمشترك فردي
      match: ":all"
      filters:
        - '{:to_peer, ["dra01.omnitouch.com.au",
"dra02.omnitouch.com.au"]}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
        - '{:avp, {1, ["505057000003606"]}}'  # تطابق IMSI دقيق
      transform:
        action: ":edit"
        avps:
          - '{:avp, {283, "epc.mnc001.mcc001.3gppnetwork.org"}}'
```

كيف يعمل: يحول المجال الوجهة لمشترك MVNO المحدد إلى شبكتهم الأساسية المستضافة.

المثال 4: تحويل فقط مع فلتر نوع الحزمة

تحويل حزم الطلب فقط (ليس الإجابات):

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: قاعدة تعليم AIR
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
        - '{:command_code, 318}'
        - '{:packet_type, :request}'
        - '{:avp, {1, "999999000000001"}}'
        - '{:avp, {264, :any}}'  # يجب أن يوجد المضيف الأصلي بأي
قيمة
      transform:
        action: ":edit"
        avps:
          - '{:avp, {1, "999999000000002"}}'
```

كيف يعمل:

- **الطلبات** (ليس حزم الإجابة) S6a AIR يطابق فقط حزم

- يتحقق من أن اسم المستخدم (AVP 1) يساوي "999999000000001"
- يتحقق من أن المضيف الأصلي موجود بأي قيمة (AVP 264)
- يعيد كتابة اسم المستخدم إلى "999999000000002"
- إذا لم يكن AVP موجودًا، فلا يتم إجراء أي تغيير

المثال 5: إزالة AVP

إزالة AVP محدد من الرسائل:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: إزالة AVP اسم المستخدم
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
      transform:
        action: ":remove"
        avps:
          - '{:avp, {1, :any}}'  # إزالة اسم المستخدم بغض النظر عن القيمة
```

كيف يعمل: يزيل AVP اسم المستخدم (الرمز 1) من جميع رسائل S6a، بغض النظر عن قيمته الحالية.

المثال 6: استبدال AVP المجمعة في حزم الإجابة

تعديل AVPs المجمعة المعقدة في حزم الإجابة باستخدام إجراء `:overwrite` مع دعم القاموس:

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: إضافة APN SOS إلى ULA
      match: ":all"
      filters:
        - '{:application_id, 16777251}'       # S6a/S6d
        - '{:command_code, 316}'              # ULA (إجابة تحديث
الموقع)
        - '{:packet_type, :answer}'           # حزم الإجابة فقط
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'  #
المجال الأصلي
      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, [], [],
[]}],
```

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

كيف يعمل:

- يطابق حزم S6a إجابة تحديث الموقع (ULA) من مجال أصلي محدد
- يستخدم إجراء `:overwrite` لاستبدال AVP المجمعة بالكامل
- **يتطلب معلمة** `dictionary` لترميز هياكل AVP المجمعة المعقدة بشكل صحيح
- يضيف ثلاث تكوينات APN: "internet" (السياق 1)، "ims" (السياق 2)، و"sos" (السياق 3)
- يتضمن كل APN ملفات تعريف QoS وحدود النطاق الترددي (AMBR) وإعدادات نوع PDN
- يضمن التحويل توفير APN خدمات الطوارئ (SOS) لجميع المشتركين من هذا المجال

متى تستخدم `:overwrite` **مع القاموس:**

- تعديل AVPs المجمعة ذات الهياكل المتداخلة (مثل APN-Configuration-Profile)
- إضافة أو إعادة هيكلة بيانات الاشتراك المعقدة من 3GPP
- التعامل مع تعقيد AVP عندما لا يمكن لـ `:edit`
- يجب أن يتطابق القاموس مع تطبيق القطر (`:diameter_gen_3gpp_s6a` لـ S6a، إلخ.)

ملاحظات مهمة:

- `:overwrite` يستبدل AVP بالكامل، وليس فقط الحقول الفردية
- يجب أن تتطابق بنية AVP مع تعريف القاموس تمامًا
- ستؤدي البنية غير الصحيحة إلى فشل الترميز وإسقاط الحزم
- هذه ميزة متقدمة - تحقق منها بدقة في بيئة الاختبار أولاً

حالات الاستخدام

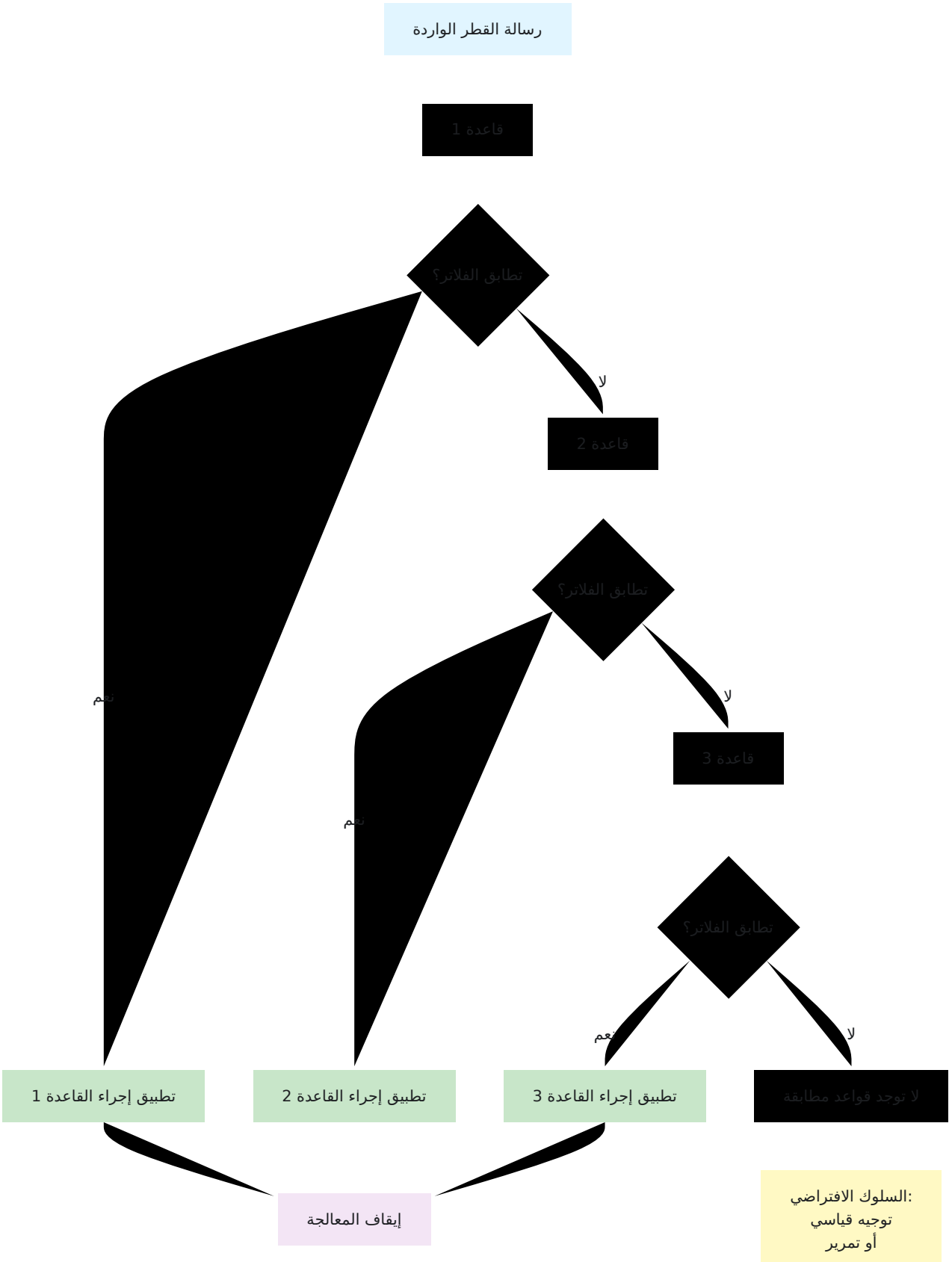
- **دعم MVNO**: توجيه حركة مشغل افتراضي إلى الشبكات الأساسية المستضافة
- **ترحيل الشبكة**: إعادة توجيه المشتركين تدريجيًا إلى بنية تحتية جديدة
- **ترجمة المجال**: التحويل بين مخططات التسمية المختلفة لشركاء التجوال

- **التعددية**: عزل مجموعات المشتركين حسب المجال
- **توجيه الناقل** IMSI: توجيه الحركة إلى الشبكات الصحيحة بناءً على نطاقات

معالجة القواعد

تنطبق على كل من التوجيه المتقدم و التحويل المتقدم الوحدات.

ترتيب التنفيذ



1. يتم تقييم القواعد **بالترتيب من الأعلى إلى الأسفل** كما هو محدد في التكوين
2. يتم تقييم الفلاتر داخل القاعدة بناءً على معلمة `match` (`:all`، `:any`، أو `:none`)

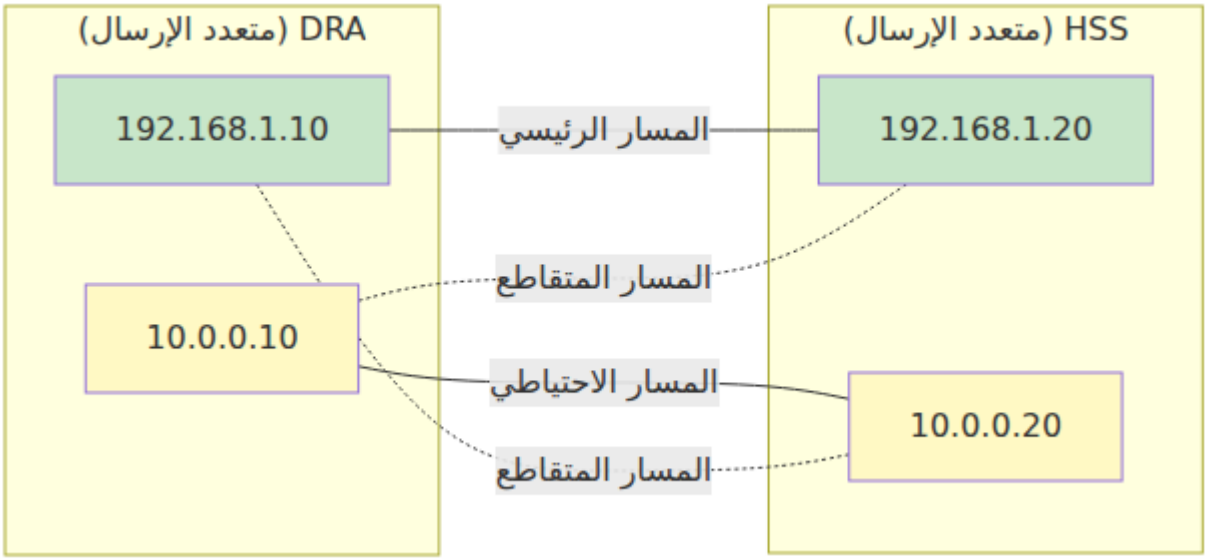
3. **تفوز أول قاعدة مطابقة** - لا يتم تقييم القواعد اللاحقة
4. إذا لم تتطابق أي قواعد، يتم استخدام سلوك التوجيه/التمرير الافتراضي

منطق الفلتر

كيفية دمج الفلاتر `match` تحدد معلمة:

match: :all (منطق AND)

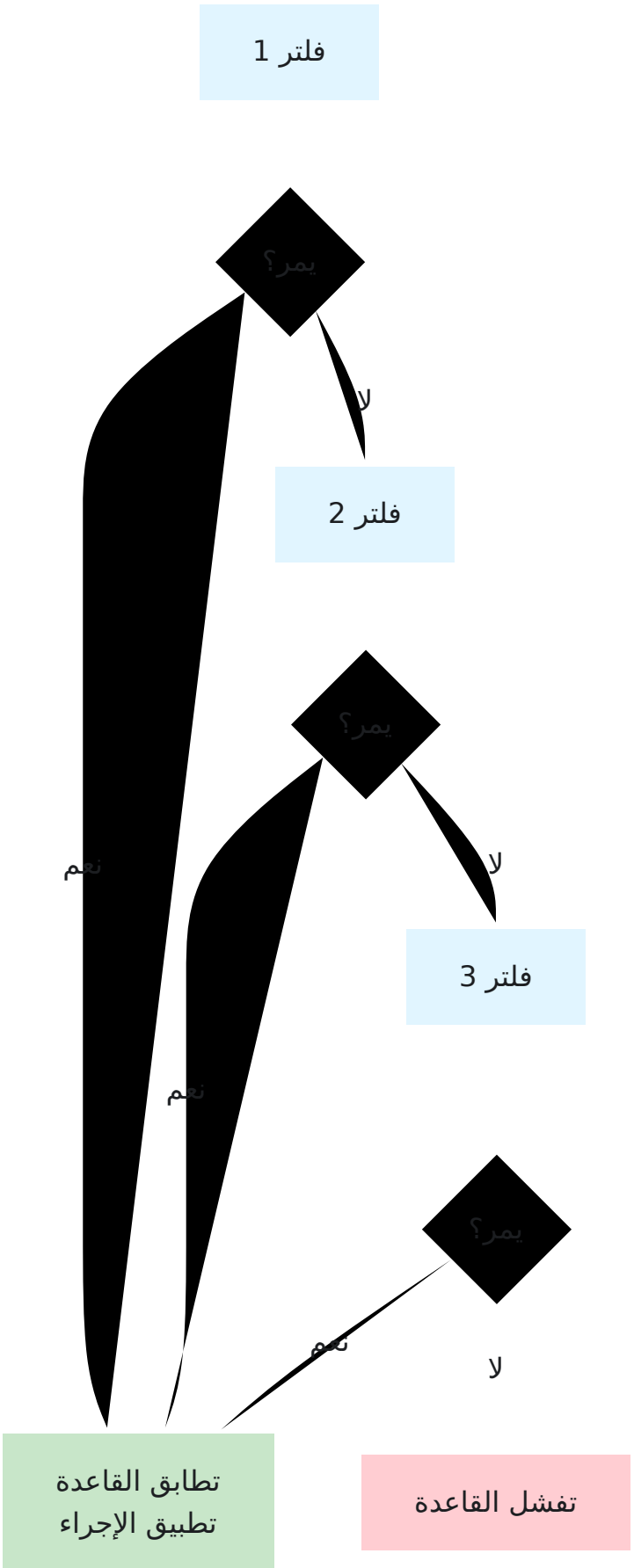
يجب أن تتطابق جميع الفلاتر حتى تنجح القاعدة.



مثال: مع 3 فلاتر، يجب أن تكون `filter1 AND filter2 AND filter3` جميعها صحيحة.

match: :any (منطق OR)

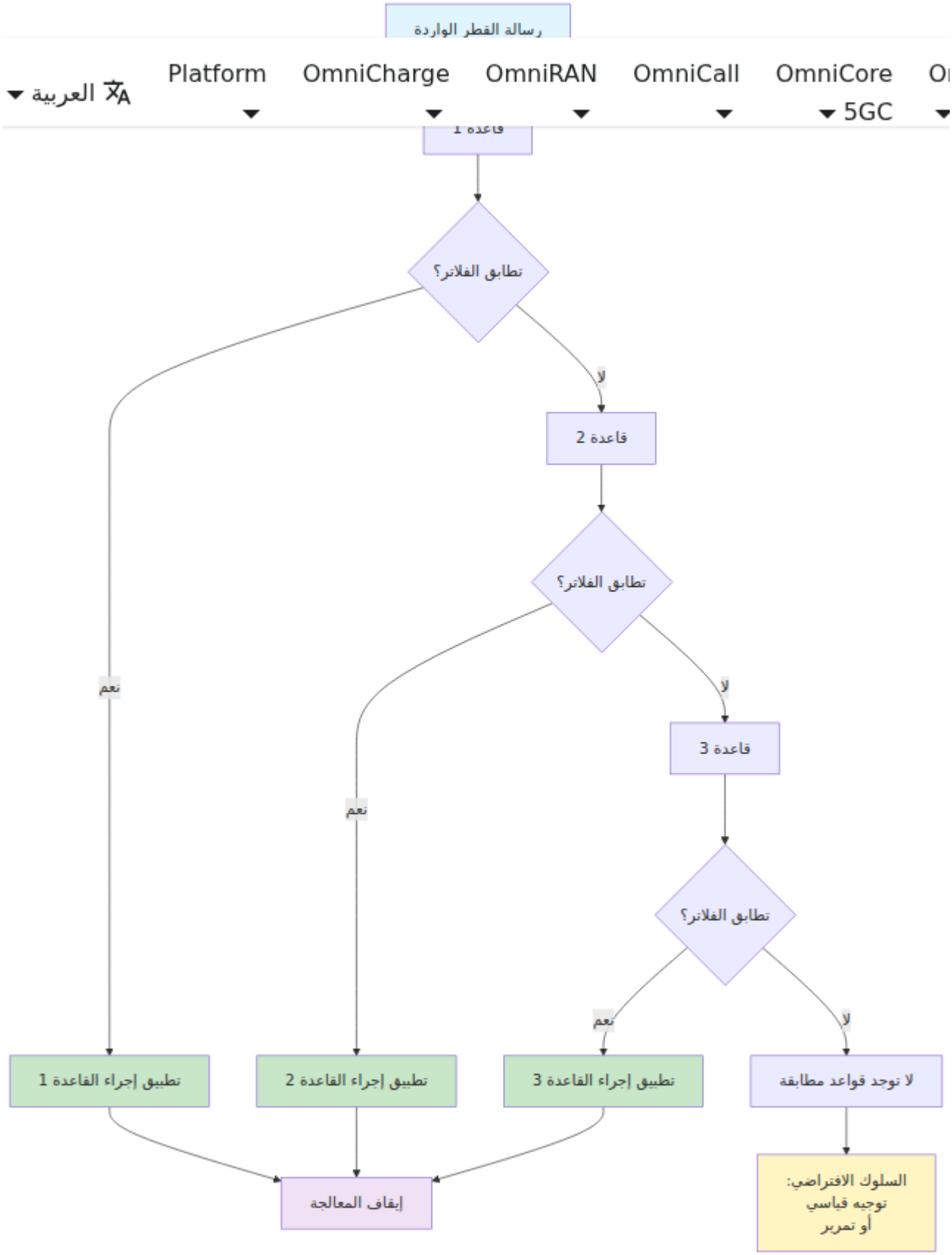
يجب أن يتطابق فلتر واحد على الأقل حتى تنجح القاعدة.



مثال: مع 3 فلاتر، يجب أن تكون `filter1 OR filter2 OR filter3` (يمر أي واحد).

match: :none (منطق NOR)

لا يمكن أن تتطابق أي فلاتر حتى تنجح القاعدة (مطابقة عكسية).



مثال: مع 3 فلاتر، يجب أن تكون `NOT filter1 AND NOT filter2 AND NOT filter3` (يجب أن تفشل جميعها).

ملاحظات إضافية:

مثل) عند استخدام عوامل التشغيل القائمة داخل قيمة الفلتر `{:avp, {1, ["value1", "value2"]}}`)، تستخدم القيم منطق **OR** (يمكن أن تتطابق أي منها).

أنماط التعبير العادي

استخدم بناء جملة `~r"pattern"` لمطابقة التعبيرات العادية:

- `~r"999001.*"` - تطابق IMSI التي تبدأ بـ 999001
- `~r"^310[0-9]{3}.*"` - تطابق IMSI مع أنماط MNC محددة
- `~r".*test$"` - تطابق القيم التي تنتهي بـ "test"

أفضل الممارسات

1. **التحديد**: ترتيب القواعد من الأكثر تحديدًا إلى الأكثر عمومية
2. **الأداء**: وضع المطابقات الأكثر شيوعًا أولاً لتقليل الحمل المعالج
3. **الاختبار**: تحقق من أنماط التعبير العادي قبل النشر
4. **التوثيق**: استخدم قيم `rule_name` الوصفية للوضوح التشغيلي
5. **المراقبة**: تتبع معدلات مطابقة القواعد للتحقق من السلوك المتوقع

وحدة المقاييس الموسعة

توفر وحدة المقاييس الموسعة قدرات متقدمة في القياس والتحليلات لتحليل أنماط حركة القطر تتجاوز المقاييس القياسية.

التكوين

قم بتمكين الوحدة وتكوين أنواع المقاييس المحددة:

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

المعلمات

المعلمة	الوصف
`enabled`	لتفعيل وحدة المقاييس `true` تعيين إلى الموسعة
`attach_attempt_reporting_enabled`	تمكين تتبع وتقرير محاولات الاتصال LTE (S6a AIR/AIA)

المقاييس المتاحة

تتبع محاولات الاتصال

يتتبع محاولات اتصال المشترك LTE من خلال مراقبة أزواج رسائل طلب معلومات المصادقة (AIR) والإجابة (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'

المحاولة مجددا

القياس: `attach_attempt_count`

الحقول:

- `imsi` - IMSI المشترك (من AVP رموز انظر - اسم المستخدم AVP)

العلامات:

- `origin_host` - القرين الذي نشأ منه طلب الاتصال
- `result_code` - رمز النتيجة من استجابة HSS

كيف يعمل:

1. عند استلام طلب AIR (S6a 16777251 - رمز الأمر 318، تطبيق انظر معرفات التطبيقات)، تستخرج الوحدة:
   - معرف من طرف إلى طرف للربط بين الطلب والإجابة
   - IMSI (رمز AVP 1 اسم المستخدم)
   - المضيف الأصلي (رمز AVP 264)
2. يتم تخزين بيانات الطلب في ETS مع TTL
3. عند استلام AIA المطابقة، تقوم الوحدة:
   - بربط باستخدام معرف من طرف إلى طرف
   - استخراج رمز النتيجة (رمز النتيجة التجريبية AVP 297 أو AVP 268)
   - إصدار المقياس مع IMSI ورمز النتيجة والمضيف الأصلي

حالات الاستخدام

- **تحليل معدل نجاح الاتصال** - تتبع محاولات الاتصال الناجحة مقابل الفاشلة حسب رمز النتيجة
- **استكشاف الأخطاء على مستوى IMSI** - تحديد المشتركين الذين يعانون من فشل الاتصال
- **مراقبة أداء الشبكة** (MME/SGSN) - مراقبة أنماط محاولات الاتصال حسب الأصل
- **تحليلات التجوال** - تحليل معدلات نجاح الاتصال التجوال الوارد

التكامل

تُصدَّر المقاييس الموسعة عبر تكامل InfluxDB:

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

رموز النتيجة هي رموز قطر قياسية:

- `2001` - النجاح (DIAMETER_SUCCESS)
- `5001` - فشل المصادقة (DIAMETER_AUTHENTICATION_REJECTED)
- `5004` - AVP غير مدعوم
- انظر RFC 6733 للحصول على قائمة كاملة برموز النتيجة

ملاحظات مهمة

- تتبع مقاييس محاولات الاتصال فقط أزواج AIR/AIA S6a (معرف التطبيق 16777251، رمز الأمر 318)
- تنتهي صلاحية بيانات الطلب بناءً على مهلة الطلب المكونة + 5 ثوانٍ
- تتم معالجة المقاييس بشكل غير متزامن (عملية منفصلة) لتجنب حظر تدفق الرسائل
- تعمل الوحدة بشكل مستقل عن وحدات التوجيه والتحويل

مقاييس بروميثيوس

مقاييس بروميثيوس شاملة لمراقبة حركة القطر وصحة الأقران وعمليات الوحدات. DRA يعرض جميع المقاييس متاحة في نقطة النهاية `/metrics`.

مقاييس القطر الأساسية

حالة القرين

المقياس: `diameter_peer_status` **النوع**: Gauge (1) **الوصف**: ما إذا كان القرين متصلًا أم لا (0) **العلامات**:

- `origin_host` - هوية القطر للقرين
- `ip` - عنوان IP للقرين

مثال:

```
# تحقق مما إذا كان القرين المحدد متصلًا
diameter_peer_status{origin_host="hss01.example.com"}

# عد الأقران المنفصلين
count(diameter_peer_status == 0)
```

عدد الرسائل

الوصف: العدد الإجمالي لرسائل القطر المتبادلة مع الأقران **النوع**: Counter **المقياس**: `diameter_peer_message_count_total` **العلامات**:

- `origin_host` - هوية القطر للقرين
- `received_from` - القرين الذي تم استلام الرسالة منه
- `application_id` - معرف تطبيق القطر (انظر مرجع معرف التطبيق)
- `cmd_code` - رمز أمر القطر (انظر رموز الأوامر الشائعة)
- `application_name` - اسم التطبيق القابل للقراءة (مثل "3GPP_S6a")
- `cmd_name` - اسم الأمر القابل للقراءة (مثل "AIR")
- `direction` - "طلب" أو "استجابة"

مثال:

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

# إجمالي معدل الرسائل حسب التطبيق
sum by (application_name)
(rate(diameter_peer_message_count_total[5m]))
```

رموز نتيجة الاستجابة

الوصف: العدد الإجمالي للاستجابات القطرية حسب رمز النتيجة **النوع**: Counter **المقياس**: `diameter_peer_message_result_code_count_total` **العلامات**:

- `origin_host` - طالب الأصلي
- `routed_to` - القرين الذي أرسل الإجابة
- `application_id` - معرف تطبيق القطر
- `cmd_code` - رمز أمر القطر
- `application_name` - اسم التطبيق
- `cmd_name` - اسم الأمر
- `result_code` - رمز النتيجة القطرية أو رمز النتيجة التجريبية

مثال:

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

# معدل الأخطاء حسب رمز النتيجة
sum by (result_code) (
  rate(diameter_peer_message_result_code_count_total{
    result_code!="2001"
  }[5m])
)
```

رموز النتيجة الشائعة:

- `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

تأخير الاستجابة

المقياس: `diameter_peer_last_response_delay` **النوع**: Gauge **الوصف**: تأخير الاستجابة الأكثر حداثة بالمللي ثانية (DRA → قرين → DRA) **العلامات**:

- `origin_host` - طالب الأصلي
- `routed_to` - القرين الذي أرسل الإجابة
- `application_name` - اسم التطبيق
- `cmd_name` - اسم الأمر

مثال:

```
# متوسط وقت الاستجابة من HSS
avg(diameter_peer_last_response_delay{routed_to="hss01.example.com"})

# وقت الاستجابة لـ P95 S6a
histogram_quantile(0.95,
  rate(diameter_peer_last_response_delay{application_name="3GPP_S6a"}
[5m])
)
```

الطلبات غير المستجابة

المقياس: `diameter_peer_unanswered_request_count_total` **النوع**: Counter
الوصف: الطلبات المرسلة ولكن لم يتم الرد عليها ضمن فترة المهلة **العلامات**:

- `origin_host` - طالب الأصلي
- `routed_to` - القرين الذي لم يرد
- `application_id` - معرف تطبيق القطر
- `cmd_code` - رمز أمر القطر
- `application_name` - اسم التطبيق
- `cmd_name` - اسم الأمر

مثال:

```
# معدل الطلبات غير المستجابة
rate(diameter_peer_unanswered_request_count_total[5m])

# تحديد الأقران المشكلة
topk(5, sum by (routed_to) (
  rate(diameter_peer_unanswered_request_count_total[5m])
))
```

محاولات الاتصال غير المصرح بها

المقياس: `diameter_peer_unauthorized_connection_count_total` **النوع**:
Counter **العلامات** **الوصف**: محاولات الاتصال من الأقران غير المصرح لهم:

- `origin_host` - هوية القرين غير المصرح له
- `supported_applications` - التطبيقات المعلنة من قبل القرين

- `peer_ip` - عنوان IP لمحاولة الاتصال

مثال:

```
# محاولات الاتصال غير المصرح بها
rate
```