دليل واجهة برمجة التطبيقات REST

← العودة إلى الوثائق الرئيسية

يوفر هذا الدليل وثائق شاملة لواجهة برمجة التطبيقات **REST API** و **Swagger UI** الخاصة بـ OmniSS7.

جدول المحتويات

1. نظرة عامة
2. تكوين خادم HTTP
3. Swagger UI
4. نقاط نهاية واجهة برمجة التطبيقات
5. أمثلة عملية
6. المصادقة
7. تنسيقات الاستجابة
8. معالجة الأخطاء
9. المقاييس (بروميثيوس)
10. طلبات أمثلة

نظرة عامة

تقوم OmniSS7 بتوفير واجهة برمجة تطبيقات REST لتنفيذ عمليات MAP على **المنفذ 9090** بالإضافة إلى حقن SS7، عبر شبكة CAMEL (جزء تطبيق CAP) و (جزء تطبيق الهاتف المحمول) SCCP/M3UA. تحدد واجهة برمجة التطبيقات **47 نقطة نهاية عملية** الخام للاختبار `POST` إلى جانب المقاييس ونقاط نهاية Swagger. يسمح لك بـ `/api/*`:

- تشغيل طلبات MAP (التنقل، الرسائل القصيرة، المصادقة، بيانات المشترك، الخدمات الإضافية، GPRS، المعدات، خدمات الموقع)

- إدارة حوارات التحكم في المكالمات CAP/CAMEL
- حقن حمولات SCCP/M3UA الخام
- مراقبة مقاييس النظام عبر بروميثيوس

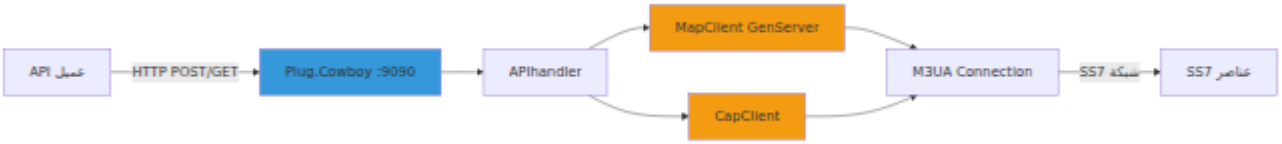
يشير هذا الدليل إلى **دليل** MAP، للحصول على مخططات الطلب/الاستجابة لكل عملية وأكواد **عميل MAP**، الذي يوثق كل عملية بالتفصيل.

المنافذ والنقل

تقوم OmniSS7 **ثلاثة خوادم** بتشغيل **HTTP** يتم (9090:). يغطي هذا الدليل الأول **متميزة**. توثيق الآخرين في مرجع التكوين.

المنفذ	النقل	الخادم	نطاق هذا الدليل
9090	HTTP (`Plug.Cowboy`)	`APIhandler`	**نعم.** MAP/CAP REST `/api/*`، `/metrics`، `/swagger`.
8087	HTTPS (`Bandit`، Phoenix)	`ControlPanelWeb.Endpoint`	لا: واجهة التحكم في الويب. انظر دليل واجهة الويب.
8445	HTTPS / TLS	`:api_ex` وحدات التحكم REST	لا: حالة/توجيه/روابط REST. انظر مرجع التكوين.

بنية واجهة برمجة التطبيقات



تكوين خادم HTTP

تفاصيل الخادم

تتم خدمة نقاط النهاية HTTP بواسطة مستمع `/swagger` و `/metrics` و `/api/*` `Plug.Cowboy`.

المعامل	القيمة	قابل للتكوين
البروتوكول	HTTP	لا
عنوان IP	0.0.0.0 (جميع الواجهات)	عبر الكود فقط
المنفذ	9090	نعم: `config :omniss7, api_port: <port>` (`9090` الافتراضي)
النقل	`Plug.Cowboy`	لا

رابط الوصول: `http://[server-ip]:9090`

> هي خوادم REST `:api_ex` (:8445) ووحدات التحكم (Bandit، :8087) لوحة التحكم منفصلة. انظر المنافذ والنقل.

تمكين/تعطيل خادم HTTP

سيبدأ HTTP :9090 تحكم فيما إذا كان خادم:

```
config :omniss7,
  start_http_server: true  # تعيين إلى false لتعطيل
```

الافتراضي: `true` (مفعل)

عند التعطيل: لن يبدأ خادم HTTP :9090، وستكون واجهة برمجة التطبيقات REST / واجهة `:api_ex` نقطة النهاية للمقاييس غير متاحة. لوحة التحكم (:8087) وخوادم Swagger / (:8445) تكون محمية بواسطة `start_web_interfaces` غير متأثرة).

Swagger UI

تتضمن واجهة برمجة التطبيقات **Swagger UI** لوثائق تفاعلية للاختبار.

الوصول إلى Swagger UI

الرابط: `http://[server-ip]:9090/swagger`

الميزات:

- وثائق واجهة برمجة تطبيقات تفاعلية
- وظيفة "جربها" لاختبار نقاط النهاية
- مخططات الطلب/الاستجابة
- حمولات أمثلة

Swagger JSON

متاحة على OpenAPI مواصفة:

الرابط: `http://[server-ip]:9090/swagger.json`

حالات الاستخدام:

- أو عملاء واجهة برمجة التطبيقات الآخرين Postman استيراد إلى
- توليد مكتبات العملاء
- أتمتة وثائق واجهة برمجة التطبيقات

نقاط نهاية واجهة برمجة التطبيقات

تتبع كل نقطة نهاية عملية نمط `POST /api/{operation}` وتقبل جسم طلب JSON. الطلبات **متزامنة**: يقوم المعالج بحظر حتى تصل استجابة SS7 أو تنتهي العملية (الافتراضي **10 ثوانٍ**؛ انظر معالجة الأخطاء المعلقة). MAP للعمليات

تسرد الجداول أدناه جميع **47** نقطة نهاية عملية مقسمة حسب الفئة. للحصول على حقول الطلب الذي ، **MAP** انظر القسم المقابل من **دليل عميل** لكل عملية، ومخطط الاستجابة، ورمز MAP، تم تحديثه جنبًا إلى جنب مع هذا الدليل ويوثق كل عملية بالكامل.

> **رموز الاستجابة (جميع نقاط النهاية):** `200` = نجاح، `400` = طلب غير صالح/خطأ (ضمن مهلة العملية SS7 لا استجابة) (حقول مفقودة أو خاطئة النوع)، `504` = انتهاء المهلة (USSD)، `500` = خطأ في العملية (خطأ غير متعلق بالمهلة)، `503` = ميزة غير مفعلة = مسار غير معروف. انظر معالجة الأخطاء.

إدارة التنقل

العملية	نقطة النهاية	الوصف
تحديث الموقع	`POST /api/updateLocation`	تسجيل VLR الخادم لمشترك مع HLR. في وضع HLR، يؤدي ذلك إلى تشغيل تسلسل InsertSubscriberData (ISD).
إلغاء الموقع	`POST /api/cancelLocation`	توجيه VLR لحذف سجل مشترك.
توفير رقم التجوال	`POST /api/prn`	الحصول على MSRN من MSC الخادم.
تطهير MS	`POST /api/purgeMS`	وضع علامة على مشترك كـ "مُطهر" في HLR.
إرسال التعريف	`POST /api/sendIdentification`	استرداد بيانات IMSI/من VLR المصادقة السابق (TMSI كـ مقدمة hex).
إعادة تعيين	`POST /api/reset`	إعادة تعيين initiated من HLR نح VLR (غير مؤكد: يرجع `{"status":"sent"}`).
استعادة البيانات	`POST /api/restoreData`	استعادة بيانات المشترك من VLR يطلب HLR.

الرسائل القصيرة

العملية	نقطة النهاية	الوصف
إرسال معلومات التوجيه للرسالة القصيرة	`POST /api/sri-for-sm`	استعلام HLR عن العقدة الخادمة لتسليم الرسائل القصيرة.
إعادة توجيه الرسالة القصيرة MT	`POST /api/MT-forwardSM`	تسليم رسالة قصيرة منتهية إلى MSC/SGSN الخادم (`smsPDU` كـ hex).
إعادة توجيه الرسالة القصيرة MO	`POST /api/forwardSM`	تقديم رسالة قصيرة منشأة من الهاتف المحمول (`smsPDU` كـ hex).
إرسال الرسالة القصيرة	`POST /api/sendSM`	بناء وتسليم SMS-DELIVER من نص GSM-7.
بناء PDU التسليم	`POST /api/deliverPDU`	بناء PDU SMS-DELIVER وإرجاع hex (بدون إرسال SS7).
تقرير حالة تسليم الرسالة القصيرة	`POST /api/reportSM-DeliveryStatus`	الإبلاغ عن نتيجة تسليم الرسالة القصيرة إلى HLR.
جاهز للرسالة القص��رة	`POST /api/readyForSM`	إبلاغ HLR أن MS متاح للرسائل القصيرة.
تنبيه مركز الخدمة	`POST /api/alertServiceCentre`	تنبيه SMSC بأن مشترك متاح.

المصادقة

العملية	نقطة النهاية	الوصف
إرسال معلومات المصادقة	`POST /api/send-auth-info`	استرداد متجهات المصادقة من HLR.
إرسال IMSI	`POST /api/sendIMSI`	حل MSISDN إلى IMSI الخاص به في HLR.

بيانات المشترك

العملية	نقطة النهاية	الوصف
إرسال معلومات التوجيه	`POST /api/sri`	استعلام HLR عن معلومات توجيه المكالمات الصوتية.
توفير معلومات المشترك	`POST /api/provideSubscriberInfo`	طلب حالة/موقع المشترك من VLR.
حذف بيانات المشترك	`POST /api/deleteSubscriberData`	إزالة بيانات المشترك من VLR (علامات `withdraw` اختيارية).
استجواب في أي وقت	`POST /api/anyTimeInterrogation`	يقوم gsmSCF باستجواب معلومات/موقع المشترك في HLR.
استجواب اشتراك في أي وقت	`POST /api/anyTimeSubscriptionInterrogation`	استعلام بيانات الاشتراك في HLR.
تعديل في أي وقت	`POST /api/anyTimeModification`	تعديل بيانات الاشتراك في HLR.
ملاحظة تعديل بيانات المشترك	`POST /api/noteSubscriberDataModified`	يقوم HLR بإبلاغ gsmSCF بالبيانات المتغيرة.
ملاحظة MM-Event	`POST /api/noteMM-Event`	الإبلاغ عن حدث إدارة التنقل إلى gsmSCF.

الخدمات الإضافية

العملية	نقطة النهاية	الوصف
استجواب SS	`POST /api/interrogateSS`	استجواب حالة خدمة إضافية.
تسجيل SS	`POST /api/registerSS`	تسجيل خدمة إضافية (مثل إعادة توجيه المكالمات / CFU).
مسح SS	`POST /api/eraseSS`	مسح خدمة إضافية.
تفعيل SS	`POST /api/activateSS`	تفعيل خدمة إضافية.
إلغاء تفعيل SS	`POST /api/deactivateSS`	إلغاء تفعيل خدمة إضافية.
تسجيل كلمة مرور	`POST /api/registerPassword`	تسجيل كلمة مرور SS (`ss_code` عدد صحيح).
الحصول على كلمة المرور	`POST /api/getPassword`	استرداد/التحقق من كل��ة مرور SS.
إعادة توجيه تحقق SS إشارة	`POST /api/forwardCheckSS-Indication`	تحقق غير مؤكد نحو VLR (يرجع `{"status":"sent"}`).

تشارك العمليات الخمس الأساسية SS (`interrogateSS`، `registerSS`، `eraseSS`، `activateSS`، `deactivateSS`) شكل طلب: `ss_code` (عدد صحيح) و `hlr_gt`، مع `basic_service` و `no_reply_time` و `forwarded_to` اختيارية.

خدمات الموقع (LCS)

العملية	نقطة النهاية	الوصف
إرسال معلومات التوجيه لخدمات الموقع	`POST /api/sendRoutingInfoForLCS`	يقوم GMLC باستجواب HLR عن توجيه LCS.
توفير موقع المشترك	`POST /api/provideSubscriberLocation`	طلب موقع مشترك من العقدة الخادمة.
تقرير موقع المشترك	`POST /api/subscriberLocationReport`	الإبلاغ عن موقع مشترك إلى GMLC.

GPRS

العملية	نقطة النهاية	الوصف
تحديث موقع GPRS	`POST /api/updateGprsLocation`	تسجيل SGSN الخادم لمشترك مع HLR.
إرسال معلومات التوجيه لـ GPRS	`POST /api/sendRoutingInfoForGprs`	استعلام HLR عن توجيه GPRS (اختيار GGSN).

المعدات

العملية	نقطة النهاية	الوصف
التحقق من IMEI	`POST /api/checkIMEI`	استعلام حالة المعدات في EIR.

USSD

العملية	نقطة النهاية	الوصف
إرسال USSD (منشأ من الشبكة)	`POST /api/ussd/send`	دفع رسالة USSD إلى مشترك. يتطلب `ussd_gateway_enabled: true` (وإلا `503`). انظر دليل بوابة USSD.

CAP / CAMEL

العملية	نقطة النهاية	الوصف
CAP InitialDP	`POST /api/cap/initialDP`	بدء حوار CAMEL نحو gsmSCF. يرجع `otid` للعمليات اللاحقة.
CAP Connect	`POST /api/cap/connect`	توجيه المكالمة إلى وجهة (`destination_number` + `otid`).
CAP Continue	`POST /api/cap/continue`	متابعة معالجة المكالمة (`otid`).
CAP Release Call	`POST /api/cap/releaseCall`	إنهاء المكالمة مع `cause` (`otid`).
CAP Apply Charging	`POST /api/cap/applyCharging`	تطبيق نافذة الشحن (`duration` + `otid`).

تحمل عمليات المتابعة CAP `otid` (hex، ا��ذي تم إرجاعه بواسطة `initialDP`) وتستجيب بـ `{"status":"sent","otid":"<hex>"}`. انظر دليل بوابة CAMEL.

الحقن الخام (الاختبار / التشخيص)

العملية	نقطة النهاية	الوصف
SCCP الخام	`POST /api/raw/sccp`	حقن حمولة SCCP مبنية يدويًا (hex). تجاوزات SCCP اختيارية (الطرف المتصل/المتصل، SSNs، OPC/DPC).
M3UA الخام	`POST /api/raw/m3ua`	حقن حمولة M3UA مبنية يدويًا (hex). تجاوزات سياق التوجيه اختيارية/OPC/DPC.

نقاط نهاية المرافق

نقطة النهاية	الطريقة	الغرض
`/metrics`	GET	مقاييس بروميثيوس (تنسيق نصي).
`/swagger`	GET	واجهة Swagger.
`/swagger.json`	GET	مواصفة OpenAPI.

أمثلة عملية

توضح الأمثلة التالية شكل الطلب/الاستجابة للعمليات التمثيلية. للحصول على مرجع كامل للحقول لكل عملية، انظر دليل عميل MAP.

SendRoutingInfo (SRI)

استرداد معلومات التوجيه لإجراء مكالمة إلى مشترك هاتف محمول.

نقطة النهاية: `POST /api/sri`

جسم الطلب:

```
{
  "msisdn": "1234567890",
  "gmsc": "5551234567"
}
```

الحقل	النوع	مطلوب	الوصف
`msisdn`	سلسلة	نعم	MSISDN الطرف المتصل
`gmsc`	سلسلة	نعم	عنوان GMSC Global Title

الاستجابة (200 OK):

```
{
  "result": {
    "imsi": "001001234567890",
    "msrn": "5551234999",
    "vlr_number": "5551234800"
  }
}
```

مثال cURL:

```
curl -X POST http://localhost:9090/api/sri \
  -H "Content-Type: application/json" \
  -d '{"msisdn": "1234567890", "gmsc": "5551234567"}'
```

SendRoutingInfoForSM (SRI-for-SM)

استرداد معلومات التوجيه لتسليم رسالة قصيرة إلى مشترك هاتف محمول.

نقطة النهاية: `POST /api/sri-for-sm`

جسم الطلب:

```
{
  "msisdn": "1234567890",
  "service_center": "5551234567"
}
```

الحقل	النوع	مطلوب	الوصف
`msisdn`	سلسلة	نعم	MSISDN الوجهة
`service_center`	سلسلة	نعم	Global Title عنوان مركز الخدمة

مثال cURL:

```
curl -X POST http://localhost:9090/api/sri-for-sm \
  -H "Content-Type: application/json" \
  -d '{"msisdn": "1234567890", "service_center": "5551234567"}'
```

UpdateLocation

إبلاغ HLR بتغيير موقع مشترك (تسجيل VLR). في وضع HLR، يؤدي ذلك إلى تشغيل تسلسل InsertSubscriberData (ISD).

نقطة النهاية: `POST /api/updateLocation`

جسم الطلب:

```
{
  "imsi": "001001234567890",
  "vlr": "5551234800"
}
```

الحقل	النوع	مطلوب	الوصف
`imsi`	سلسلة	نعم	IMSI المشترك
`vlr`	سلسلة	نعم	عنوان VLR Global Title

USSD Send (منشأ من الشبكة)

دفع رسالة USSD إلى مشترك. يتطلب `ussd_gateway_enabled: true`. للحصول على بروتوكول الاستدعاء الكامل ودورة حياة الجلسة، انظر دليل بوابة USSD.

نقطة النهاية: `POST /api/ussd/send`

جسم الطلب:

```
{
  "msisdn": "+254712345678",
  "text": "لديك فاتورة معلقة. رد 1 للدفع.",
  "callback_url": "http://billing-app:9000/ussd"
}
```

الح��ل	النوع	مطلوب	الوصف
`msisdn`	سلسلة	نعم	MSISDN الوجهة المشترك
`text`	سلسلة	نعم	نص USSD لعرضه (GSM 7-bit)
`callback_url`	سلسلة	نعم	URL لاستلام ردود المشترك عبر HTTP POST

الاستجابة (200 OK):

```
{
  "session_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "status": "sent"
}
```

(503، بوابة معطلة) **خطأ**:

```
{
  "error": "USSD gateway not enabled"
}
```

المصادقة

الحالة الحالية: لا تتطلب واجهة برمجة التطبيقات **المصادقة**.

اعتبارات الأمان:

- تم تصميم واجهة برمجة التطبيقات للاستخدام في الشبكات الداخلية/الموثوقة
- ضع في اعتبارك استخدام قواعد جدار الحماية لتقييد الوصول
- بالنسبة لنشر الإنتاج، ضع في اعتبارك تنفيذ واجهة مصادقة

تنسيقات الاستجابة

تستخدم جميع الاستجابات تنسيق **JSON**.

استجابة النجاح

حالة HTTP: 200 OK

الهيكل:

```
{
  "result": {
    // بيانات استجابة محددة للعملية
  }
}
```

استجابة الخطأ

حالة HTTP:

- 400 Bad Request - حقول الطلب غير صالحة أو مفقودة
- 500 Internal Server Error - فشلت العملية (خطأ غير متعلق بالمهلة تم إرجاعه بواسطة كومة SS7)
- 503 Service Unavailable - ميزة غير مفعلة (بوابة USSD)
- 504 Gateway Timeout - تجاوزت العملية مهلة (الافتراضي 10 ثوانٍ)
- 404 Not Found - نقطة نهاية غير صالحة

الهيكل:

```
{
  "error": "timeout"
}
```

أو

```
{
  "error": "invalid request"
}
```

معالجة الأخطاء

الأخطاء الشائعة

الخطأ	رمز HTTP	الوصف	الحل
JSON غير صالح	400	جسم الطلب ليس JSON صالحًا	تحقق من بناء جملة JSON
حقول مفقودة/ غير صالحة	400	الحقول المطلوبة مفقودة أو خاطئة النوع (غالبًا ما تتضمن الاستجابات قائمة `"required"`)	تضمين جميع المعلمات المطلوبة مع الأنواع الصحيحة
خطأ في العملية	500	خطأ غير SS7 أرجعت كومة متعلق بالمهلة	افحص رسالة `"error"` تحقق؛ من سجلات أحداث SS7
الميزة معطلة	503	لم يتم تفعيل بوابة USSD	تعيين `ussd_gateway_enabled: true`
انتهاء المهلة	504	تجاوزت العملية مهلة (الافتراضي 10 ثوانٍ)	تحقق من اتصال M3UA، توفر HLR/VLR
غير موجود	404	نقطة نهاية غير صالحة	تحقق من عنوان URL لنقطة النهاية

سلوك انتهاء المهلة

تستخدم العمليات المعلقة **مهلة 10 ثوانٍ** بشكل افتراضي:

1. يتم إرسال الطلب إلى MapClient GenServer
2. ينتظر الرد لمدة تصل إلى 10 ثوانٍ
3. إذا لم يكن هناك رد ← يرجع 504 Gateway Timeout
4. إذا تم استلام الرد ← يرجع 200 OK مع النتيجة

استكشاف أخطاء انتهاء المهلة:

- تحقق من حالة اتصال M3UA (صفحة واجهة الويب M3UA)
- تحقق من أن العنصر الشبكي (HLR/VLR/MSC) يمكن الوصول إليه
- تحقق من تكوين التوجيه
- راجع سجلات SS7 بحثًا عن الأخطاء

المقاييس (بروميثيوس)

تقوم واجهة برمجة التطبيقات بتعريض مقاييس بروميثيوس للمراقبة.

نقطة نهاية المقاييس

الرابط: `http://[server-ip]:9090/metrics`

التنسيق: تنسيق نص بروميثيوس

مثال على المخرجات:

```
# HELP map_requests_total إجمالي طلبات MAP
# TYPE map_requests_total counter
map_requests_total{operation="sri"} 42
map_requests_total{operation="sri_for_sm"} 158
map_requests_total{operation="updateLocation"} 23

# HELP cap_requests_total إجمالي طلبات CAP
# TYPE cap_requests_total counter
cap_requests_total{operation="initialDP"} 87
cap_requests_total{operation="requestReportBCSMEvent"} 91

# HELP map_request_duration_milliseconds مدة طلبات/استجابات MAP
بالمللي ثانية
# TYPE map_request_duration_milliseconds histogram
map_request_duration_milliseconds_bucket{operation="sri",le="10"}
5
map_request_duration_milliseconds_bucket{operation="sri",le="50"}
12
map_request_duration_milliseconds_bucket{operation="sri",le="100"}
35
...

# HELP map_pending_requests عدد طلبات MAP TID المعلقة
# TYPE map_pending_requests gauge
map_pending_requests 3

# HELP omniss7_license_status الحالة الحالية للرخصة (1 = صالحة، 0
= غير صالحة)
# TYPE omniss7_license_status gauge
omniss7_license_status 1
```

المقاييس المتاحة

تُجمع المقاييس حسب طبقة البروتوكول / النظام الفرعي. يتم الإعلان عن جميعها عند بدء التشغيل وإصدارها بمساعدين محميين، لذا فإن مقياس غير مسجل لا يعطل مسار البروتوكول الساخن.

التطبيق (MAP / CAP / USSD / الترخيص)

المقياس	النوع	التسميات	الوصف
`map_requests_total`	عداد	`operation`	إجمالي عدد طلبات MAP حسب نوع العملية
`map_request_errors_total`	عداد	`operation`	إجمالي أخطاء طلبات MAP حسب نوع العملية
`cap_requests_total`	عداد	`operation`	إجمالي عدد طلبات CAP حسب نوع العملية
`cap_request_errors_total`	عداد	`operation`	إجمالي أخطاء طلبات CAP حسب نوع العملية
`map_request_duration_milliseconds`	هيستوجرام	`operation`	مدة الطلب بالمللي ثانية
`map_pending_requests`	مقياس	-	عدد معاملات MAP المعلقة

المقياس	النوع	التسميات	الوصف
`ussd_requests_total`	عداد	`direction`	إجمالي طلبات USSD (داخلي/خارجي)
`ussd_active_sessions`	مقياس	-	عدد ��لسات USSD النشطة
`omniss7_license_status`	مقياس	-	الحالة الحالية للرخصة (1 = صالحة، 0 = غير صالحة)

SCCP

المقياس	النوع	التسميات	
`sccp_messages_total`	عداد	`message_type`، `direction`	وع SCCP رسائل (`udt`/`xudt`/`lu` والاتجاه (`sent`/
`sccp_decode_errors_total`	عداد	-	ير SCCP رسائل
`sccp_segments_received_total`	عداد	-	XUDT/L مقاطع
`sccp_reassembly_completed_total`	عداد	-	ل SCCP رسائل
`sccp_reassembly_timeouts_total`	عداد	-	ع بواسطة مؤقت التنظيف

TCAP

المقياس	النوع	التسميات	
`tcap_dialogues_total`	عداد	`type`	سب نوع الرسالة TCAP حوارات
`tcap_aborts_total`	عداد	`reason`	حسب السبب TCAP رسائل إلغاء `unrecognized_message_ty` (أو `p_abort_other`); U-Abo
`tcap_decode_errors_total`	عداد	-	وف/غير قابل للفك TCAP رسائل
`tcap_active_dialogues`	مقياس	-	المفتوحة حاليًا MAP معاملات

SCTP

المقياس	النوع	التسميات	
`sctp_association_state`	مقياس	`remote_ip`، `remote_port`	غير نشط)
`sctp_association_state_changes_total`	عداد	`event`	ت الارتباط (`comm_up`
`sctp_send_errors_total`	عداد	`reason`	بات إرسال M2PA)
`sctp_heartbeat_sent_total`	عداد	`type`	M3 نبضات
`sctp_heartbeat_received_total`	عداد	`type`	M3 نبضات

الرسائل القصيرة / SMSc

المقياس	النوع	التسميات	الوصف
`sms_messages_total`	عداد	`direction`، `operation`	الرسائل القصيرة حسب الاتجاه (`mo`/`mt`) وعملية (`submit`/`deliver`)
`sms_delivery_failures_total`	عداد	`reason`	عمليات تسليم الرسائل القصيرة الفاشلة حسب السبب
`sms_tracked_subscribers`	مقياس	-	المشتركين الذين يتم تتبعهم حاليًا بواسطة SubscriberTracker

M2PA (تكييف MTP2 Peer-to-Peer)

المقياس	النوع	التسميات	الوصف
`m2pa_messages_sent_total`	عداد	`link_name`، `point_code`، `adjacent_pc`	رسائل MTP3 المرسلة عبر M2PA
`m2pa_messages_received_total`	عداد	`link_name`، `point_code`، `adjacent_pc`	رسائل MTP3 المستلمة عبر M2PA
`m2pa_bytes_sent_total`	عداد	`link_name`، `point_code`، `adjacent_pc`	البايتات المرسلة عبر M2PA
`m2pa_bytes_received_total`	عداد	`link_name`، `point_code`، `adjacent_pc`	البايتات المستلمة عبر M2PA
`m2pa_link_state_changes_total`	عداد	`link_name`، `from_state`، `to_state`	انتقالات حالة ارتباط M2PA
`m2pa_errors_total`	عداد	`link_name`، `error_type`	أخطاء M2PA حسب النوع

M3UA STP

المقياس	النوع	التسميات	لوصف
`m3ua_stp_messages_received_total`	عداد	`peer_name`، `point_code`	من M3UA رسائل نظير
`m3ua_stp_messages_sent_total`	عداد	`peer_name`، `point_code`	من M3UA رسائل نظير
`m3ua_stp_routing_failures_total`	عداد	`reason`	فشل توجيه STP (`no_route`/`no_`

تكوين بروميثيوس

أضف إلى `prometheus.yml` الخاص بك:

```
scrape_configs:
  - job_name: 'omniss7'
    static_configs:
      - targets: ['server-ip:9090']
    metrics_path: '/metrics'
    scrape_interval: 15s
```

طلبات أمثلة

مثال بايثون

```python
import requests
import json

# طلب SRI-for-SM
url = "http://localhost:9090/api/sri-for-sm"
payload = {
    "msisdn": "1234567890",
    "service_center": "5551234567"
}

response = requests.post(url, json=payload, timeout=15)

if response.status_code == 200:
    result = response.json()
    print(f"نجاح: {result}")
elif response.status_code == 504:
    print("انتهاء المهلة - لا استجابة من الشبكة")
else:
    print(f"خطأ: {response.status_code} - {response.text}")
```

مثال جافا سكريبت

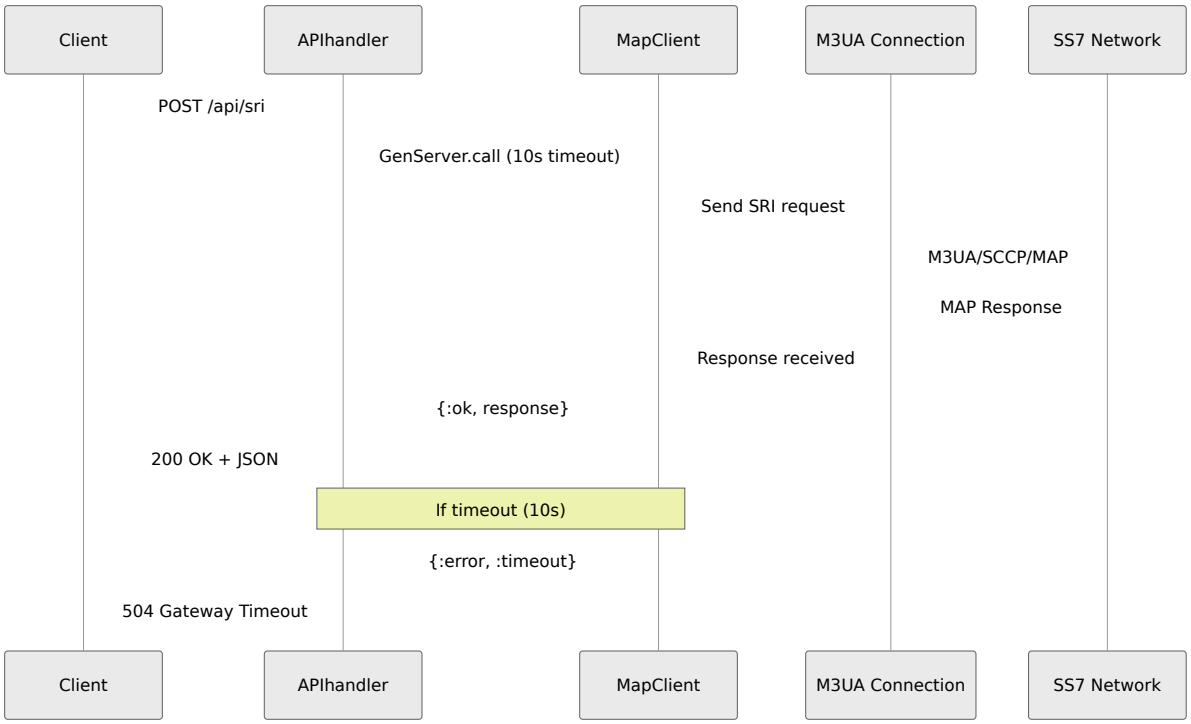
```
const axios = require('axios');

async function sendSRI() {
  try {
    const response = await
axios.post('http://localhost:9090/api/sri', {
      msisdn: '1234567890',
      gmsc: '5551234567'
    }, {
      timeout: 15000
    });

    console.log('نجاح:', response.data);
  } catch (error) {
    if (error.code === 'ECONNABORTED') {
      console.error('انتهاء المهلة - لا استجابة من الشبكة');
    } else {
      console.error('خطأ:', error.response?.data ||
error.message);
    }
  }
}

sendSRI();
```

مثال Bash/cURL

```
#!/bin/bash

# طلب UpdateLocation
response=$(curl -s -w "\n%{http_code}" -X POST
http://localhost:9090/api/updateLocation \
  -H "Content-Type: application/json" \
  -d '{
    "imsi": "001001234567890",
    "vlr": "5551234800"
  }')

http_code=$(echo "$response" | tail -n 1)
body=$(echo "$response" | sed '$d')

if [ "$http_code" -eq 200 ]; then
  echo "نجاح: $body"
elif [ "$http_code" -eq 504 ]; then
  echo "انتهاء المهلة - لا استجابة من الشبكة"
else
  echo "خطأ $http_code: $body"
fi
```

مخططات التدفق

تدفق طلب واجهة برمجة التطبيقات



الملخص

توفر واجهة برمجة التطبيقات REST API لـ OmniSS7:

- **47 نقطة نهاية عملية** عبر التنقل، الرسائل القصيرة، المصادقة، بيانات المشترك، والحقن CAP/CAMEL، USSD، المعدات، GPRS، الخدمات الإضافية، خدمات الموقع الخام.
- **واجهة Swagger UI** في `/swagger` لوثائق تفاعلية واختبار.
- في **مقاييس بروميثيوس** `/metrics` للمراقبة والرؤية.
- **مهلة افتراضية مدتها 10 ثوانٍ** للعمليات المعلقة (`504` عند انتهاء المهلة).
- خادم **HTTP** `Plug.Cowboy` يتم، **على المنفذ 9090** بواسطة toggled `start_http_server`. ووحدات (:8087) هذا الخادم منفصل عن لوحة التحكم REST `:api_ex` (:8445).

انظر دليل عميل MAP، للحصول على مخططات الطلب/الاستجابة لكل عملية وأكواد MAP.

للوصول إلى واجهة الويب، انظر دليل واجهة الويب.

للحصول على تفاصيل التكوين، انظر مرجع التكوين.

المرجع الفني (الملحق)

← العودة إلى الوثائق الرئيسية

المرجع الفني لبروتوكولات SS7 وتنفيذ OmniSS7.

كومة بروتوكولات SS7



رموز عمليات MAP

تغطي هذه 3GPP TS 29.002 وفقًا لـ MAP الخاصة بـ `local` الرموز أدناه هي رموز العمليات الجدول العمليات التي تولدها OmniSS7 من مجموعة كاملة للحصول على و/أو تتعامل معها؛ العمليات القابلة للتفعيل من قبل العميل ومعلمات الطلب الخاصة بها، انظر **دليل عميل MAP**.

العملية	الرمز	الغرض
updateLocation	2	تسجيل موقع المشترك (CS)
cancelLocation	3	إلغاء تسجيل المشترك من VLR/SGSN
provideRoamingNumber	4	طلب MSRN
noteSubscriberDataModified	5	إخطار gsmSCF بتغيير بيانات المشترك
insertSubscriberData	7	دفع ملف تعريف المشترك إلى VLR/SGSN
deleteSubscriberData	8	سحب بيانات المشترك من VLR/SGSN
registerSS	10	تسجيل خدمة إضافية (مثل CFU)
eraseSS	11	مسح خدمة إضافية
activateSS	12	تفعيل خدمة إضافية
deactivateSS	13	إلغاء تفعيل خدمة إضافية
interrogateSS	14	استجواب حالة الخدمة الإضافية
registerPassword	17	تسجيل كلمة مرور SS
getPassword	18	طلب كلمة مرور SS من المشترك
sendRoutingInfo	22	استعلام عن توجيه المكالمات (SRI)
updateGprsLocation	23	تسجيل الموقع المعتمد على الحزم (SGSN)

العملية	الرمز	الغرض
sendRoutingInfoForGprs	24	استعلام عن توجيه GPRS نحو GGSN
reset	37	إعادة تعيين HLR نحو VLR/SGSN
forwardCheckSS-Indication	38	الإشارة إلى أن بيانات SS يجب إعادة التحقق منها
checkIMEI	43	استعلام EIR عن حالة المعدات
mt-forwardSM	44	تسليم SMS إلى المشترك (MT)
sendRoutingInfoForSM	45	استعلام عن توجيه SMS (SRI-for-SM)
mo-forwardSM	46	إعادة توجيه SMS من المشترك (MO)
reportSM-DeliveryStatus	47	الإبلاغ عن نتيجة تسليم SMS إلى HLR
sendIdentification	55	استرجاع IMSI/auth من VLR السابق
sendAuthenticationInfo	56	طلب متجهات المصادقة
restoreData	57	استعادة بيانات المشترك بعد فشل VLR
sendIMSI	58	حل MSISDN إلى IMSI في HLR
processUnstructuredSS-Request	59	طلب USSD منشأ من الهاتف المحمول (داخلي)
unstructuredSS-Request	60	طلب USSD منشأ من الشبكة (خارجي)
unstructuredSS-Notify	61	إشعار USSD منشأ من الشبكة

العملية	الرمز	الغرض
anyTimeSubscriptionInterrogation	62	استعلام gsmSCF عن بيانات الاشتراك
alertServiceCentre	64	تنبيه SMSc بأن المشترك متاح
anyTimeModification	65	تعديل gsmSCF لبيانات المشترك
readyForSM	66	إخطار HLR لـ بأن المشترك جاهز SMS
purgeMS	67	تطهير سجل المشترك من VLR/SGSN
provideSubscriberInfo	70	طلب حالة/موقع المشترك
anyTimeInterrogation	71	استعلام gsmSCF عن حالة/موقع المشترك
provideSubscriberLocation	83	طلب تقدير الموقع من العقدة الخدمية (LCS)
sendRoutingInfoForLCS	85	استعلام GMLC عن العقدة الخدمية للمشترك (LCS)
subscriberLocationReport	86	بموقع GMLC العقدة الخدمية تبلغ (LCS)
noteMM-Event	89	الإبلاغ عن حدث إدارة الحركة إلى gsmSCF

أنواع رسائل TCAP

- **BEGIN** - بدء المعاملة
- **CONTINUE** - أثناء المعاملة

- **END** - الاستجابة النهائية
- **ABORT** - إلغاء المعاملة

عنونة SCCP

تنسيقات العنوان العالمية

- **E.164** - رقم الهاتف الدولي (مثل، 447712345678)
- **E.212** - تنسيق IMSI (مثل، 234509876543210)
- **E.214** - تنسيق رمز النقطة

أرقام الأنظمة الفرعية (SSN)

- **SSN 6**: HLR
- **SSN 7**: VLR
- **SSN 8**: MSC/SMSC
- **SSN 9**: GMLC
- **SSN 10**: SGSN

SMS TPDU

أنواع الرسائل

- **SMS-DELIVER** (MT) - من الشبكة إلى الهاتف المحمول
- **SMS-SUBMIT** (MO) - من الهاتف المحمول إلى الشبكة
- **SMS-STATUS-REPORT** - حالة التسليم
- **SMS-COMMAND** - أمر عن بعد

ترميزات الأحرف

- **GSM7** - أبجدية GSM ذات 7 بت (160 حرف لكل SMS)

- **UCS2** - Unicode ذو 16 بت (70 SMS لكل حرف)
- **8-bit** - بيانات ثنائية (140 SMS بايت لكل)

حالات M3UA

- **DOWN** - لا يوجد اتصال SCTP
- **CONNECTING** - الاتصال بـ SCTP
- **ASPUP_SENT** - في انتظار تأكيد ASPUP
- **INACTIVE** - نشط ولكن غير مفعل ASP
- **ASPAC_SENT** - في انتظار تأكيد ASPAC
- **ACTIVE** - جاهز للحركة

رموز النقاط الشائعة في SS7

عادةً ما تكون رموز النقاط قيمًا ذات 14 بت (ITU) أو 24 بت (ANSI).

تنسيق المثال (ITU):

- الشبكة: 3 بت
- العنقود: 8 بت
- العضو: 3 بت

رموز أخطاء SCCP

- **0** - لا يوجد ترجمة للعنوان
- **1** - لا يوجد ترجمة لعنوان محدد
- **2** - ازدحام النظام الفرعي
- **3** - فشل النظام الفرعي
- **4** - مستخدم غير مجهز
- **5** - فشل MTP

- **6** - ازدحام الشبكة
- **7** - غير مؤهل
- **8** - خطأ في نقل الرسالة

رموز أخطاء MAP

الرمز	الخطأ	الوصف
1	unknownSubscriber	المشترك غير موجود في HLR
27	absentSubscriber	المشترك غير متاح
34	systemFailure	فشل الشبكة
35	dataMissing	البيانات المطلوبة غير متاحة
36	unexpectedDataValue	قيمة معلمة غير صالحة

الوثائق ذات الصلة

- ← العودة إلى الوثائق الرئيسية
- دليل STP
- دليل عميل MAP
- دليل مركز SMS
- دليل HLR
- الميزات الشائعة

دليل تكوين بوابة CAMEL

نظرة عامة

توفر التحكم (IN) إلى منصة شبكة ذكية OmniSS7 يحول **CAMEL (CAMELGW)** وضع **بوابة** في المكالمات والشحن في الوقت الحقيقي باستخدام بروتوكول جزء تطبيق CAMEL (CAP). حيث يتلقى المحفزات من ،(وظيفة التحكم في الخدمة) **gsmSCF** كـ OmniSS7 يعمل **gsmSSF** في MSC/VLR/GMSC ويقوم بتحريك قرارات الشحن من خلال CGrateS.

ما هو CAMEL؟

CAMEL (تطبيقات مخصصة لشبكة الهاتف المحمول المنطق المعزز) هو مجموعة من المعايير التي تعمل على شبكة GSM الأساسية أو شبكة UMTS. يسمح للمشغلين بتوفير خدمات تتطلب التحكم في المكالمات في الوقت الحقيقي، مثل:

- **المكالمات المدفوعة مسبقًا** - التحقق من الرصيد ��الشحن في الوقت الحقيقي
- **خدمات الأسعار المميزة** - الفوترة الخاصة للخدمات ذات القيمة المضافة
- **تحكم في توجيه المكالمات** - توجيه الوجهة الديناميكي بناءً على الوقت/الموقع

- **الشبكات الخاصة الافتراضية** - خطط ترقيم الشركات
- **تصفية المكالمات** - السماح/حظر المكالمات بناءً على معايير

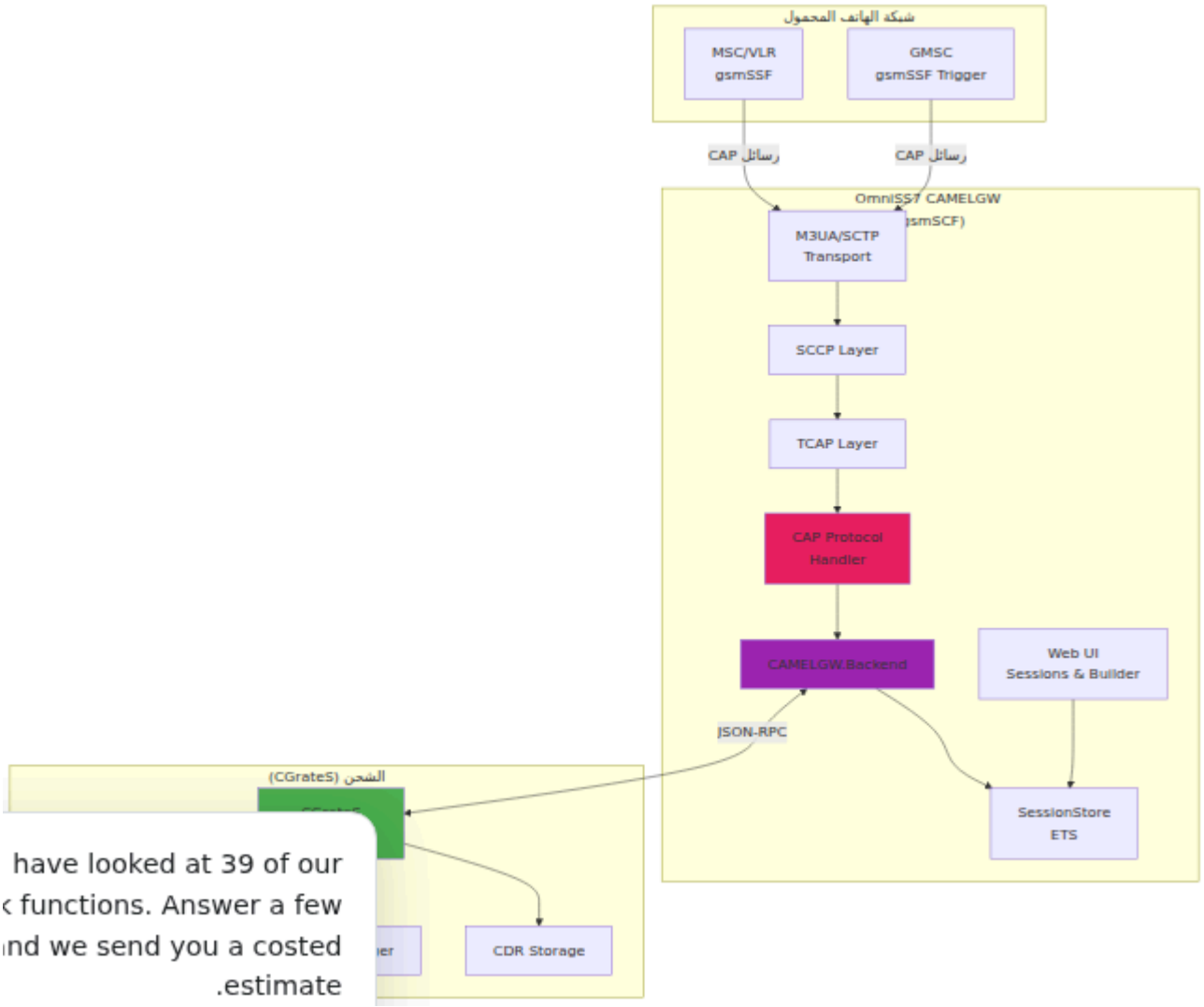
إصدارات بروتوكول CAP

يدعم OmniSS7 CAMELGW إصدارات متعددة من CAP. تتحكم النسخة في OID سياق التطبيق الذي يتم حمله في حوار TCAP:

الإصدار	المرحلة	سياق التطبيق OID
CAP v1	المرحلة 1 من CAMEL	`0.4.0.0.1.0.50.0`
CAP v2	المرحلة 2 من CAMEL	`0.4.0.0.1.0.50.1`
CAP v3	المرحلة 3 من CAMEL	`0.4.0.0.1.21.3.4`
CAP v4	المرحلة 4 من CAMEL	`0.4.0.0.1.23.3.4`

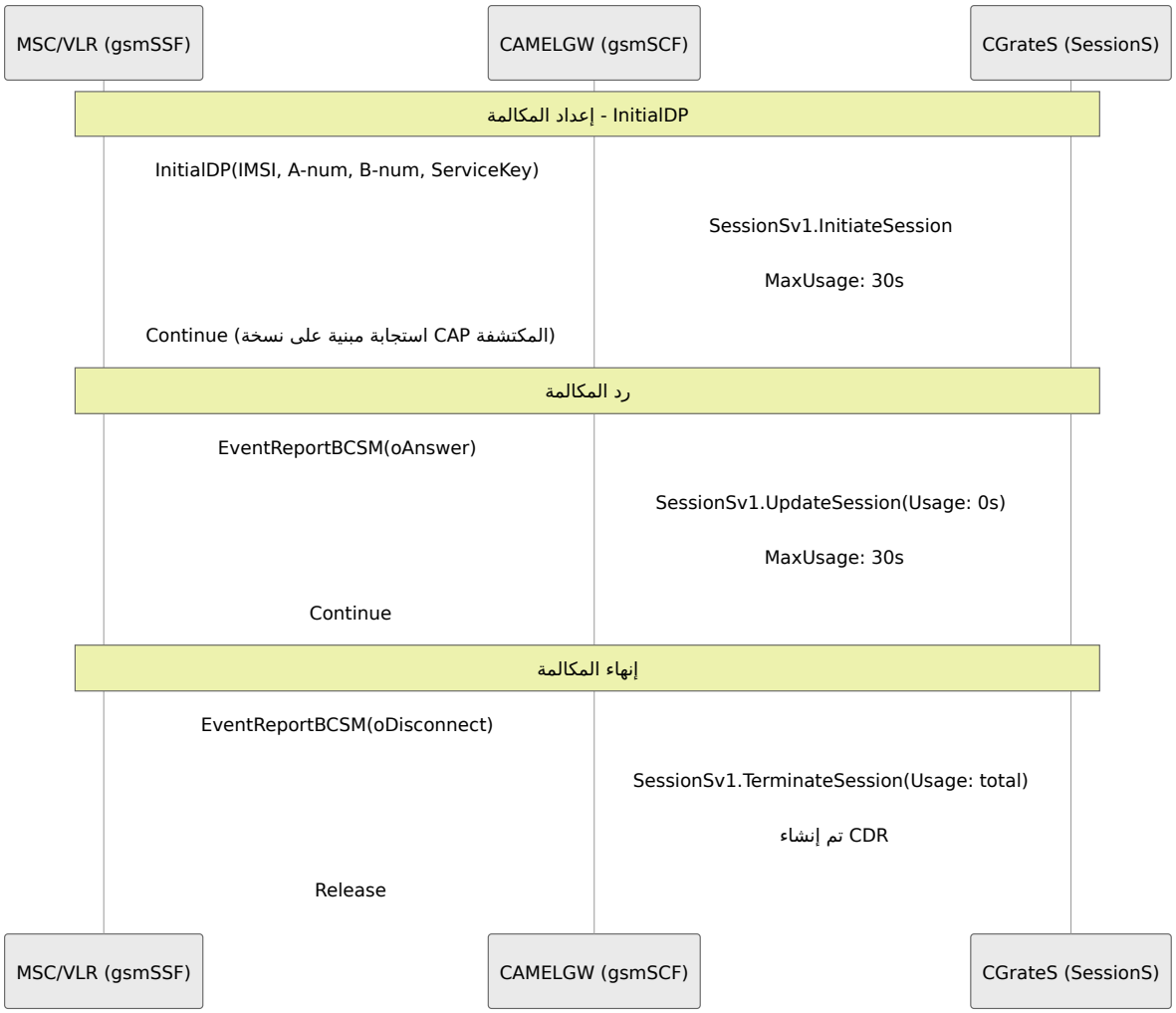
الإعداد الافتراضي: CAP v2 (الأكثر انتشارًا). يتم اكتشاف النسخة الواردة تلقائيًا من جزء الحوار InitialDP وإعادة استخدامها عند توليد الاستجابة، بحيث تتطابق الاستجابات مع نسخة CAP لكل MSC.

الهيكلية



have looked at 39 of our
k functions. Answer a few
nd we send you a costed
.estimate

مثال على تدفق المكالمات



التكوين

المتطلبات الأساسية

- تثبيت وتشغيل OmniSS7
- اتصال M3UA بـ MSC/GMSC (gsmSSF)
- اختياري، للشحن في الوقت الحقيقي) يمكن الوصول إليه عبر JSON-RPC مثيل CGrateS

تفعيل وضع بوابة CAMEL

يتم التحكم في معالجة CAMEL بواسطة علامتين مستقلتين. `cap_client_enabled` يشغل عميل CAP/M3UA ومكونات جلسة CAMEL؛ `cgrates_enabled` يقوم بتشغيل تكامل شحن CGrateS خلف الواجهة الخلفية. قم بتكوين هذه في `config/runtime.exs`:

```
config :omniss7,
  # تشغيل عميل CAP/CAMEL (M3UA ASP، SessionStore، معالجة الطلبات)
  cap_client_enabled: true,

  # أوضاع مستقلة أخرى (تعيين حسب الحاجة لنشر الخاص بك)
  map_client_enabled: false,
  smsc_mode_enabled: false,

  # نسخة CAP الحوار / المستخدمة للطلبات الصادرة (:v1 | :v2 | :v3 |
:v4)
  cap_version: :v2,

  # تكامل شحن CGrateS
  cgrates_enabled: true,

  # اتصال M3UA لـ CAP (الاتصال كـ ASP نحو MSC/GMSC gsmSSF)
  cap_client_m3ua: %{
    routing_context: 1,
    opc: 5013,
    dpc: 5011
  }
```

> **ملاحظة حول `cap_client_m3ua`:** يقرأ عميل CAMEL فقط `:routing_context` و `:opc` و `:dpc` من هذه الخريطة. يتم تجاهل أي مفاتيح إضافية بواسطة عميل CAP. اسم عملية M3UA ASP ثابت داخليًا كـ `:camelgw_client_asp`.

> لا يوجد علم `camelgw_mode_enabled` أو `hlr_mode_enabled` أو `smsc_mode_enabled`-لـ-CAMEL. العلامات الوحيدة التي تتحكم في سلوك CAMEL هي `cap_client_enabled` و `cgrates_enabled`.

أعلام الوضع

المفتاح	الافتراضي	الغرض
`cap_client_enabled`	`false`	يبدأ عميل CAP/CAMEL (M3UA ASP، SessionStore، معالجة طلبات CAP)
`cgrates_enabled`	`false`	يمكّن تكامل شحن CGrateS في `CAMELGW.Backend`
`cap_version`	`:v2`	نسخة CAP الافتراضية / OID الحوار للطلبات الصادرة
`map_client_enabled`	`false`	وضع عميل MAP المستقل
`smsc_mode_enabled`	`false`	وضع SMSc المستقل (يتطلب عميل MAP)

معلمات شحن CGrateS

المفتاح	الافتراضي	
`cgrates_enabled`	`false`	ي لتكامل CGrate
`cgrates_url`	`http://localhost:2080/jsonrpc`	طة نهاية CGrate
`cgrates_tenant`	`cgrates.org`	تخدم في كل حدث
`cgrates_request_type`	`*prepaid`	وع طلب `*prepa` `*postp`
`cgrates_timeout`	`5000`	هلة طلب ثانية
`cgrates_voice_usage_seconds`	*مطلوب*	ة الصوت واني: كل ApplyC دار ويمنح ApplyC س القيمة مؤقتة لـ

تكوين صفحات واجهة الويب

تقوم واجهة الويب بعرض صفحات CAMEL في تكوينها. يتم تكوينها من خلال لوحة التحكم:
`config/config.exs`:

```
config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "أحداث SS7"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "الأقران"},
    {SS7.Web.CAMELSessionsLive, "/camel_sessions", "جلسات CAMEL"},
    {SS7.Web.CAMELRequestLive, "/camel_request", "منشئ طلب CAMEL"}
  ],
  page_order: [
    "/events", "/m3ua", "/camel_sessions", "/camel_request",
    "/application", "/configuration"
  ]
```

تسمية الصفحة	المسار	الوحدة
الأقران	`/m3ua`	`SS7.Web.M3UAStatusLive`
جلسات CAMEL	`/camel_sessions`	`SS7.Web.CAMELSessionsLive`
منشئ طلب CAMEL	`/camel_request`	`SS7.Web.CAMELRequestLive`

عمليات CAP المدعومة

تُحدد عمليات CAP بالاسم والاتجاه (خريطة CAP غير متسقة داخليًا). لاحظ أن بعض أرقام أوبكود أوبكود سجل الأحداث ومولد الطلبات تختلف في بعض العمليات)، لذلك يتم وصف العمليات أدناه بواسطة **الاسم ونقطة النهاية**. حيث يتم الاستشهاد بأوبكود، فهو الأوبكود ا��ذي يصدره مولد الطلبات فعليًا.

العمليات الواردة (gsmSSF → gsmSCF)

يتم إرسالها بناءً على أوبكود مكون TCAP المفكك:

العملية	الملاحظات
InitialDP	إشعار إعداد المكالمة؛ يحفز CGrateS `InitiateSession`. مقبول عند الأوبكود 0، وأيضًا InitialDP/InitialDPSMS) يُعتبر كـ) عند الأوبكود 60
AssistRequestInstructions	معترف بها؛ يتم إرسالها عند الأوبكود 1 في بداية TCAP
EventReportBCSM	أحداث BCSM. يتم تسجيل oAnswer (بدون رد)؛ oDisconnect / oAbandon / tDisconnect / tAbandon / عدم الرد تنهي جلسة / مشغول وتنهي الحوار CGrateS
ApplyChargingReport	أوبكود 36. المكالمة نشطة: تقارير الفترة إلى CGrateS وتمنح الشريحة التالية. المكالمة انتهت: تنهي جلسة CGrateS مع الاستخدام الكلي
Continue	معترف بها
ConnectToResource	معترف بها
RequestReportBCSMEvent	معترف بها
ApplyCharging	تم فك تشفير المعلمات BER ومعترف بها
Connect	تم فك تشفير المعلمات BER ومعتر�ف بها
ReleaseCall	معترف بها

تُجاب العمليات غير المعالجة بخطأ TCAP (المرفق غير مدعوم / قيمة بيانات غير متوقعة).

العمليات الصادرة (gsmSCF → gsmSSF)

الأوبكود أدناه هي القيم التي يصدرها OmniSS7. ترسل نحو gsmSSF:

العملية	الأوبكود
InitialDP	0
Connect	20
ReleaseCall	22
RequestReportBCSMEvent	23
Continue	31
ApplyCharging	35
InitialDPSMS	60
NoteMM-Event	89 (MAP)

> **InitialDPSMS** (الأوبكود 60) هو للتحكم في CAMEL لـ MO-SMS: يُرسل بواسطة gsmSCF إلى gsmSSF عندما يتم تسليح SMS-CSI / mo-sms-CSI، قبل أن يتم تمرير الرسالة القصيرة إلى SMSC. يتم ترميز المعامل باستخدام وحدة CAP ASN.1 المولدة ويحمل الحوار سياق التطبيق `id-ac-cap3-sms-AC`.
>
> **NoteMM-Event** هو عملية MAP (الأوبكود 89، ليست CAP) تُستخدم لإشعارات تنقل VLR إلى gsmSCF. يتم ترميزها باستخدام وحدة MAP ASN.1 من CAMEL M-CSI (`:TCAPMessages.encode(:MapSpecificPDUs, …)`) وتحمل سياق التطبيق `mm-EventReporting-v3`.

تُشفّر جميع رسائل CAP TCAP باستخدام `:GenericTCAP.encode(:GenericPDUs, …)`. فقط NoteMM-Event المعتمد على MAP يستخدم `:TCAPMessages`.

ميزات واجهة الويب

صفحة جلسات CAMEL

جلسات CAMEL · **العنوان:** **المسار:** `/camel_sessions` (HTTPS، المنفذ 8087)

مراقبة في الوقت الحقيقي لجلسات المكالمات النشطة CAMEL:

- **قائمة الجلسات الحية** مع تحديث تلقائي
- **تفاصيل الجلسة** - OTID، معرف المكالمة، الحالة، المدة
- **نسخة CAP** المكتشفة من InitialDP والمخزنة لكل جلسة
- **معلومات المكالمة** - IMSI، الرقم A، الرقم B، مفتاح الخدمة
- **تتبع الحالة** - `initiated`، `answered`، `terminated`

أعمدة الجدول: معرف المكالمة، الحالة، النسخة، IMSI، مفتاح الخدمة، الرقم المدعو، رقم المتصل، OTID، الخدمة، المدة، وقت البدء

تتوفر نفس بيانات الجلسة عبر واجهة برمجة التطبيقات REST عند `/camel-sessions` (انظر المراقبة والعمليات).

منشئ طلب CAMEL

منشئ ��لب · **العنوان:** CAMEL · **المسار:** `/camel_request` (HTTPS، المنفذ 8087)

أداة تفاعلية لبناء وإرسال طلبات CAP للاختبار.

أنواع الطلبات المقدمة بواسطة النموذج: InitialDP، Connect، ReleaseCall، RequestReportBCSMEvent، ApplyCharging، Continue.

محدد نسخة CAP: يحتوي النموذج على قائمة منسدلة لنسخة CAP لكل طلب (v1/v2/v3/v4 مع OIDs الخاصة بهم). يتجاوز الاختيار الافتراضي المكون مسبقًا لهذا الطلب ويستمر عبر تغييرات النموذج؛ تظهر تاريخ الطلب النسخة المستخدمة في عمود **CAP Ver**.

خيارات SCCP/M3UA المتقدمة:

الحقل	الافتراضي	الملاحظات
GT للجهة المدعوة	`99988415011`	العنوان العالمي للوجهة
GT للجهة المتصلة	(قابل للتكوين)	العنوان العالمي للمنشئ
SSN للجهة المدعوة	`146`	gsmSSF
SSN للجهة المتصلة	`146`	gsmSSF
OPC	`5013`	رمز النقطة الأصلية
DPC	`5011`	رمز النقطة الوجهة

التكامل مع CGrateS

تقوم OmniSS7 بشحن المكالمات من خلال واجهة برمجة التطبيقات JSON-RPC لـ CGrateS المكون ويقوم `cgrates_url` بالإرسال إلى `CGrateS.SessionsClient`. يقوم **SessionS** بربط عمليات CAP بأساليب SessionS:

محفز CAP	أسلوب SessionS
InitialDP	`SessionSv1.InitiateSession` (احجز شريحة واحدة)
ApplyChargingReport، المكالمة نشطة	`SessionSv1.UpdateSession` (`LastUsed` = الفترة، احجز الشريحة التالية)
ApplyChargingReport، المكالمة انتهت؛ EventReportBCSM / قطع MSC أو إلغاء من TCAP تخلي؛ إنهاء	`SessionSv1.TerminateSession` (الاستخدام الكلي)، ثم `SessionSv1.ProcessCDR`

يتبع شحن الصوت نموذج TAS Ro: نفس المستأجر (`cgrates_tenant`)، `ToR *voice`، `RequestType` `cgrates_request_type`، `Category call`، شريحة ثابتة

(`cgrates_voice_usage_seconds`)، و `GetAttributes` على كل طلب بحيث تتطابق ملفات تعريف AttributeS الخاصة بالمستأجر مع حساب الفوترة. `Account` هو **IMSI** المشترك الذي تم خدمته من InitialDP (يتم الإفراج عن InitialDP بدون IMSI). `Destination` هو الرقم المدعو. `OriginID` هو `CAMEL-<hex OTID>`، `OriginHost` هو `CAMELGW`.

دورة حياة المكالمة مع الشحن

1. بدء المكالمة (InitialDP)

عندما يرسل MSC InitialDP، يقوم CAMELGW:

1. **اكتشاف نسخة CAP** من جزء الحوار
2. **فك تشفير InitialDP** (IMSI، رقم المتصل/المدعو ISUP، CalledPartyBCDNumber، مفتاح الخدمة، نوع الحدث)
3. **استدعاء CGrateS** `SessionSv1.InitiateSession` لشريحة واحدة
4. **الرد على MSC** في TCAP Continue مع RequestReportBCSMEvent (O-BCSM أو T-BCSM جواب، قطع الساق 1 و2، التخلي؛ جميعها notifyAndContinue)، وContinue ApplyCharging

`MaxUsage` يحدد ApplyCharging `maxCallPeriodDuration`. تُمنح شريحة كاملة بدون `releaseIfDurationExceeded`؛ تُمنح باقي أقصر معها، بحيث يفرج MSC عن المكالمة عندما يتم استخدام الائتمان الأخير. لا ائتمان، أو خطأ CGrateS، يتم الرد عليه بـ **ReleaseCall** (السبب 31) في TCAP End.

2. رد المكالمة (EventReportBCSM - oAnswer / tAnswer)

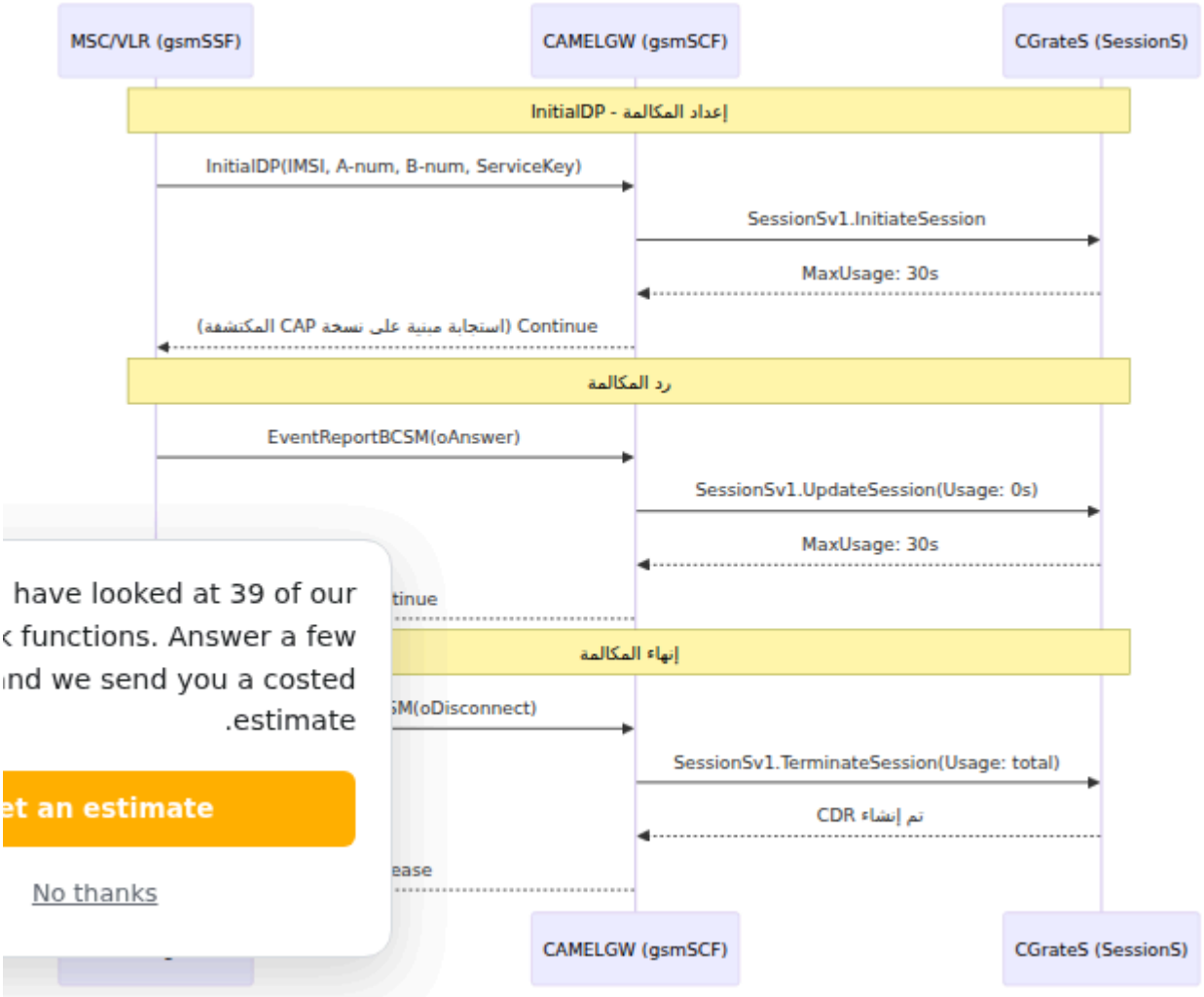
تُعَلَّم الجلسة بأنها تمت الإجابة. لا يتم إرسال شيء مرة أخرى؛ يبقى الحوار مفتوحًا.

3. انتهاء الفترة (ApplyChargingReport)

في كل مرة تنتهي الفترة الممنوحة، يقوم MSC بالإبلاغ عن الوقت المستخدم. يقوم البوابة بالإبلاغ عنه إلى CGrateS ومنح الشريحة التالية، أو إرسال ReleaseCall عندما يتم استنفاد الائتمان.

4. إنهاء المكالمة

تقوم ApplyChargingReport النهائية (المكالمة لم تعد نشطة)، أو قطع / تخلي MSC أو إغلاق، للحوار بإنهاء جلسة CGrateS مع الاستخدام المجمع، كتابة EventReportBCSM، وحذف الجلسة CDR (`ProcessCDR`).



حقول CDR من CAMEL

تنتج CDRs بواسطة CGrateS وعادة ما تشمل:

- `Account` - IMSI (أو رقم ال��تصل كخيار احتياطي)
- `Destination` - رقم الطرف المدعو
- `OriginID` - معرف المكالمة (`CAMEL-<hex OTID>`)
- `Usage` - إجمالي مدة المكالمة (بالثواني)
- `IMSI`، `CallingPartyNumber`، `CalledPartyNumber`
- `MSCAddress` - MSC الخادم
- `ServiceKey` - مفتاح خدمة CAMEL

الاختبار

الاختبار اليدوي باستخدام منشئ الطلب

1. افتح `https://your-server:8087/camel_request`
2. اختر **InitialDP**، قم بتعيين مفتاح الخدمة / الأرقام المتصلة / المدعوة، اختر نسخة CAP. لاحظ OTID المُولد. إذا رغبت، وأرسل.
3. افتح `https://your-server:8087/camel_sessions` لرؤية الجلسة في حالة `initiated`.
4. أرسل **EventReportBCSM** مع `oAnswer` لنقل الجلسة إلى `answered`.
5. أرسل **ReleaseCall** لإنهاء (عادي) `16` مع السبب.

الاختبار مع MSC حقيقي

قم بتكوين خدمة CAMEL على MSC/VLR الخاص بك (مفتاح الخدمة، عنوان gsmSCF العالمي) بحيث يحفز InitialDP نحو OmniSS7. مثال لمركز MSC من Huawei:

```
ADD CAMELSERVICE:
  SERVICEID=1,
  SERVICEKEY=100,
  GSMSCFADDR="55512341234",   # عنوان CAMELGW العالمي
  DEFAULTCALLHANDLING=CONTINUE;

ADD CAMELSUBSCRIBER:
  IMSI="310150123456789",
  SERVICEID=1,
  TRIGGERTYPE=TERMCALL;
```

مراقبة السجلات

يصدر OmniSS7 سجلات JSON منظمة (كائن واحد لكل سطر عبر `OmniLogger.JsonFormatter`)، لذا يمكنك تصفية على حقول البيانات الوصفية:

```
# تابع وتصفيه أحداث CAP حسب حقل map_event
tail -f /var/log/omniss7/omniss7.log | jq 'select(.map_event |
tostring | startswith("CAP:"))'
```

المراقبة والعمليات

مقاييس Prometheus

تقوم CAMELGW بعرض المقاييس عند `http://localhost:9090/metrics`.

مقاييس CAP:

- `cap_requests_total{operation}` - إجمالي طلبات CAP حسب العملية. قيم التسمية هي snake_case، مثل `initial_dp`، `connect`، `release_call`، `continue`، `apply_charging`.
- `cap_request_errors_total{operation}` - أخطاء توليد/إرسال طلب CAP حسب العملية.

مقاييس MAP / API:

- `map_requests_total{operation}` - إجمالي طلبات MAP حسب العملية
- `map_pending_requests` - عدد الانتظار لطلبات MAP TID

مقاييس M3UA STP (إذا تم تمكين وضع STP):

- `m3ua_stp_routing_failures_total{reason}` - إخفاقات التوجيه حسب السبب

```
# جميع عدادات طلبات CAP
curl http://localhost:9090/metrics | grep cap_requests_total

# InitialDP فقط (لاحظ تسمية snake_case)
curl http://localhost:9090/metrics | grep
'cap_requests_total{operation="initial_dp"}'
```

فحوصات الصحة

```
# حالة نظير M3UA (واجهة برمجة التطبيقات TLS 8445 على المنفذ)
curl -k https://localhost:8445/m3ua-status

# جلسات CAMEL النشطة (ترجع session_count + تفاصيل لكل جلسة)
curl -k https://localhost:8445/camel-sessions
```

لا يوجد نقطة نهاية `/api/ocs-status`، `/api/events`، أو `/api/camel/sessions/count`. تأتي أعداد الجلسات من `/camel-sessions`.

تكوين السجلات

قم بضبط مستوى السجل في `config/runtime.exs`:

```
config :logger, level: :info   # :debug | :info | :warning |
:error
```

تم تعيين المهيئ إلى JSON منظم في `config/config.exs`:

```
config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all,
  truncate: :infinity
```

استكشاف الأخطاء وإصلاحها

CAP المشكلة: لم يتم استلام رسائل

يعمل منشئ الطلب، لكن **الأعراض:** MSC لا يرسل InitialDP أبدًا.

تحقق من:

1. حالة نظير M3UA: `curl -k https://localhost:8445/m3ua-status`

2. تكوين خدمة CAMEL لمركز MSC (مفتاح الخدمة، عنوان gsmSCF)
3. توجيه SCCP (يجب أن يوجه عنوان gsmSCF إلى CAMELGW العالمي)
4. قواعد جدار الحماية (السماح بمنفذ SCTP 2905)

المشكلة: أخطاء CGrateS

الأعراض: `not_configured`، `http_error`، أو `cgrates_error` في السجلات.

تحقق من:

1. `cgrates_enabled: true` و `cgrates_url` تشير إلى نقطة نهاية JSON-RPC قابلة للوصول
2. الحساب (IMSI أو رقم المتصل) موجود ولديه رصيد في CGrateS
3. `cgrates_tenant` و `cgrates_request_type` تتطابق مع إعداد تقييم CGrateS الخاص بك
4. `cgrates_timeout` لديك CGrateS كبير بما يكفي لزمن استجابة

في المكالمة بدلاً من إسقاطها؛ يتم CAMELGW في حالة فشل الشحن أثناء المكالمة، يستمر تسجيل الخطأ.

المشكلة: الجلسة غير موجودة

الأعراض: تسجل EventReportBCSM "الجلسة غير موجودة".

السبب: عدم تطابق OTID أو تم إنهاء الجلسة بالفعل.

تحقق من: تحقق من OTID في السجلات وتأكد من أن DTID/OTID تتطابق عبر الحوار.

المشكلة: أخطاء فك التشفير

الأعراض: "فشل في فك تشفير InitialDP" في السجلات.

السبب: عدم تطابق نسخة CAP أو رسالة مشوهة.

تحقق من: تأكد من أن إعداد نسخة CAP يتطابق مع MSC، والتقاط PCAP للتحليل:

```
tcpdump -i eth0 -w cap_trace.pcap sctp port 2905
```

الملخص

يمكن وضع بوابة CAMEL OmniSS7 للعمل كـ gsmSCF مع:

- **دعم بروتوكول CAP** (v1/v2/v3/v4) مع تجاوز النسخة لكل طلب
- **شحن في الوقت الحقيقي** عبر واجهة برمجة التطبيقات JSON-RPC لـ CGrateS SessionS
- **عمليات التحكم في المكالمات** (Connect، Release، Continue، ApplyCharging)
- **إدارة الجلسات** في ETS، مفاتيحها بواسطة OTID
- **اختبار تفاعلي** عبر واجهة الويب منشئ الطلب
- **مراقبة حية** للجلسات النشطة (واجهة الويب وواجهة برمجة التطبيقات `/camel-sessions`)
- **توليد CDR** بواسطة CGrateS للفوترة والتحليلات

للمزيد من التفاصيل:

- المرجع الفني - عمليات CAP

إصدار الوثائق: 1.1 **المنتج:** OmniSS7 CAMEL Gateway

دليل الميزات الشائعة

← العودة إلى الوثائق الرئيسية

يغطي هذا الدليل الميزات الشائعة لجميع أوضاع تشغيل OmniSS7.

جدول المحتويات

1. نظرة عامة على واجهة الويب
2. وثائق API
3. المراقبة والقياسات
4. أفضل الممارسات
5. تعدد الموصلات SCTP من أجل تكرار الشبكة

نظرة عامة على واجهة الويب

تُقدم واجهة الويب عبر HTTPS على المنفذ `8087` (يتم فرض TLS عبر `force_ssl`)، على سبيل المثال `https://localhost:8087/`.

التنقل الرئيسي

تُبنى القائمة من الصفحات المفعلة في `config/config.exs` وترتيبها حسب `page_order`. الإدخالات الشائعة (تسمية القائمة بالخط العريض):

- **الموارد** (`/application`) - حالة التطبيق ومعلومات وقت التشغيل
- **التكوين** (`/configuration`) - عارض تكوين النظام
- **أحداث SS7** (`/events`) - أحداث الإشارة SS7 وسجلات الرسائل في الوقت الحقيقي
- **سجلات النظام** (`/logs`) - عارض سجلات التطبيق
- **عميل SS7** (`/client`) - عميل اختبار طلب MAP تفاعلي
- **الأقران** (`/m3ua`) - اتصالات نظير M3UA/M2PA
- **التوجيه** (`/routing`) - رموز النقاط ومسارات العنوان العالمي
- **اختبار التوجيه** (`/routing_test`) - أداة اختبار حل المسار
- **روابط SMSc** (`/smsc_links`) - روابط خلفية SMS (SMSc)
- **مشتركي SMSc** (`/smsc_subscribers`) - مشتركون في SMS يتم تتبعهم (SMSc)
- **روابط HLR** (`/hlr_links`) - روابط خلفية HLR (HLR)

- **المشتركون النشطون** (`/subscribers`) - مشتركون في تحديث الموقع يتم تتبعهم
- **جلسات CAMEL** (`/camel_sessions`) - جلسات مكالمات CAMEL الحية (CAMEL GW)
- **باني طلب CAMEL** (`/camel_request`) - باني طلب CAP تفاعلي (CAMEL GW)
- **اختبار MSISDN/IMSI** (`/msisdn_imsi_test`) - أداة اختبار تحويل الأرقام

الوصول إلى واجهة الويب

1. افتح متصفح الويب الخاص بك
2. انتقل إلى المضيف المكون على المنفذ 8087 (على سبيل المثال، `https://localhost:8087/`)
3. عرض لوحة معلومات حالة النظام

وثائق Swagger API

تُقدم وثائق API التفاعلية بواسطة واجهة MAP على منفذ HTTP 9090:

```
http://your-server:9090/swagger
```

تكوين واجهة الويب

نقطة نهاية واجهة الويب مُكونة تحت `config :control_panel, ControlPanelWeb.Endpoint` في `config/config.exs`، وقائمة التنقل تحت `config :control_panel`:

```
config :control_panel, ControlPanelWeb.Endpoint,
  force_ssl: [rewrite_on: [:x_forwarded_proto]],
  server: true,
  url: [host: "0.0.0.0", path: "/"],
  https: [
    port: 8087,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ]

config :control_panel,
  page_order: [
    "/application", "/configuration", "/events", "/logs",
"/client",
    "/m3ua", "/routing", "/routing_test", "/smsc_links",
"/smsc_subscribers",
    "/hlr_links", "/subscribers", "/camel_sessions",
"/camel_request",
    "/msisdn_imsi_test"
  ]
```

معلمات التكوين:

المعلمة	النوع	الافتراضي
`ControlPanelWeb.Endpoint.url.host`	سلسلة	`"0.0.0.0"`
`ControlPanelWeb.Endpoint.https.port`	عدد صحيح	`8087`
`ControlPanelWeb.Endpoint.https.keyfile`	سلسلة	`"priv/cert/omnito`
`ControlPanelWeb.Endpoint.https.certfile`	سلسلة	`"priv/cert/omnito`
`ControlPanelWeb.Endpoint.force_ssl`	كلمة رئيسية	`[rewrite_on: [:x_forwarded_prot`
`page_order`	قائمة من	انظر المثال أعلاه

المعلمة	النوع	الافتراضي
	السلاسل	

> انظر إلى مرجع التكوين للحصول على مخطط لوحة التحكم الكامل.

تكوين السجل

تُصدر السجلات كـ JSON بواسطة (كائن واحد لكل سطر) منظم `OmniLogger.JsonFormatter`، والذي يتضمن جميع `Logger.metadata` بحيث يتم حمل عملية/otid، MSISDNs من/إلى SMS، بوابة STP OPC/DPC/GTs، توجيه) السياق لكل عملية (المعاملة. يتم تكوين هذا التنسيق في `config/config.exs` تلقائيًا:

```
config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all,
  truncate: :infinity
```

يتم تعيين مستوى السجل بشكل منفصل:

```
config :logger,
  level: :debug  # الخيارات: :debug، :info، :warning، :error
```

معلمات التكوين:

المعلمة	النوع	الافتراضي
`:logger.level`	ذرة	`:debug`
`:logger.default_formatter.format`	مجموعة	`{OmniLogger.JsonFormatt :format}`
`:logger.default_formatter.metadata`	ذرة/قائمة	`:all`
`:logger.default_formatter.truncate`	عدد صحيح/ذرة	`:infinity`

وثائق API

يقدم OmniSS7 واجهتين HTTP:

السطح	المنفذ	البروتوكول	المحتويات
واجهة MAP	`9090`	HTTP	عمليات MAP `/api/*`، `/swagger`، `/swagger.json`، `/metrics`
واجهة الحالة JSON (`api_ex`)	`8445`	HTTPS	حالة القراءة فقط (`/api/status`، `/api/m3ua-status`، `/api/routing`، `/api/subscribers`، `/api/camel-sessions`، ...)

عنوان URL الأساسي لواجهة MAP API

```
http://your-server:9090/api
```

رموز الاستجابة

- **200** - نجاح
- **400** - طلب غير صحيح
- **504** - مهلة البوابة

مواصفة OpenAPI

```
http://your-server:9090/swagger.json
```

المراقبة والقياسات

نقطة نهاية قياسات Prometheus

```
http://your-server:9090/metrics
```

فئات القياسات الرئيسية

قياسات M3UA/SCTP:

- تغييرات حالة ارتباط SCTP
- انتقالات حالة M3UA ASP
- وحدات البيانات البروتوكول المرسلة/المستلمة

قياسات M2PA:

- انتقالات حالة الرابط (DOWN → ALIGNMENT → PROVING → READY)
- الرسائل والبايتات المرسلة/المستلمة لكل رابط
- أخطاء محددة للرابط (فك التشفير، تشفير، SCTP)

قياسات STP:

- الرسائل المستلمة/المرسلة لكل نظير
- فشل التوجيه حسب السبب
- توزيع الحركة عبر الأقران

قياسات عميل MAP:

- طلبات MAP حسب نوع العملية
- مخططات مدة الطلب
- مقياس المعاملات المعلقة

قياسات CAP:

- طلبات CAP حسب نوع العملية
- عمليات بوابة CAMEL

قياسات SMSc:

- عمق الطابور
- معدلات التسليم
- الرسائل الفاشلة

تكامل Grafana

تتوافق قياسات OmniSS7 مع Prometheus وGrafana.

أفضل الممارسات

توصيات الأمان

1. **عزل الشبكة**
    - نشر في VLAN مخصص
    - قواعد جدار الحماية لتقييد الوصول
    - السماح لـ SCTP فقط من عناوين معروفة
2. **أمان واجهة الويب**
    - تمكين TLS للإنتاج
    - استخدام وكيل عكسي مع المصادقة
    - تقييد الوصول إلى عناوين IP الإدارية
3. **أمان API**
    - تنفيذ تحديد المعدل
    - استخدام مفاتيح API أو OAuth
    - تسجيل جميع الطلبات للتدقيق

ضبط الأداء

1. **دود TPS**
    - تكوين TPS مناسب
    - مراقبة حمل النظام
    - ضبط مخازن SCTP
2. **ضبط خلفية SMS**

- سريعة الاستجابة (تنتهي الطلبات بعد 5 SMS الحفاظ على واجهة خلفية ثوانٍ)
- ضبط `auto_flush_tps` و `auto_flush_interval` لتتناسب مع سعة الخلفية
- مراقبة عمق الطابور ومعدلات التسليم

3. **ضبط M3UA**

- ضبط فترات نبض SCTP
- تكوين قيم المهلة
- استخدام روابط متعددة من أجل التكرار

من أجل تكرار SCTP تعدد الموصلات الشبكة

ما هو تعدد الموصلات SCTP؟

تعدد الموصلات SCTP هي ميزة مدمجة في بروتوكول SCTP تسمح لاتصال واحد M3UA بالارتباط بعدة عناوين IP على نفس واجهة الشبكة أو عبر واجهات شبكة مختلفة. يوفر ذلك الفشل التلقائي والتكرار على مستوى النقل.

الفوائد الرئيسية:

- **الفشل التلقائي** SCTP: إذا فشل أحد مسارات الشبكة، يقوم بالتبديل تلقائيًا إلى مسار بديل دون قطع الاتصال
- **الفشل بدون تكوين** SCTP: لا حاجة ��لى منطق على مستوى التطبيق - يتولى مراقبة المسار والفشل
- **موثوقية محسّنة**: البقاء على قيد الحياة في حالات فشل الشبكة، وفشل التبديل، أو فشل NIC
- **توازن الحمل** SCTP: يمكن لـ (يعتمد على التنفيذ) توزيع الحركة عبر مسارات متعددة

كيف يعمل

عند تكوين عدة عناوين IP لاتصال M3UA، يقوم SCTP:

1. **الارتباط بجميع IPs**: يرتبط المقبس بجميع عناوين IP المكونة في نفس الوقت
2. **مراقبة المسارات**: يرسل SCTP باستمرار حزم نبض القلب على جميع المسارات لمراقبة صحتها
3. **اكتشاف الفشل**: إذا فشلت نبضات القلب على المسار الرئيسي، يقوم SCTP بوضع علامة عليه على أنه غير قابل للوصول
4. **الفشل التلقائي**: يتم التبديل على الفور إلى مسار احتياطي دون تدخل من التطبيق
5. **استعادة المسار**: عندما يتم استعادة المسار الفاشل، يكتشف SCTP ذلك ويضع علامة عليه كمتاح مرة أخرى

التكوين

يتم تكوين تعدد الموصلات SCTP **قائمة من عناوين IP** بدلاً من مجموعة **IP** واحدة.

IP واحدة (تقليدية)

```
# IP واحدة - لا يوجد تعدد موصلات
local_ip: {10, 179, 4, 10}
```

عدة IPs (تمكين تعدد الموصلات)

```
# عدة IPs - تمكين تعدد الموصلات
# IP الأول هو الرئيسي، والـ IPs التالية هي مسارات احتياطية
local_ip: [{10, 179, 4, 10}, {10, 179, 4, 11}]
```

أمثلة التكوين

> **ملاحظة مهمة لدور الخادم (الاتصالات الواردة):**
>
> قبول الاتصالات الواردة من الأقران متعددة) `role: :server` عند تكوين الأقران مع الذي يستخدمه IP كـ **مجموعة واحدة** - عنوان `remote_ip` يجب عليك تحديد ،(الموصلات النظير البعيد لبدء اتصال SCTP (SCTP INIT). لا تستخدم قائمة.
>
> **لماذا؟** يقوم STP بمطابقة الاتصالات الواردة بناءً على عنوان المصدر لحزمة SCTP INIT. الأخرى متعددة الموصلات للنظير أثناء مصافحة الارتباط. IP تلقائيًا عناوين SCTP سيكتشف (الاتصالات الصادرة) `role: :client` صالح فقط لـ `remote_ip` تنسيق القائمة لـ.

أمثلة:

```
# ✓ صحيح - دور الخادم مع remote_ip واحدة
%&#123;
  role: :server,
  remote_ip: &#123;10, 0, 2, 100&#125;,  # مجموعة واحدة
  # ...
&#125;

# ✗ خاطئ - دور الخادم مع قائمة لن تتطابق مع الاتصالات الواردة
%&#123;
  role: :server,
  remote_ip: [&#123;10, 0, 2, 100&#125;, &#123;10, 0, 2,
101&#125;],  # القائمة لن تعمل
  # ...
&#125;
```

المثال 1: نظير STP (دور العميل - صادرة) مع تعدد الموصلات

```
# تكوين نظير وضع STP (الاتصال الصادر)
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "Partner_STP_Redundant",
      role: :client,  # صادرة - نحن نبدأ الاتصال
      # تعدد الموصلات: الارتباط باثنين من IPs المحلية من أجل التكرار
      local_ip: [{213, 57, 23, 200}, {213, 57, 23, 201}],
      local_port: 0,
      # النظير البعيد يدعم أيضًا تعدد الموصلات - القائمة مقبولة لدور
العميل
      remote_ip: [{213, 57, 23, 100}, {213, 57, 23, 101}],
      remote_port: 2905,
      routing_context: 1,
      point_code: 100,
      network_indicator: :international
    }
  ]
```

المثال 2: عميل MAP مع تعدد الموصلات

```
# وضع عميل MAP مع تعدد الموصلات
config :omniss7,
  map_client_enabled: true,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    # تعدد الموصلات: اثنين من IPs المحلية من أجل الفشل
    local_ip: [{10, 0, 0, 100}, {10, 0, 0, 101}],
    local_port: 2905,
    # STP البعيد مع دعم تعدد الموصلات
    remote_ip: [{10, 0, 0, 1}, {10, 0, 0, 2}],
    remote_port: 2905,
    routing_context: 1
  }
```

المثال 3: مستمع STP مع تعدد الموصلات

```
# خادم STP مستقل مع تعدد الموصلات
config :omniss7,
  sctp_handler: %{
    enabled: true,
    # IPs من أجل الاتصالات الواردة الاستماع على عدة
    local_ip: [{172, 16, 0, 10}, {172, 16, 0, 11}],
    local_port: 2905,
    point_code: 100
  }
```

المثال 4: دور الخادم - قبول الوارد من نظير متعدد الموصلات

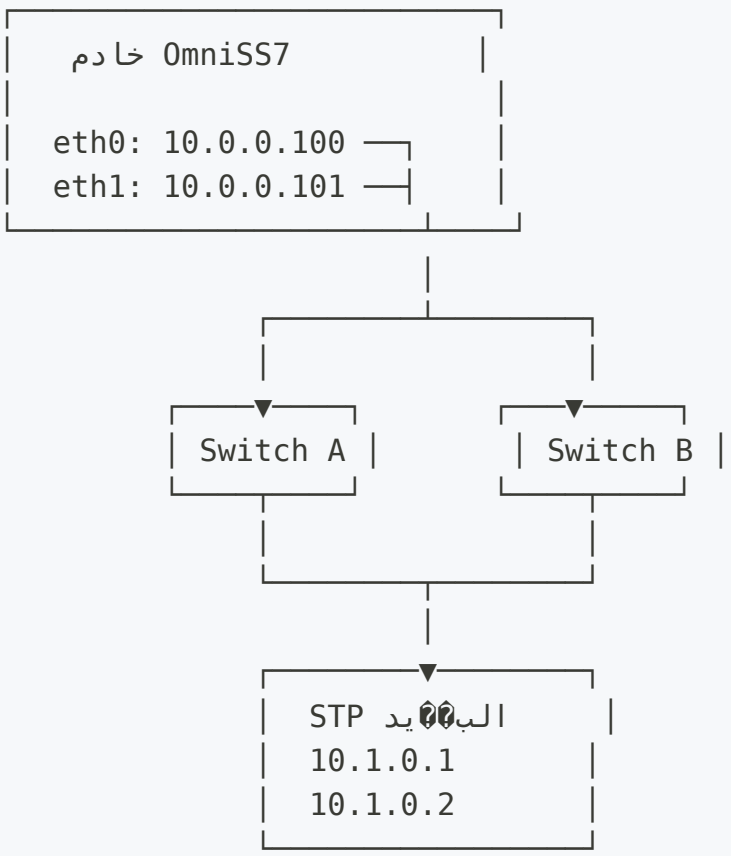
```
# متعدد الموصلات HLR قبول اتصال وارد من
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "HLR",
      role: :server,  # وارد - HLR يتصل بنا
      # تعدد الموصلات: IPs (القائمة مقبولة) المحلية لدينا
      local_ip: [{10, 0, 1, 10}, {10, 0, 1, 11}],
      local_port: 2905,
      # مهم: مجموعة واحدة فقط - IP الذي يستخدمه HLR لبدء SCTP INIT
      # أثناء المصافحة HLR الأخرى لـ IPs تلقائيًا SCTP سيكتشف
      remote_ip: {10, 0, 2, 100},  # IP (ليس قائمة!) الرئيسي فقط
      remote_port: 0,  # قبول من أي منفذ مصدر
      routing_context: 1,
      point_code: 100,
      network_indicator: :international
    }
  ]
```

المثال 5: تكوين مختلط (متوافق مع الإصدارات السابقة)

```
# مزيج من الأقران ذات IP واحدة ومتعددة الموصلات
config :omniss7,
  m3ua_peers: [
    # نظير قديم - IP واحدة
    %{
      peer_id: 1,
      name: "Legacy_STP",
      role: :client,
      local_ip: {10, 0, 0, 1},      # مجموعة IP واحدة
      local_port: 0,
      remote_ip: {10, 0, 0, 10},
      remote_port: 2905,
      routing_context: 1,
      point_code: 100
    },
    # نظير جديد - تعدد الموصلات
    %{
      peer_id: 2,
      name: "Redundant_STP",
      role: :client,
      local_ip: [{10, 0, 0, 2}, {10, 0, 0, 3}],  # قائمة IP
      local_port: 0,
      remote_ip: [{10, 0, 0, 20}, {10, 0, 0, 21}],
      remote_port: 2905,
      routing_context: 2,
      point_code: 200
    }
  ]
```

سيناريوهات تخطيط الشبكة

السيناريو 1: بطاقات NIC مزدوجة (نشر شائع)



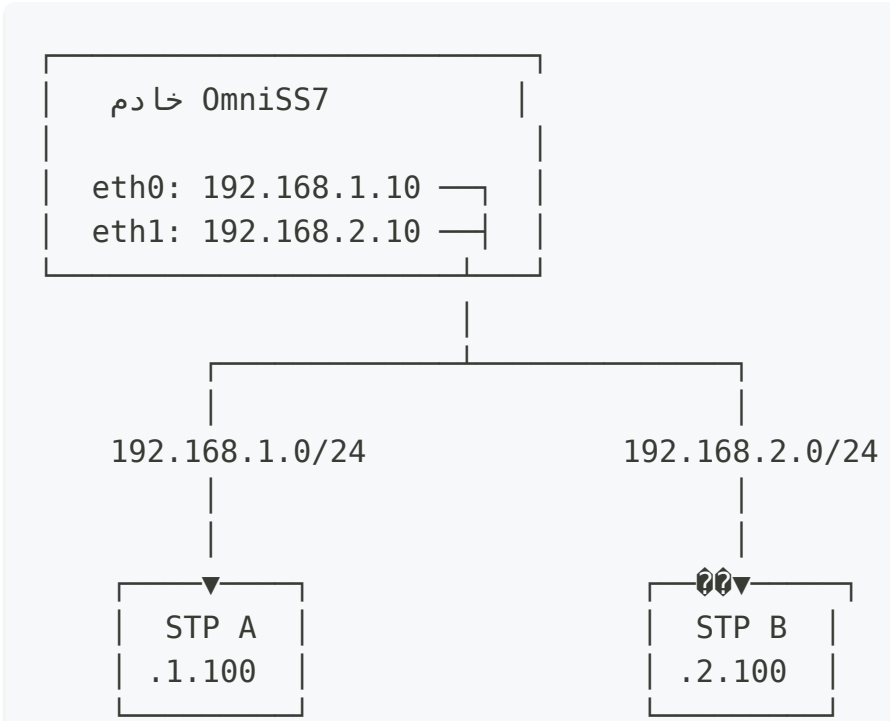
التكوين:

```
local_ip: [{10, 0, 0, 100}, {10, 0, 0, 101}]   # كلا بطاقتي NIC
remote_ip: [{10, 1, 0, 1}, {10, 1, 0, 2}]      # النظير البعيد
```

الفوائد:

- البقاء على قيد الحياة في فشل أحد بطاقات NIC
- البقاء على قيد الحياة في فشل أحد المحولات
- الفشل التلقائي في أقل من 1 ثانية

السيناريو 2: عدة شبكات فرعية



التكوين:

```
local_ip: [{192, 168, 1, 10}, {192, 168, 2, 10}]
remote_ip: [{192, 168, 1, 100}, {192, 168, 2, 100}]
```

الفوائد:

- البقاء على قيد الحياة في فشل الشبكة الفرعية
- إمكانية التكرار الجغرافي
- مسارات توجيه مستقلة

المراقبة والتسجيل

عند تمكين تعدد الموصلات، سترى رسائل سجل تشير إلى التكوين:

تعدد الموصلات الناجح

```
[info] عميل SCTP من مرتبط باثنين متعدد الموصلات: IPs المحلية
[info] مستمع STP 2 تم تمكين تعدد الموصلات: IPs محلية مرتبطة
```

أحداث فشل المسار

```
[warning] [MULTIHOMING] المسار 10.0.0.100 غير قابل للوصول للنظير
Partner_STP (assoc_id=1)
[info] [MULTIHOMING] المسار 10.0.0.101 هو الآن الرئيسي للنظير
Partner_STP (assoc_id=1)
[info] [MULTIHOMING] المسار 10.0.0.100 متاح الآن للنظير Partner_STP
(assoc_id=1)
```

عرض واجهة الويب

تظهر واجهة الويب تلقائيًا معلومات تعدد الموصلات:

صفحة حالة M3UA:

- **IP واحدة**: تظهر كـ `10.0.0.100`
- **عدة IPs**: تظهر كـ `10.0.0.100 (1+)` أو `10.0.0.100 (2+)`
- **عرض التفاصيل**: يظهر جميع IPs احتياطية/رئيسية مع تسميات

أفضل الممارسات

1. تصميم الشبكة

- **استخدم بطاقات NIC مختلفة** لتحقيق أقصى قدر من التكرار
- **محولات مختلفة** للبقاء على قيد الحياة في فشل المحولات
- **شبكات فرعية مختلفة** إذا كان ذلك ممكنًا لتنوع التوجيه
- **نفس مركز البيانات في البداية** - اختبر قبل الفصل الجغرافي

2. تخطيط عنوان IP

- **الأول هو الرئيسي IP** - تأكد من أنه على المسار الأكثر موثوقية
- **الترتيب مهم** - قم بإدراج IPs حسب ترتيب التفضيل
- **توجيه متسق** - استخدم أنظمة توجيه مشابهة لتسهيل استكشاف الأخطاء وإصلاحها

3. اختبار الفشل

```
# تعطيل واجهة IP الرئيسية لاختبار الفشل
sudo ip link set eth0 down

# مراقبة السجلات للفشل
tail -f /var/log/omniss7.log | grep MULTIHOMING

# إعادة تمكين الواجهة
sudo ip link set eth0 up
```

4. يجب أن يدعم الجانبان تعدد الموصلات

- **الأمثل**: كلا الجانبين المحلي والبعيد يستخدمان عدة IPs
- **مقبول**: جانب واحد فقط يستخدم تعدد الموصلات
- **ملاحظة**: التكرار هو الأفضل عندما يدعم كلا الطرفين ذلك

5. تكوين جدار الحماية

```
# السماح لـ SCTP على جميع IPs الموصلات متعددة
iptables -A INPUT -p sctp --dport 2905 -s 10.0.0.0/24 -j ACCEPT
iptables -A INPUT -p sctp --dport 2905 -s 10.1.0.0/24 -j ACCEPT
```

استكشاف الأخطاء وإصلاحها

المشكلة: عدم عمل تعدد الموصلات

الأعراض: يتم استخدام IP الرئيسي فقط، ولا يوجد فشل

التحققات:

1. تحقق من دعم Erlang لـ SCTP: `erl -eval 'gen_sctp:open(9999, [binary, {ip, {127,0,0,1}}]).'`
2. تحقق من وحدة SCTP في النواة: `lsmod | grep sctp`
3. إذا لزم الأمر قم بتحميل SCTP: `sudo modprobe sctp`
4. تحقق من تكوين كلا IPs على النظام: `ip addr show`

المشكلة: عدم فشل المسار

الأعراض: يتم وضع علامة على المسار الرئيسي على أنه معطل ولكن الحركة لا تتبدل

التحققات:

1. تحقق من إعدادات نبض SCTP
2. تحقق من أن جدول التوجيه يحتوي على مسارات لجميع المسارات
3. تحقق من أن جدار الحماية يسمح بـ SCTP على جميع IPs
4. راجع سجلات مراقبة مسار SCTP

المشكلة: تذبذب المسار المتكرر

الأعراض: تتبدل المسارات باستمرار بين التشغيل والإيقاف

التحققات:

1. عدم استقرار الشبكة - تحقق من الروابط الفيزيائية
2. عدواني جدًا - قد يحتاج إلى ضبط نبض SCTP
3. جدار الحماية يقوم بإسقاط نبضات SCTP
4. مشكلات MTU على أحد المسارات

اعتبارات الأداء

- **أقل تأثير**: نبضات SCTP صغيرة وغير متكررة
- **لا تغييرات على مستوى التطبيق**: تعدد الموصلات شفاف لطبقة التطبيق
- **فشل سريع**: عادةً أقل من 1 ثانية للكشف والفشل
- **استعادة تلقائية**: لا حاجة لتدخل يدوي

التوافق

- **متوافق مع الإصدارات السابقة**: لا يزال تنسيق مجموعة IP واحدة يعمل
- **نشر مختلط**: يمكن خلط الأقران ذات IP واحدة ومتعددة
- **تدعم جميع الأوضاع**: تعمل في أوضاع MAP Client وSMSc وHLR وSTP
- **متطلبات Erlang**: تتطلب Erlang مع دعم SCTP مُجمع فيه

المراقبة والتنبيه

المقاييس الرئيسية:

- حالة اتصال M3UA

- معدل نجاح طلبات MAP
- أوقات استجابة API
- عمق طابور الرسائل

حدود التنبيه:

- معطلة > 1 دقيقة M3UA
- معدل مهلة MAP > 10%
- عمق الطابور > 1000
- معدل أخطاء API > 5%

مرجع التكوين الكامل

جميع معلمات التكوين

يوفر هذا القسم مرجعًا كاملاً لجميع معلمات التكوين المتاحة عبر جميع أوضاع التشغيل.

تكوين السجل (`:logger`)

```
config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all,
  truncate: :infinity

config :logger,
  level: :debug  # :debug | :info | :warning | :error
```

المعلمة	النوع	مطلوب	راضي
`:logger.level`	ذرة	لا	`:debug`
`:logger.default_formatter.format`	مجموعة	لا	`{OmniLogger.Js :format}`
`:logger.default_formatter.metadata`	ذرة/ قائمة	لا	`:all`
`:logger.default_formatter.truncate`	عدد صحيح/ ذرة	لا	`:infinity`

تكوين واجهة الويب (`:control_panel`)

```
config :control_panel, ControlPanelWeb.Endpoint,
  force_ssl: [rewrite_on: [:x_forwarded_proto]],
  server: true,
  url: [host: "0.0.0.0", path: "/"],
  https: [
    port: 8087,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ]

config :control_panel,
  page_order: [
    "/application", "/configuration", "/events", "/logs",
"/client",
    "/m3ua", "/routing", "/routing_test", "/smsc_links",
"/smsc_subscribers",
    "/hlr_links", "/subscribers", "/camel_sessions",
"/camel_request",
    "/msisdn_imsi_test"
  ]
```

المعلمة	النوع	مطلوب	
`ControlPanelWeb.Endpoint.url.host`	سلسلة	لا	`"0.0.0.0`
`ControlPanelWeb.Endpoint.https.port`	عدد صحيح	لا	`8087`
`ControlPanelWeb.Endpoint.https.keyfile`	سلسلة	لا	`"priv/ce`
`ControlPanelWeb.Endpoint.https.certfile`	سلسلة	لا	`"priv/ce`
`ControlPanelWeb.Endpoint.force_ssl`	كلمة رئيسية	لا	`[rewrite_` `[:x_forwa`
`page_order`	قائمة من السلاسل	لا	المثال أعلاه

تكوين SocketHandler SCTP (`:omniss7`)

```
config :omniss7,
  sctp_handler: %{
    enabled: false,
    local_ip: {127, 0, 0, 1},
    local_port: 2905
  },
  enable_gt_routing: true,
  m3ua_peers: [...],
  m3ua_routes: [...],
  m3ua_gt_routes: [...]
```

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`sctp_handler.enabled`	Boolean	نعم	`false`	تمكين وضع STP عند الإقلاع
`sctp_handler.local_ip`	مجموعة	نعم	`{127, 0, 0, 1}`	IP للارتباط بالاتصالات الواردة M3UA
`sctp_handler.local_port`	عدد صحيح	نعم	`2905`	منفذ SCTP لـ M3UA
`enable_gt_routing`	Boolean	لا	`false`	تمكين توجيه العنوان العالمي

معلمات نظير M3UA:

المعلمة	النوع	مطلوب	الوصف
`peer_id`	عدد صحيح	نعم	معرف نظير فريد
`name`	سلسلة	نعم	اسم وصفي للنظير
`role`	ذرة	نعم	`:server` أو `:client`
`local_ip`	مجموعة أو قائمة	إذا `:client`	IPs المحلية للارتباط. واحدة: `{10, 0, 0, 1}` أو قائمة: `[{10, 0, 0, 1}, {10, 0, 0, 2}]`
`local_port`	عدد صحيح	إذا `:client`	المنفذ المحلي (0 للديناميكي)
`remote_ip`	مجموعة أو قائمة	نعم	IPs النظير البعيد. واحدة: `{10, 0, 0, 10}` أو قائمة: `[{10, 0, 0, 10}, {10, 0, 0, 11}]`
`remote_port`	عدد صحيح	إذا `:client`	المنفذ البعيد للنظير
`routing_context`	عدد صحيح	نعم	سياق توجيه M3UA
`point_code`	عدد صحيح	نعم	رمز نقطة SS7
`network_indicator`	ذرة	لا	`:international` أو `:national`

معلمات مسار M3UA:

الوصف	مطلوب	النوع	المعلمة
رمز النقطة الوجهة	نعم	عدد صحيح	`dest_pc`
النظير الذي يتم التوجيه من خلاله	نعم	عدد صحيح	`peer_id`
أولوية المسار (الأقل = أعلى أولوية)	نعم	عدد صحيح	`priority`
`:national` أو `:international`	لا	ذرة	`network_indicator`

معلمات مسار GT M3UA:

الوصف	مطلوب	النوع	المعلمة
بادئة العنوان العالمي للمطابقة	نعم	سلسلة	`gt_prefix`
النظير الوجهة	نعم	عدد صحيح	`peer_id`
أولوية المسار	نعم	عدد صحيح	`priority`
وصف المسار للتسجيل	لا	سلسلة	`description`
المصدر SSN المطابقة فقط إذا تطابقت	لا	عدد صحيح	`source_ssn`
الوجهة إلى هذه القيمة SSN إعادة كتابة	لا	عدد صحيح	`dest_ssn`

تكوين عميل MAP (`:omniss7`)

```
config :omniss7,
  map_client_enabled: false,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :map_client_asp,
    local_ip: {10, 0, 0, 100},
    local_port: 2905,
    remote_ip: {10, 0, 0, 1},
    remote_port: 2905,
    routing_context: 1
  }
```

المعلمة	النوع	مطلوب	الافتراضي
`map_client_enabled`	Boolean	نعم	`false`
`map_client_m3ua.mode`	سلسلة	نعم	`"ASP"`
`map_client_m3ua.callback`	مجموعة	نعم	`{MapClient, :handle_payload, []}`
`map_client_m3ua.process_name`	ذرة	نعم	`:map_client_asp`
`map_client_m3ua.local_ip`	مجموعة	نعم	-
`map_client_m3ua.local_port`	عدد صحيح	نعم	`2905`
`map_client_m3ua.remote_ip`	مجموعة	نعم	-
`map_client_m3ua.remote_port`	عدد صحيح	نعم	`2905`

المعلمة	النوع	مطلوب	الافتراضي
`map_client_m3ua.routing_context`	عدد صحيح	نعم	-

تكوين مركز SMS (`:omniss7`)

```
config :omniss7,
  auto_flush_enabled: false,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: nil,
  auto_flush_tps: 10
```

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`auto_flush_enabled`	Boolean	لا	`false`	تمكين التفريغ التلقائي لطابور SMS
`auto_flush_interval`	عدد صحيح	لا	`10000`	فترة استعلام الطابور (بالملي ثانية)
`auto_flush_dest_smsc`	سلسلة/nil	لا	`nil`	تصفية حسب الوجهة SMSC (nil = الكل)
`auto_flush_tps`	عدد صحيح	لا	`10`	الحد الأقصى للمعاملات في الثانية

تكوين واجهة API HTTP (`:omniss7`)

تستخدم خلفية SMS الآن واجهة API HTTP بقاعدة البيانات مباشرة بدلاً من الاتصالات المباشرة.

```
config :omniss7,
  smsc_api_base_url: "https://omnimessage.example.com:8443",
  smsc_name: "omni-smsc01"  # اختياري: الافتراضي هو اسم مستمد من اسم المضيف
```

معلمات API:

المعلمة	النوع	مطلوب	الافتراضي
`smsc_api_base_url`	سلسلة	نعم	`"https://omnimessage.example.com`
`smsc_name`	سلسلة	لا	مستمد من اسم المضيف

نقاط النهاية المستخدمة في API:

- `POST /api/frontends` - تسجيل هذه الواجهة الأمامية مع الخلفية
- `POST /api/messages_raw` - إدراج رسائل SMS جديدة
- `GET /api/messages` - استرجاع طابور الرسائل (مع رأس `smsc`)
- `PATCH /api/messages/{id}` - وضع علامة على الرسالة كمُسلمة
- `PUT /api/messages/{id}` - تحديث حالة الرسالة
- `POST /api/events` - إضافة تتبع الأحداث
- `GET /api/status` - نقطة نهاية فحص الصحة

تسجيل الواجهة الأمامية:

الخلفية عند بدء التشغيل ويعيد التسجيل تقريبًا كل 85 API يسجل النظام نفسه تلقائيًا مع واجهة ثانية. يتضمن التسجيل:

- اسم الواجهة الأمامية ونوعها (يتم الإبلاغ عنها كـ `SS7`)
- اسم المضيف
- وقت التشغيل بالثواني
- تفاصيل التكوين (بتنسيق JSON)

ملاحظات التكوين:

- يتم تعطيل التحقق من SSL بشكل افتراضي للشهادات الموقعة ذاتيًا
- تنتهي مهلات طلبات HTTP بعد 5 ثوانٍ
- جميع الطوابع الزمنية بتنسيق ISO 8601
- تستخدم واجهة API JSON لأجسام الطلب/الاستجابة

الوثائق ذات الصلة

- العودة إلى الوثائق الرئيسية ←
- دليل STP
- دليل عميل MAP
- دليل مركز SMS
- دليل HLR

OmniSS7 بواسطة خدمات شبكة Omnitouch

مرجع التكوين

← العودة إلى الوثائق الرئيسية

تقدم هذه الوثيقة مرجعًا شاملاً لجميع معلمات تكوين OmniSS7.

جدول المحتويات

1. نظرة عامة
2. أين توجد التكوينات
3. علامات وضع التشغيل
4. معلمات وضع HLR
5. معلمات وضع SMSc
6. معلمات وضع STP
7. معلمات وضع بوابة CAMEL
8. معلمات NAT لعنوان عالمي
9. معلمات اتصال M3UA
10. حالة نظام SCCP
11. تصفية البوابة
12. معلمات بوابة USSD
13. علامات وقت التشغيل / الإشراف
14. معلمات البنية التحتية
15. معلمات قاعدة البيانات
16. قيم ثابتة

نظرة عامة

يدعم OmniSS7 أربعة أوضاع تشغيل، يتم اختيارها بالكامل من خلال التكوين:

- **وضع STP** - نقطة نقل الإشارة للتوجيه
- **وضع HLR** - سجل الموقع المنزلي لإدارة المشتركين

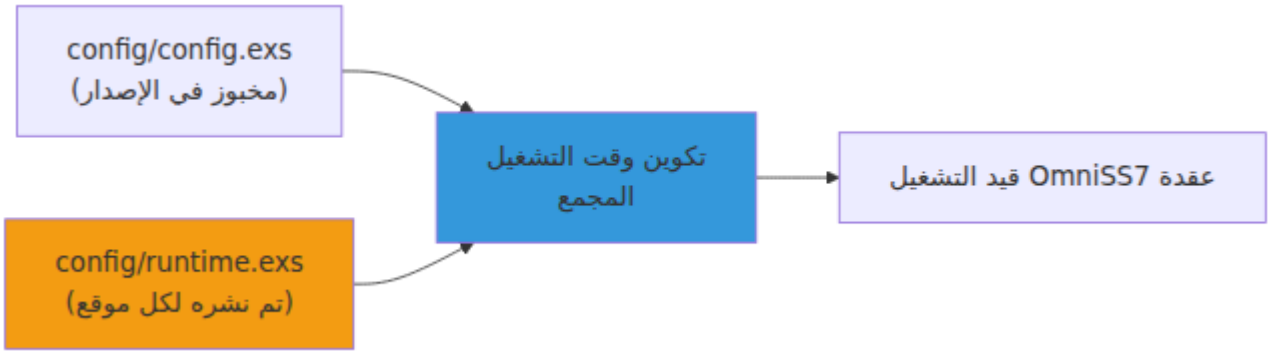
- **وضع SMSc** - مركز الرسائل القصيرة لتوصيل الرسائل
- **وضع CAMEL GW** - بوابة CAMEL للتحكم الذكي في المكالمات

من خلال تمكين (HLR + SMSc على سبيل المثال) يمكن أن يجمع العقدة الواحدة بين الأدوار علامات وضع التشغيل.

أين توجد التكوينات

يقرأ OmniSS7 تكوينه من طبقتين. فهم أي ملف يمتلك قيمة يحدد مكان تغييرها في نظام تم نشره.

الملف	الدور	هل يتم التعديل عند النشر؟
`config/config.exs`	**التكوين الافتراضي / وقت الترجمة** المرسل مع الإصدار. يحدد صفحات لوحة التحكم، ووحدات التحكم REST `:api_ex`، ومُنسق السجلات، وجميع القيم الافتراضية الأساسية.	لا: جزء من البناء.
`config/runtime.exs` (في المستودع)	**نموذج اختبار مكون من 4 أسطر** فقط. في شجرة المصدر لا يقوم بإعداد أي شيء يتجاوز بيئة الاختبار.	لا.
`config/runtime.exs` (المستضاف)	**تكوين وقت التشغيل لكل نشر**، يتم عادة) نشره بشكل منفصل على كل خادم Ansible/Jinja). ما يتم تقديمه من قالب هنا يتم تعيين علامات الوضع، GTs، رموز M3UA، وواجهات برمجة النقاط، أقران التطبيقات لموقع معين.	**نعم. هذا هو الملف الموجه للمشغل.**



> **باختصار:** تستخدم الأمثلة في هذه الوثيقة `config :omniss7, ...`. على عقدة حية، تذهب هذه إلى **المستضاف** `config/runtime.exs`. الـ `config/runtime.exs` الملتزم به في المستودع هو عمداً نموذج يستخدم فقط من قبل مجموعة الاختبار الآلي. لا تخلط بينه وبين تكوين الإنتاج.

تُقرأ جميع المفاتيح أدناه عبر `Application.get_env(:omniss7, ...)` (أو التطبيق ذي في وقت التشغيل، لذا فإن القيم الافتراضية المعروضة هي القيم المستخدمة عند حذف (الصلة مفتاح.

علامات وضع التشغيل

تحكم في الميزات المفعلة.

الأوضاع	الوصف	الافتراضي	النوع	المعلمة
جميعها	تمكين عميل MAP واتصال M3UA	`false`	Boolean	`map_client_enabled`
HLR	تمكين الميزات الخاصة بـ HLR	`false`	Boolean	`hlr_mode_enabled`
SMSc	تمكين الميزات الخاصة بـ SMSc	`false`	Boolean	`smsc_mode_enabled`
CAMEL GW	تمكين عميل CAP لعمليات CAMEL	`false`	Boolean	`cap_client_enabled`
CAMEL GW	تمكين ميزات بوابة CAMEL	`false`	Boolean	`camelgw_mode_enabled`
USSD GW	تمكين بوابة USSD (جسر HTTP/JSON)	`false`	Boolean	`ussd_gateway_enabled`

مثال:

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false
```

معلمات وضع HLR

تكوين لوضع HLR (سجل الموقع المنزلي).

HLR تكوين واجهة برمجة تطبيقات

المعلمة	النوع	الافتراضي	مطلوب	
`hlr_api_base_url`	String	-	**نعم**	وان ات HL
`hlr_api_key`	String	-	**نعم**	جة ات سله س ke On الة. تاح عينه دور hl رغًا، لب On hl
`hlr_api_status_path`	String	`"/api/status"`	لا	إلى hl حالة بط **بط** كم /h
`hlr_service_center_gt_address`	String	-	**نعم**	وان HL Up

المعلمة	النوع	الافتراضي	مطلوب	
`smsc_service_center_gt_address`	String	-	**نعم**	وان دود for

> **التحقق من SSL:** لا يوجد مفتاح `hlr_api_verify_ssl`. لا يتم قراءته في أي مكان في لاستدعاءات واجهة برمجة التطبيقات الخلفية TLS الشيفرة. يتم تنظيم التحقق من شهادة بواسطة حقل `verify_ssl` في خريطة `:api` العامة (انظر التحقق من SSL لواجهة برمجة التطبيقات الخلفية).

مثال:

```
config :omniss7,
  hlr_api_base_url: "https://omnihss.example.com:8443",
  hlr_api_key: "<مفتاح واجهة برمجة تطبيقات OmniHSS مع دور
hlr_interworking>",
  hlr_api_status_path: "/api/status",
  hlr_service_center_gt_address: "55512341111",
  smsc_service_center_gt_address: "55512341112"
```

تكوين AlertServiceCenter

عندما يقوم المشترك بتنفيذ UpdateLocation، يرسل HLR رسائل alertServiceCenter إلى GTs الخاصة بـ SMSc المكونة للإشارة إلى أن المشترك أصبح الآن قابلًا للوصول.

المعلمة	النوع	الافتراضي	مطلوب	
`hlr_smsc_alert_gts`	قائمة من السلاسل	`[]`	لا	وين ة لـ رات ale
`hlr_alert_location_expiry_seconds`	عدد صحيح	`172800`	لا	وقع ي: على تهاء حية

مثال:

```
config :omniss7,
  hlr_smsc_alert_gts: [
    "15559876543",
    "15559876544"
  ],
  hlr_alert_location_expiry_seconds: 172800  # 48 ساعة
```

ربط MSISDN ↔ IMSI

تكوين لتوليد IMSI الاصطناعية من MSISDNs. للحصول على شرح تقني مفصل لخوارزمية الربط، انظر ربط MSISDN ↔ IMSI في دليل HLR.

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`hlr_imsi_plmn_prefix`	String	`"50557"`	لا	بادئة PLMN (MCC+MNC) لتوليد IMSI الاصطناعية
`hlr_msisdn_country_code`	String	`"61"`	لا	بادئة رمز الدولة للربط العكسي IMSI→MSISDN
`hlr_msisdn_nsn_offset`	عدد صحيح	`0`	لا	الإزاحة في MSISDN حيث يبدأ NSN (عادةً طول رمز الدولة)
`hlr_msisdn_nsn_length`	عدد صحيح	`9`	لا	طول رقم المشترك الوطني لاستخراجه من MSISDN

مثال (رمز دولة م��ون من رقمين):

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "99",      # مثال على رمز دولة مكون من رقمين
  hlr_msisdn_nsn_offset: 2,           # تخطي رمز الدولة المكون من رقمين
  hlr_msisdn_nsn_length: 9            # استخراج NSN مكون من 9 أرقام
```

مثال (رمز دولة مكون من 3 أرقام):

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "999",     # مثال على رمز دولة مكون من
3 أرقام
  hlr_msisdn_nsn_offset: 3,           # تخطي رمز الدولة المكون من
3 أرقام
  hlr_msisdn_nsn_length: 8            # استخراج NSN مكون من 8
أرقام
```

مهم: اضبط `nsn_offset` على طول رمز دولتك لاستخراج NSN بشكل صحيح. على سبيل المثال:

- رمز الدولة "9" (رقم واحد) → `nsn_offset: 1`
- رمز الدولة "99" (رقمان) → `nsn_offset: 2`
- رمز الدولة "999" (3 أرقام) → `nsn_offset: 3`

تكوين InsertSubscriberData (ISD)

تكوين لبيانات توفير المشترك المرسلة إلى VLRs أثناء UpdateLocation. للحصول على شرح م��صل لتسلسل ISD وتدفق الرسائل، انظر تكوين InsertSubscriberData في دليل HLR.

المعلمة	النوع	الافتراضي	مطلوب	

`isd_network_access_mode`	Atom	`:packetAndCircuit`	لا	شبكة
				`:pa`
				`:pa`
				`:ci`

`isd_send_ss_data`	Boolean	`true`	لا	رسال ميلية
`isd_send_call_barring`	Boolean	`true`	لا	رسال لمات

مثال:

```
config :omniss7,
  isd_network_access_mode: :packetAndCircuit,
  isd_send_ss_data: true,
  isd_send_call_barring: true
```

تكوين CAMEL

تكوين لتوجيه المكالمات الذكية المعتمد على CAMEL. للحصول على شرح مفصل لتكامل CAMEL في دليل HLR ومفاتيح الخدمة، انظر تكامل.

المعلمة	النوع	الافتراضي	لوب
`camel_service_key`	عدد صحيح	`11_110`	لا
`camel_trigger_detection_point`	Atom	`:termAttemptAuthorized`	لا
`camel_gsmscf_gt_address`	String	(المدعو GT يستخدم)	لا

| `hlr_camel_subscriptions` | لا | `[]` | قائمة | O-CSI / T-CSI لكل بادئة IMSI المرسلة إلى VLR في InsertSubscriberData (انظر اشتراكات CAMEL في ISD) |

مثال:

```
config :omniss7,
  camel_service_key: 11_110,
  camel_trigger_detection_point: :termAttemptAuthorized,
  hlr_camel_subscriptions: [
    %{
      imsi_prefix: "3133800000009",
      o_csi: %{service_key: 100, gsmscf_address: "14155550300",
default_call_handling: :release_call},
      t_csi: %{service_key: 101, gsmscf_address: "14155550300",
default_call_handling: :continue_call}
    }
  ]
```

المنزلية VLR بادئات

تكوين لتمييز المشتركين المحليين مقابل المشتركين المتجولين. للحصول على شرح مفصل انظر HLR معالجة المشتركين المتجولين في دليل، PRN لاكتشاف المنزل/التجوال وعمليات.

المعلمة	ا��نوع	الافتراضي	مطلوب	الوصف
`home_vlr_prefixes`	قائمة	`["5551231"]`	لا	بادئات GT لـ VLR التي تعتبر شبكة منزلية""

مثال:

```
config :omniss7,
  home_vlr_prefixes: ["5551231", "5551234"]
```

معلمات وضع SMSc

تكوين لوضع مركز الرسائل القصيرة.

تكوين واجهة برمجة تطبيقات SMSc

المعلمة	النوع	الافتراضي
`smsc_api_base_url`	String	`"https://omnimessage.example`
`smsc_api_status_path`	String	`"/api/status"`
`smsc_name`	String	`"{hostname}_SMSc"`
`smsc_service_center_gt_address`	String	-

> **التحقق من SSL:** لا يوجد مفتاح `smsc_api_verify_ssl`. لا يتم قراءته في أي مكان لاستدعاءات واجهة برمجة التطبيقات الخلفية TLS في الشيفرة. يتم تنظيم التحقق من شهادة SSL لواجهة برمجة التطبيقات الخلفية بواسطة حقل `verify_ssl` في خريطة `:api` العامة (انظر التحقق من التطبيقات الخلفية).

مثال:

```
config :omniss7,
  smsc_api_base_url: "https://omnimessage.example.com:8443",
  smsc_api_status_path: "/api/status",
  smsc_name: "ipsmgw",
  smsc_service_center_gt_address: "55512341112"
```

ملاحظة: يحدث التسجيل في الواجهة الأمامية كل **5 دقائق** (محدد مسبقًا) عبر وحدة `SMS.FrontendRegistry`.

تكوين التفريغ التلقائي

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`auto_flush_enabled`	Boolean	`true`	لا	تمكين معالجة قائمة انتظار الرسائل القصيرة تلقائيًا
`auto_flush_interval`	عدد صحيح	`10_000`	لا	فترة معالجة قائمة الانتظار بالمللي ثانية
`auto_flush_dest_smsc`	String	-	**نعم**	اسم SMSc الوجهة للتفريغ التلقائي
`auto_flush_tps`	عدد صحيح	`10`	لا	معدل معالجة الرسائل (المعاملات/الثانية)

مثال:

```
config :omniss7,
  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: "ipsmgw",
  auto_flush_tps: 10
```

معلمات وضع STP

تكوين لوضع نقطة نقل إشارة M3UA. للحصول على تكوين توجيه مفصل وأمثلة، انظر دليل تكوين STP.

خادم STP مستقل

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`sctp_handler.enabled`	Boolean	`false`	لا	تمكين خادم SCTP SocketHandle المستقل
`sctp_handler.local_ip`	Tuple أو قائمة	`{127, 0, 0, 1}`	لا	عنوان IP (عناوين) للاستماع للاتصالات. IP واحد: `{10, 0, 0, 1}` أو عناوين IP متعددة لتعدد الإرسال SCTP: `[{10, 0, 0, 1}, {10, 0, 0, 2}]`
`sctp_handler.local_port`	عدد صحيح	`2905`	لا	المنفذ للاستماع عليه
`sctp_handler.point_code`	عدد صحيح	-	**نعم** (إذا تم تمكينه)	رمز النقطة SS7 الخاص بـ STP هذا

مثال (واحد IP):

```
config :omniss7,
  sctp_handler: %{
    enabled: true,
    local_ip: {10, 179, 4, 10},
    local_port: 2905,
    point_code: 100
  }
```

مثال (تعدد الإرسال SCTP):

```
config :omniss7,
  sctp_handler: %{
    enabled: true,
    # عناوين IP متعددة للموثوقية
    local_ip: [{10, 179, 4, 10}, {10, 179, 4, 11}],
    local_port: 2905,
    point_code: 100
  }
```

> **ملاحظة**: للحصول على معلومات مفصلة حول تكوين تعدد الإرسال والفوائد، انظر SCTP تعدد الإرسال في الدليل المشترك SCTP.

توجيه العنوان العالمي

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`enable_gt_routing`	Boolean	`false`	لا	تمكين توجيه GT بالإضافة إلى توجيه PC

مثال:

```
config :omniss7,
  enable_gt_routing: true
```

تكوين أقران M3UA/M2PA

يتم تكوين الأقران عبر قائمة `m3ua_peers` (يدعم كل من بروتوكولات M3UA و M2PA). للحصول على أمثلة تكوين كاملة، انظر دليل تكوين STP و دعم بروتوكول M2PA.

المعلمات الشائعة للأقران:

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`peer_id`	عدد صحيح	-	**نعم**	عرف فريد للند
`name`	String	-	**نعم**	سم وصفي للند
`protocol`	Atom	`:m3ua`	لا	وع البروتوكول `:m3ua` أو `:m`
`role`	Atom	`:client`	لا	دور الاتصال: `:client`، `:server`، `:a` أو `:sgp`
`local_ip`	Tuple أو قائمة	-	**نعم**	وين) IP عنوان المحلية للربط
`local_port`	عدد صحيح	-	**نعم**	المنفذ المحلي SCTP (M3UA 2905، M2PA 3565)
`remote_ip`	Tuple أو قائمة	-	**نعم**	وين) IP عنوان البعيد
`remote_port`	عدد صحيح	-	**نعم**	S المنفذ البعيد
`routing_context`	عدد صحيح	-	لا	ف سياق توجيه M3UA (M3U فقط)
`point_code`	عدد صحيح	-	**نعم**	النقطة المحلي

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`network_indicator`	Atom	`:international`	لا	مؤشر الشبكة: `:internati` أو `:nationa`
`initiate_connection`	Boolean	`true`	لا	ا كان يجب بدء اتصال SCTP

المعلمات الخاصة بـ M2PA:

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`adjacent_point_code`	عدد صحيح	-	**نعم** (M2PA)	رمز النقطة للند المجاور

إدارة المقابس:

لإدارة المقابس المشتركة. تستخدم جميع أقران SCTP.SocketHandler تلقائيًا M2PA تستخدم بكفاءة. SCTP المقبس المشترك، مما يسمح للعديد من الأقران بمشاركة نفس منفذ M2PA للحصول على تكوين مفصل، انظر متطلبات مقبس M2PA.

مثال (ند M3UA):

```
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "HLR_East",
      protocol: :m3ua,
      role: :sgp,
      local_ip: {10, 179, 4, 10},
      local_port: 2905,
      remote_ip: {10, 179, 4, 20},
      remote_port: 2905,
      point_code: 100,
      network_indicator: :international
    }
  ]
```

مثال (ند M2PA):

```
config :omniss7,
  sctp_handler: %{
    enabled: true,
    local_ip: {10, 179, 4, 10},
    local_port: 3565,
    point_code: 100
  },
  m3ua_peers: [
    %{
      peer_id: 2,
      name: "M2PA_Link_STP_West",
      protocol: :m2pa,
      role: :client,
      local_ip: {10, 179, 4, 10},
      local_port: 3565,
      remote_ip: {10, 179, 4, 30},
      remote_port: 3565,
      point_code: 100,
      adjacent_point_code: 200
    }
  ]
```

طرق نقطة كود M3UA

تكوين توجيه نقطة الكود. تحدد الطرق أي ند يجب استخدامه للوصول إلى رموز النقاط الوجهة المحددة.

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`m3ua_routes`	قائمة من الخرائط	`[]`	لا	قائمة بطرق نقطة الكود. إذا لم يتم تحديدها، يتم إنشاء الطرق تلقائيًا من رموز النقاط للأقران.

تنسيق الطريق: يجب أن تكون كل طريقة في `m3ua_routes` خريطة تحتوي على:

- `dest_pc`: (عدد صحيح) رمز النقطة الوجهة
- `peer_id`: (عدد صحيح) معرف الند للتوجيه من خلاله
- `priority`: (عدد صحيح) أولوية الطريق - القيمة الأقل = أولوية أعلى
- `network_indicator`: مؤشر الشبكة (Atom): `:international` أو `:national`

مثال:

```
config :omniss7,
  m3ua_routes: [
    # طريق إلى PC 100 (أعلى أولوية) 1 عبر الند
    %{dest_pc: 100, peer_id: 1, priority: 1, network_indicator:
:international},
    # طريق إلى PC 200 2 عبر الند
    %{dest_pc: 200, peer_id: 2, priority: 1, network_indicator:
:international},
    # نفس dest_pc مع أولويات مختلفة: توازن التحميل
    %{dest_pc: 300, peer_id: 3, priority: 1, network_indicator:
:international},
    %{dest_pc: 300, peer_id: 4, priority: 2, network_indicator:
:international}
  ]
```

طرق العنوان العالمي M3UA

توجيه قائم على بادئة العنوان العالمي مع تحويلات متقدمة لبارامترات SCCP. يتم استخدام أطول تطابق بادئة أولاً، ثم الأولوية.

الوصف	مطلوب	الافتراضي	النوع	المعلمة
قائمة بقواعد توجيه العنوان العالمي مع تحويلات اختيارية	لا	`[]`	قائمة من الخرائط	`m3ua_gt_routes`

تنسيق الطريق: يجب أن تكون كل طريقة في `m3ua_gt_routes` خريطة تحتوي على:

المعلمات الأساسية:

- `gt_prefix`: بادئة العنوان العالمي للمطابقة (String) - سلسلة فارغة تطابق الجميع
- `peer_id`: معرف الند للتوجيه من خلاله (عدد صحيح) - استخدم 0 لإسقاط الحركة
- `priority`: أولوية الطريق (عدد صحيح) - القيمة الأقل = أولوية أعلى
- `description`: وصف قابل للقراءة البشرية (String)

المعلمات الاختيارية للمطابقة:

- `source_ssn`: مطابقة رقم النظام الفرعي المصدر (عدد صحيح)
- `source_tt`: مطابقة نوع الترجمة المصدر (عدد صحيح)
- `source_npi`: مطابقة مؤشر خطة الترقيم المصدر (عدد صحيح)
- `source_nai`: مطابقة مؤشر طبيعة العنوان المصدر (عدد صحيح)

المعلمات الاختيارية للتحويل:

- `dest_ssn`: تحويل SSN في الرسالة المرسلة (عدد صحيح)
- `dest_tt`: تحويل نوع الترجمة في الرسالة المرسلة (عدد صحيح)
- `dest_npi`: تحويل مؤشر خطة الترقيم في الرسالة المرسلة (عدد صحيح)
- `dest_nai`: تحويل مؤشر طبيعة العنوان في الرسالة المرسلة (عدد صحيح)
- `route_on_ssn`: قلب مؤشر توجيه الطرف المدعو للتوجيه بناءً على SSN (Boolean، الافتراضي `false`). اترك `false` لتوجيه GT العابر حتى يقوم العقدة السفلية بأداء GTT (ITU-T Q.714 §2.4)؛ اضبط `true` عندما يكون هذا **فقط** STP

فقط. انظر SSN هو المترجم النهائي الذي يقوم بالتسليم إلى نقطة نهاية تعتمد على دليل STP → التوجيه بناءً على SSN بعد GTT.

القيم الشائعة:
- **نوع الترجمة (TT):** 0=غير معروف، 1=دولي، 2=وطني، 3=خاص بالشبكة
- **خطة الترقيم (NPI):** 0=غير معروف، 1=ISDN(E.164)، 6=محمول(E.212)
- **طبيعة العنوان (NAI):** 0=غير معروف، 1=مشترك، 3=وطني، 4=دولي
- **رقم النظام الفرعي (SSN):** 6=HLR، 7=VLR، 8=MSC، 9=EIR، إلخ.

مثال:
```elixir
config :omniss7,
  m3ua_gt_routes: [
    # توجيه بادئة أساسية
    %&#123;gt_prefix: "1234", peer_id: 1, priority: 1, description: "125#&"أرقام الولايات المتحدة;,
    %&#123;gt_prefix: "44", peer_id: 2, priority: 1, description: "125#&"أرقام المملكة المتحدة;,

    # تحويل نوع الترجمة
    %&#123;
      gt_prefix: "61",
      peer_id: 3,
      priority: 1,
      description: "تحويل :أرقام أسترالية TT 0→1",
      source_tt: 0,  # مطابقة TT=0 (غير معروف)
      dest_tt: 1     # تحويل إلى TT=1 (دولي)
    &#125;,

    # تحويل NPI
    %&#123;
      gt_prefix: "49",
      peer_id: 1,
      priority: 1,
      description: "تحويل :أرقام ألمانية NPI المحمول→ISDN",
      source_npi: 6,  # مطابقة NPI=6 (محمول/E.212)
      dest_npi: 1     # تحويل إلى NPI=1 (ISDN/E.164)
    &#125;,

    # تحويل مشترك مع توجيه SSN
    %&#123;
      gt_prefix: "86",
      source_ssn: 8,   # مطابقة SSN=8 (MSC)
      peer_id: 3,
      dest_ssn: 6,     # إعادة كتابة إلى SSN=6 (HLR)
```

```
      priority: 1,
      description: "حركة المرور الصينية: تطبيع كامل",
      source_tt: 0,
      dest_tt: 2,
      source_npi: 6,
      dest_npi: 1,
      source_nai: 4,
      dest_nai: 3
    &amp;#125;,

    # مسار افتراضي/احتياطي
    %&amp;#123;
      gt_prefix: "",
      peer_id: 1,
      priority: 99,
      description: "مسار احتياطي افتراضي"
    &amp;#125;
  ]
```

معلمات وضع بوابة CAMEL

تكوين وضع بوابة CAMEL (بروتوكول CAP).

علامات وضع CAMEL

تمكين ميزات CAMEL/CAP (قم بتعيين `cap_client_enabled: true` و `camelgw_mode_enabled: true` في علامات الوضع التشغيلي).

تكوين بروتوكول CAP

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`cap_version`	Atom	`:v2`	لا	إصدار بروتوكول CAP: `:v1`، `:v2`، `:v3`، أو `:v4`
`camel_gsmscf_gt_address`	String	(يستخدم GT المدعو)	لا	عنوان العنوان العالمي الافتراضي لـ gsmSCF لاستجابات CAMEL

تعيين إصدار CAP:

- `:v1` → معرف سياق التطبيق OID: 0.4.0.0.1.0.50.0
- `:v2` → معرف سياق التطبيق OID: 0.4.0.0.1.0.50.1 (افتراضي - الأكثر دعمًا)
- `:v3` → معرف سياق التطبيق OID: 0.4.0.0.1.21.3.4
- `:v4` → معرف سياق التطبيق OID: 0.4.0.0.1.23.3.4

ملاحظة: يتم اكتشاف الطلبات الواردة تلقائيًا من معرف سياق التطبيق الخاص بها وتطابق الاستجابات إصدار الطلب.

مثال:

```
config :omniss7,
  cap_client_enabled: true,
  camelgw_mode_enabled: true,
  cap_version: :v2,
  camel_gsmscf_gt_address: "99988411553"
```

تكامل CGrateS

تكامل الشحن في الوقت الفعلي مع CGrateS للفوترة المدفوعة مسبقًا/المتأخرة.

المعلمة	النوع	الافتراضي	مطلوب	لوصف
`cgrates_enabled`	Boolean	`false`	لا	تمكين تكامل CGrateS
`cgrates_url`	String	-	**نعم** (إذا تم تمكينه)	ظة URL عنوان نهاية JSON-RP CGrateS
`cgrates_tenant`	String	`"cgrates.org"`	لا	معرف مستأجر CGrateS
`cgrates_request_type`	String	`"*prepaid"`	لا	نوع الشحن: `"*prepaid"` `"*postpaid` `"*pseudopr`
`cgrates_timeout`	Integer	`5000`	لا	CG مهلة طلب بالمللي ثانية

مثال:

```
config :omniss7,
  cgrates_enabled: true,
  cgrates_url: "http://localhost:2080/jsonrpc",
  cgrates_tenant: "cgrates.org",
  cgrates_request_type: "*prepaid",
  cgrates_timeout: 5000
```

اتصال CAP M3UA

تستخدم بوابة CAMEL اتصال M3UA منفصل لعمليات CAP.

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`cap_client_m3ua`	Map	-	**نعم**	تكوين اتصال CAP M3UA (نفس هيكل `map_client_m3ua`، بما في ذلك المعلمات الاختيارية `opc` و `dpc`)

مثال:

```
config :omniss7,
  cap_client_m3ua: %&#123;
    mode: "ASP",
    callback: &#123;CapClient, :handle_payload, []&#125;,
    process_name: :camelgw_client_asp,
    local_ip: &#123;10, 5, 198, 200&#125;,
    local_port: 2905,
    remote_ip: &#123;10, 179, 4, 10&#125;,
    remote_port: 2905,
    routing_context: 4,
    opc: 5013,    # رمز النقطة الأصلية
    dpc: 5011     # رمز النقطة الوجهة
  &#125;
```

معلمات NAT للعنوان العالمي

استجابة مختلفة بناءً على بادئة الطرف المتصل، بادئة GTs للعنوان العالمي بوجود NAT يسمح الطرف المدعو، أو كليهما. يتم مطابقة القواعد حسب الوزن (الأقل = أولوية أعلى)، ثم خصوصية البادئة (الطول المركب الأطول = أكثر تحديدًا). لمزيد من الشرح والأمثلة، انظر دليل NAT للعنوان العالمي.

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`gt_nat_enabled`	Boolean	`false`	لا	تمكين/تعطيل ميزة NAT للعنوان العالمي
`gt_nat_rules`	List of Maps	`[]`	**نعم** (إذا تم تمكينه)	قائمة بقواعد NAT للعنوان العالمي مع مطابقة البادئة

خريطة تحتوي على `gt_nat_rules` **تنسيق القاعدة:** يجب أن تكون كل قاعدة في:

- `calling_prefix`: بادئة سلسلة لمطابقتها مع GT (اختياري) المتصل
- `called_prefix`: بادئة سلسلة لمطابقتها مع GT (اختياري) المدعو
- `weight`: 100 :الافتراضي - (الأقل = أولوية أعلى) أولوية عددية
- `response_gt`: العنوان العالمي الذي سيتم استخدامه في الاستجابات (مطلوب)

أولوية المطابقة:

1. يتم مطابقة القواعد حسب `weight` (القيمة الأقل = أولوية أعلى)
2. إذا كانت الأوزان متساوية، فإن طول البادئة المركبة الأطول يفوز
3. يمكن استخدام كل من `called_prefix` و `calling_prefix` معًا للمطابقة الدقيقة

مثال:

```
config :omniss7,
  gt_nat_enabled: true,
  gt_nat_rules: [
    # أولوية عالية: مطابقة كل من المتصل "8772" والمدعو إلى
555""
    %&#123;calling_prefix: "8772", called_prefix: "555", weight:
1, response_gt: "111111"&#125;,

    # أولوية متوسطة: مطابقة فقط المتصل من "8772
    %&#123;calling_prefix: "8772", weight: 10, response_gt:
"99988411553"&#125;,

    # أولوية متوسطة: مطابقة فقط المدعو إلى "555
    %&#123;called_prefix: "555", weight: 10, response_gt:
"99988411554"&#125;,

    # مطابقة فقط المتصل من "8773
    %&#123;calling_prefix: "8773", weight: 10, response_gt:
"99988411554"&#125;,

    # قاعدة احتياطية عامة (تطابق كل شيء، أعلى وزن)
    %&#123;weight: 100, response_gt: "99988411555"&#125;
  ]
```

انظر أيضًا: دليل NAT للاستخدام التفصيلي والأمثلة للعنوان العالمي.

معلمات اتصال M3UA

تكوين اتصال M3UA لوضع عميل MAP. لمزيد من الاستخدام التفصيلي والأمثلة، انظر دليل عميل MAP.

المعلمة	النوع	الافتراضي	مطلوب
`map_client_m3ua.mode`	String	-	**نعم**
`map_client_m3ua.callback`	Tuple	-	**نعم**
`map_client_m3ua.process_name`	Atom	-	**نعم**
`map_client_m3ua.local_ip`	Tuple or List	-	**نعم**
`map_client_m3ua.local_port`	Integer	`2905`	**نعم**
`map_client_m3ua.remote_ip`	Tuple or List	-	**نعم**
`map_client_m3ua.remote_port`	Integer	`2905`	**نعم**
`map_client_m3ua.routing_context`	Integer	-	**نعم**
`map_client_m3ua.opc`	Integer	`5013`	لا
`map_client_m3ua.dpc`	Integer	`5011`	لا

المعلمة	النوع	الافتراضي	مطلوب
`map_client_m3ua.receive_watchdog`	Boolean	`true`	لا
`map_client_m3ua.receive_watchdog_idle`	Integer	`15`	لا

مثال (IP فردي):

```
config :omniss7,
  map_client_m3ua: %&#123;
    mode: "ASP",
    callback: &#123;MapClient, :handle_payload, []&#125;,
    process_name: :hlr_client_asp,
    local_ip: &#123;10, 179, 4, 11&#125;,
    local_port: 2905,
    remote_ip: &#123;10, 179, 4, 10&#125;,
    remote_port: 2905,
    routing_context: 1,
    opc: 5013,    # (5-114-2) رمز النقطة الأصلية
    dpc: 5011     # (3-114-2) رمز النقطة الوجهة
  &#125;
```

> إلى أعداد صحيحة `X-Y-Z` **تنسيق رمز النقطة**: يتم تحويل رموز النقطة في تنسيق كالتالي: `(X * 2048) + (Y * 8) + Z`. على سبيل المثال، `5-114-2` = `(2048 * 2)` + `5 + (8 * 114) +` = `5013`.

مثال (SCTP Multihoming):

```
config :omniss7,
  map_client_m3ua: %&#123;
    mode: "ASP",
    callback: &#123;MapClient, :handle_payload, []&#125;,
    process_name: :hlr_client_asp,
    # عدة IPs محلية للحد من الفشل
    local_ip: [&#123;10, 179, 4, 11&#125;, &#123;10, 179, 4,
12&#125;],
    local_port: 2905,
    # عدة IPs بعيدة للحد من الفشل في STP
    remote_ip: [&#123;10, 179, 4, 10&#125;, &#123;10, 179, 4,
20&#125;],
    remote_port: 2905,
    routing_context: 1
  &#125;
```

> **ملاحظة**: لمزيد من المعلومات التفصيلية حول تكوين SCTP multihoming والفوائد، انظر SCTP Multihoming في الدليل الشائع.

مراقب الاستقبال

يقوم SCTP **مراقب الاستقبال** بمراقبة اتصالات بحثًا عن المقابس الميتة. المقبس الميت هو ارتباط يبقى في حالة مُثبتة على مستوى نظام التشغيل، ولكن الطرف البعيد قد توقف عن إرسال البيانات بشكل صامت. بدون المراقب، قد لا يكتشف العقدة اتصالًا ميتًا حتى تفشل محاولة الإرسال التالية.

تم تلقيها تعيد تعيين مؤقت الخمول، بما في ذلك استجابات SCTP أي حمولة M3UA BEAT، اختياري. يقوم M3UA BEAT، فإن §3.8 RFC 4666، وبيانات التطبيق. وفقًا لـ NOTIFY، ورسائل SCTP يرسل ا�ً بالفعل بأداء نبضات القلب الإلزامية على مستوى طبقة النقل. عندما SG البعيد قم بتعطيل المراقب، (ولا يرسل M3UA BEATs) حركة مرور دورية على مستوى طبقة التطبيق لتجنب دورات إعادة الاتصال غير الضرورية.

المعلمة	النوع	الافتراضي	الوصف
`receive_watchdog`	Boolean	`true`	تمكين أو تعطيل مراقب الاستقبال. عندما يكون `false`، لا يتم إنهاء الاتصالات الخاملة بواسطة المراقب؛ لا يزال نبض القلب على مستوى طبقة النقل SCTP يعمل بشكل طبيعي.
`receive_watchdog_idle`	Integer	`15`	ثواني من عدم النشاط قبل أن يقوم المراقب بإنهاء الاتصال وبدء إعادة الاتصال. ليس له تأثير عندما يكون `receive_watchdog: false`.

مثال: تعطيل المراقب لعميل ASP:

```
config :omniss7,
  map_client_m3ua: %&#123;
    mode: "ASP",
    local_ip: &#123;10, 179, 4, 11&#125;,
    local_port: 2905,
    remote_ip: &#123;10, 179, 4, 10&#125;,
    remote_port: 2905,
    routing_context: 1,
    receive_watchdog: false        # تعطيل - SG البعيد لا يرسل حركة مرور دورية
  &#125;
```

مثال: تخصيص عتبة الخمول:

```
config :omniss7,
  map_client_m3ua: %&#123;
    mode: "ASP",
    local_ip: &#123;10, 179, 4, 11&#125;,
    local_port: 2905,
    remote_ip: &#123;10, 179, 4, 10&#125;,
    remote_port: 2905,
    routing_context: 1,
    receive_watchdog: true,
    receive_watchdog_idle: 30      # إنهاء بعد 30 ثانية من الصمت
(الافتراضي: 15 ثانية)
  &#125;
```

حالة النظام الفرعي SCCP

تجيب OmniSS7 على اختبارات حالة النظام الفرعي (SST) لإدارة SCCP من الحالة الحقيقية للنظام الفرعي المحلي الذي تم الاستعلام عنه (ITU-T Q.714 §5.3.4). المفتاح `local_subsystems` يستضيفها هذا العقد فعليًا. يتم يعلن عن أي أرقام أنظمة فرعية (SSNs) آخر يتم الرد عليه SSN ؛ أي(SSA) `Subsystem-Allowed` المدرج بـ SSN لرقم SST الرد على بـ `Subsystem-Prohibited` (SSP).

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`local_subsystems`	List of Integers or `:all`	`:all`	لا	SSNs المحلية في الخدمة المستخدمة للرد على SST. `:all` يعامل كل (الافتراضي) نظام فرعي على أنه متاح. يناسب جه��ز إعادة إرسال نقي لا يستضيف ��ي أنظمة فرعية. يجب على المشغلين الذين ينتهون من SSNs محددة إدراجها حتى تعكس SST الواقع.

SSNs الشائعة: 6 = HLR، 7 = VLR، 8 = MSC، 9 = EIR، 146 = gsmSCF/CAMEL.

مثال:

```
config :omniss7,
  local_subsystems: [6, 7, 8, 146]
```

تصفية البوابة

تسمح تصفية البوابة (جدار الحماية SCCP) بحركة المرور العابرة أو تمنعها قبل توجيهها. تتطابق للطرف المدعو/المتصل و GTA مؤشر الخدمة، بادئة، OPC، DPC، القواعد على نظير المصدر عندما لا تتطابق أي `allow`/`deny`؛ القاعدة الأولى المفعلة المطابقة (الأقل أولوية) تقرر SSN قاعدة، يتم تطبيق الإجراء الافتراضي أدناه. يتم إسقاط الرسائل المرفوضة ويتم احتسابها كـ `m3ua_stp_routing_failures_total{reason="screened"}`. انظر دليل STP للنموذج الكامل.

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`screening_default_action`	`:allow` أو `:deny`	`:allow`	لا	الإجراء المطبق على رسالة عبور عندما لا تتطابق أي قاعدة تصفية. `:allow` (الافتراضي) هو موقف افتراضي للقبول؛ `:deny` هو افتراضي للرفض (قائمة السماح). يجعل تعيين `:deny` التصفية نشطة حتى مع عدم تكوين أي قواعد.

مثال:

```
config :omniss7,
  # رفض افتراضي: يتم توجيه حركة المرور المطابقة فقط لقواعد السماح
الصريحة.
  screening_default_action: :deny
```

وتدار برمجيًا؛ يتم تعيين الإجراء `ss7_screen_rule` تحتفظ القواعد نفسها في جدول Mnesia الافتراضي فقط عبر التكوين.

معلمات بوابة USSD

بوابة USSD تربط بين USSD و USSDالتي تبدأ من الشبكة إلى USSD التي تبدأ من الهاتف المحمول. واجهة خلفية HTTP/JSON. قم بتمكينها باستخدام `ussd_gateway_enabled: true` (انظر علامات الوضع التشغيلي)، ثم قم بتكوين خريطة `ussd_gateway`. لمزيد من البروتوكول الكامل للرد والعملية الدورية، انظر دليل بوابة USSD.

المعلمة	النوع	الافتراضي	مطلوب	سف
`ussd_gateway.routes`	List of Maps	`[]`	**نعم** (إذا تم تمكينه)	د توجيه short → URI خال هو `%{patt` `"` `<pref` `url:` `<call` `URL>"` **بادئة طابقة تفوز**؛ ستخدم قصيرة ط طراز `"*"` نياطات عامة.
`ussd_gateway.session_timeout_ms`	Integer	`180_000`	لا	لأقصى لجلسة إجمالي ي ثانية قائق). م إنهاء ة إذا تم جاوزها
`ussd_gateway.turn_timeout_ms`	Integer	`30_000`	لا	لأقصى (مللي لانتظار شترك

المعلمة	النوع	الافتراضي	مطلوب	سف
				ن حوار الأدوار
`ussd_gateway.http_timeout_ms`	Integer	`5_000`	لا	طلب/ اتصال ي ثانية) ود على طبيقك
`ussd_gateway.max_text_length`	Integer	`182`	لا	لأقصى لأحرف سلسلة نص US (GSM bit). م تقليم نصوص ل ويتم نسجيل تحذير.

التوجيه: عندما يصل رمز USSD، يقوم البوابة بفرز الطرق المكونة حسب طول `pattern` تنازليًا. ثم تختار البوا��ة أول طريق يكون له بادئة من الرمز المكون. وهذا يعطي مطابقة محددة لأطول بادئة.

Lexical error on line 2. Unrecognized text. ...5*100* ثل #"] --> B&fl° °123¶ßمطابقة الطرق ----------^

المحاولة مجددا

مثال:

```
config :omniss7,
  ussd_gateway_enabled: true,
  ussd_gateway: %&#123;
    routes: [
      %&#123;pattern: "*100", url: "http://balance-
app:9000/ussd"&#125;,
      %&#123;pattern: "*200", url: "http://topup-
app:9000/ussd"&#125;,
      %&#123;pattern: "*",    url: "http://default-
app:9000/ussd"&#125;
    ],
    session_timeout_ms: 180_000,
    turn_timeout_ms: 30_000,
    http_timeout_ms: 5_000,
    max_text_length: 182
  &#125;
```

علامات التشغيل / الإشراف

تتحكم هذه العلامات في مقدار شجرة الإشراف التي تبدأ. إنها مفيدة لتشغيل OmniSS7 بدون واجهة (كاعتماد مكتبة) أو لتقليص الطبقة الويب في عمليات النشر المقيدة.

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`start_web_interfaces`	Boolean	`true`	لا	واجهة التحكم، خادم REST `:api_ex`، ي ا يكون .SS7 أحداث ط مسجل، `false` الحزم (`SS7.Event` تجاوزها عند التمهيد بواسطة متغير البيئة `ENABLE_WEB_INT` (`true`/`1` أو `fals`
`start_http_server`	Boolean	`true`	لا	بدء خادم HTTP نفذ Plug.Cowboy `api_port` (REST `/api/*`، `/metri` `/swagger`). خادم REST.
`api_port`	Integer	`9090`	لا	منفذ TCP لخادم HT Plug.Cowboy (RI `/api/*`، `/metri` `/swagger`). حديث هدف سحب Prome إذا قمت بتغييره.
`headless`	Boolean	`false`	لا	سيط فقط للمكتبة. `true`، عندما يكون مشرفي OmniSS7 ت) الترميز/المعاملة MAP، CAP، USSD شرفين الديناميكيين ب، لا عميل ترخيص، ادم SCTP، لا معالج ص لتضمين M3UA. ترميزات OmniSS7

المعلمة	النوع	الافتراضي	مطلوب	الوصف
`vlr_cache_ttl_seconds`	Integer	`3600`	لا	ة (بالثواني) لعمليات VLR البحث المؤقتة في SMSc مسار تسليم

مثال (وضع مكتبة بدون واجهة):

```
config :omniss7,
  headless: true
```

مثال (بدون طبقة ويب، API REST فقط):

```
config :omniss7,
  start_web_interfaces: false,
  start_http_server: true
```

معلمات البنية التحتية

تكوين لمكونات بنية النظام بما في ذلك الترخيص، واجهة الويب، خادم API، والتسجيل.

تكوين الترخيص

المعلمة	النوع	الافتراضي	مطلوب
`license_client.license_server_api_urls`	List of Strings	-	**نعم**
`license_client.licensee`	String	-	**نعم**

مثال:

```
config :license_client,
  license_server_api_urls: ["https://localhost:10443/api"],
  licensee: "Omnitouch Network Services Pty. Ltd."
```

واجهة التحكم في لوحة التحكم

المعلمة	النوع	ي
`control_panel.parent_application_readable_name`	String	`"OmniSS7 S`
`control_panel.use_additional_pages`	List of Tuples	`[]`
`control_panel.page_order`	List of Strings	`[]`
`ControlPanelWeb.Endpoint.url.host`	String	`"0.0.0.0"`
`ControlPanelWeb.Endpoint.https.port`	Integer	`8087`
`ControlPanelWeb.Endpoint.https.keyfile`	String	`"priv/cert/`

المعلمة	النوع	ي
`ControlPanelWeb.Endpoint.https.certfile`	String	`"priv/cert/`

مثال:

```
config :control_panel, ControlPanelWeb.Endpoint,
  url: [host: "0.0.0.0", path: "/"],
  https: [
    port: 8087,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ],
  parent_application_readable_name: "OmniSS7 Stack STP"

config :control_panel,
  use_additional_pages: [
    &amp;#123;SS7.Web.EventsLive, "/events", "أحداث SS7"&amp;#125;,
    &amp;#123;SS7.Web.M3UAStatusLive, "/m3ua", "125#&amp;"الأقران;
  ],
  page_order: ["/events", "/m3ua", "/application",
"/configuration"]
```

REST خادم واجهة برمجة التطبيقات

المعامل	النوع	الافتراضي	لوب
`start_http_server`	Boolean	`true`	لا
`api_ex.api.port`	Integer	`8445`	لا
`api_ex.api.listen_ip`	String	`"0.0.0.0"`	لا
`api_ex.api.product_name`	String	`"OmniSS7"`	لا
`api_ex.api.title`	String	`"API - OmniSS7"`	لا

المعامل	النوع	الافتراضي	لوب
`api_ex.api.hostname`	String	`"localhost"`	لا
`api_ex.api.enable_tls`	Boolean	`true`	لا
`api_ex.api.tls_cert_path`	String	`"priv/cert/omnitouch.crt"`	لا
`api_ex.api.tls_key_path`	String	`"priv/cert/omnitouch.pem"`	لا

مثال:

```
config :omniss7,
  start_http_server: true

config :api_ex,
  api: %&#123;
    port: 8445,
    listen_ip: "0.0.0.0",
    product_name: "OmniSS7",
    title: "API - OmniSS7",
    hostname: "localhost",
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem"
  &#125;
```

المنافذ ووسائل النقل في لمحة:

الغرض	الخادم	وسيلة النقل	المنفذ
MAP/CAP REST `/ap` `/metrics`، `/swagg` `/swagger.json`. تم toggled بواسطة `start_http_serve`	`APIhandler`	HTTP (Plug.Cowboy)	9090
التحكم في لوحة الويب. جزء من طبقة الويب (`start_web_inter`	`ControlPanelWeb.Endpoint`	HTTPS (Bandit/Phoenix)	8087
التوجيه، CRUD، الحالة بط، المشتركين، عرض CAMEL/SCTP/M3UA من طبقة الويب.	REST وحدات التحكم `:api_ex`	HTTPS / TLS	8445

نقاط نهاية واجهة برمجة التطبيقات:

- واجهة برمجة التطبيقات MAP/CAP + Swagger: `http://[server-ip]:9090/...` (انظر دليل واجهة برمجة التطبيقات)

- وحدات التحكم REST `:api_ex`: `https://[server-ip]:8445/api/*`
- لوحة التحكم على الويب: `https://[server-ip]:8087/`
- مقاييس Prometheus: `http://[server-ip]:9090/metrics`

التحقق من SSL في الخلفية لواجهة برمجة التطبيقات

في الخلفية مفاتيح HLR/SMSc لا تستخدم المكالمات الصادرة إلى واجهات برمجة التطبيقات `*_verify_ssl` لكل هدف. يتم التحكم في التحقق من الشهادة بواسطة الخريطة العامة `:api`:

المعامل	النوع	الافتراضي	الوصف
`api.base_url`	String	يعود إلى `smsc_api_base_url`	عنوان URL الأساسي لعميل واجهة برمجة التطبيقات العامة.
`api.timeout`	Integer	`5000`	مهلة الطلب بالمللي ثانية.
`api.verify_ssl`	Boolean	`false`	ما إذا كان يجب التحقق من شهادات TLS في الخلفية. **هذا هو مفتاح التحقق من SSL الوحيد.** لا يوجد `hlr_api_verify_ssl` أو `smsc_api_verify_ssl`.

مثال:

```
config :omniss7,
  api: %&#123;
    base_url: "https://backend.example.com:8443",
    timeout: 5000,
    verify_ssl: false
  &#125;
```

تكوين السجلات

المعامل	النوع	الافتراضي	مطلوب
`logger.level`	Atom	`:debug`	لا
`logger.backends`	List	`[:console]`	لا
`logger.default_formatter.format`	String	-	لا
`logger.default_formatter.metadata`	List	`[]`	لا
`logger.default_formatter.truncate`	Atom/Integer	`:infinity`	لا

مثال:

```
config :logger,
  level: :debug,
  backends: [:console, SS7.Web.LoggerBackend]

config :logger, :default_formatter,
  format: "[$date] [$time] [$level] $message\n",
  metadata: [:error_code, :file],
  truncate: :infinity
```

معلمات قاعدة البيانات

تكوين لاستمرارية قاعدة بيانات Mnesia.

المعامل	النوع	الافتراضي	مطلوب	الوصف
`mnesia_storage_type`	Atom	`:disc_copies`	لا	نوع تخزين Mnesia: `:disc_copies` أو `:ram_copies`

مثال:

```
config :omniss7,
  mnesia_storage_type: :disc_copies  # الإنتاج
  # mnesia_storage_type: :ram_copies  # للاختبار فقط
```

أنواع التخزين:

- `:disc_copies` - **موصى به للإنتاج** - تخزين قرص دائم (يستمر بعد إعادة التشغيل)
- `:ram_copies` - للاختبار فقط - فقط في الذاكرة (يفقد عند إعادة التشغيل)

جداول Mnesia:

- `m3ua_peer` - اتصالات نظير M3UA
- `m3ua_route` - طرق رمز النقطة
- `m3ua_gt_route` - طرق العنوان العالمي

الموقع: دليل `Mnesia.{node_name}/`

القيم الثابتة

القيم التالية **مثبتة في الشيفرة المصدرية** ولا يمكن تغييرها عبر التكوين.

المهلات

الحل	التأثير	القيمة
تعديل الشيفرة المصدرية	تنتهي بعد 10 MAP جميع عمليات ثواني	مهلة طلب MAP: **10 ثواني**
تعديل الشيفرة المصدرية	تنتهي بعد 10 ثواني ISD كل رسالة	مهلة ISD: **10 ثواني**

خادم HTTP

الحل	التأثير	القيمة
تعديل الشيفرة المصدرية	يستمع خادم المقاييس/Swagger على جميع الواجهات	عنوان IP HTTP: **0.0.0.0**
تعيين `config :omniss7, api_port: <port>` (الافتراضي `9090`)	يعمل نقطة نهاية المقاييس/Swagger على المنفذ 9090	منفذ HTTP: **9090**

فترات التسجيل

الحل	التأثير	القيمة
تعديل الشيفرة المصدرية	يسجل مع الخلفية كل 5 SMSc دقائق	تسجيل الواجهة الأمامية: **5 دقائق**

تحديث تلقائي لواجهة الويب

الفاصل الزمني	الصفحة
5 ثواني	إدارة التوجيه
2 ثواني	المشتركين النشطين

أمثلة التكوين

تكوين HLR الحد الأدنى

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false,

  hlr_api_base_url: "https://omnihss.example.com:8443",
  hlr_api_key: "<مفتاح واجهة برمجة التطبيقات OmniHSS مع دور
hlr_interworking>",
  hlr_service_center_gt_address: "55512341111",
  smsc_service_center_gt_address: "55512341112",

  map_client_m3ua: %&amp;#123;
    mode: "ASP",
    callback: &amp;#123;MapClient, :handle_payload, []&amp;#125;,
    process_name: :hlr_client_asp,
    local_ip: &amp;#123;10, 179, 4, 11&amp;#125;,
    local_port: 2905,
    remote_ip: &amp;#123;10, 179, 4, 10&amp;#125;,
    remote_port: 2905,
    routing_context: 1
  &amp;#125;
```

تكوين SMSc الحد الأدنى

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: true,

  smsc_api_base_url: "https://omnimessage.example.com:8443",
  smsc_name: "ipsmgw",
  smsc_service_center_gt_address: "55512341112",

  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: "ipsmgw",
  auto_flush_tps: 10,

  map_client_m3ua: %&#123;
    mode: "ASP",
    callback: &#123;MapClient, :handle_payload, []&#125;,
    process_name: :stp_client_asp,
    local_ip: &#123;10, 179, 4, 12&#125;,
    local_port: 2905,
    remote_ip: &#123;10, 179, 4, 10&#125;,
    remote_port: 2905,
    routing_context: 1
  &#125;
```

مع خادم مستقل STP

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: false,

  enable_gt_routing: true,
  mnesia_storage_type: :disc_copies,

  sctp_handler: %&#123;
    enabled: true,
    local_ip: &#123;10, 179, 4, 10&#125;,
    local_port: 2905,
    point_code: 100
  &#125;,

  map_client_m3ua: %&#123;
    mode: "ASP",
    callback: &#123;MapClient, :handle_payload, []&#125;,
    process_name: :stp_client_asp,
    local_ip: &#123;10, 179, 4, 10&#125;,
    local_port: 2906,
    remote_ip: &#123;10, 179, 4, 11&#125;,
    remote_port: 2905,
    routing_context: 1
  &#125;
```

الملخص

إجمالي معلمات التكوين: 85+

حسب الفئة:

- وضع التشغيل: 6 معلمات
- وضع HLR: 18 معلمة
- وضع SMSc: 8 معلمات

- وضع STP: 5+ معلمات (بالإضافة إلى قوائم m3ua_peers، m3ua_routes، m3ua_gt_routes)
- وضع CAMEL GW: 14 معلمة
- العنوان العالمي: 2 معلمات NAT
- اتصال M3UA: 8 معلمات
- حالة نظام SCCP: 1 معلمة
- بوابة USSD: 5 معلمات
- وقت التشغيل / الإشراف: 4 معلمات
- البنية التحتية (الرخصة، الويب، واجهة برمجة التطبيقات، السجلات): 23 معلمة
- قاعدة البيانات: 1 معلمة

المعلمات المطلوبة حسب الوضع:

وضع HLR:

- `hlr_api_base_url`
- `hlr_api_key`
- `hlr_service_center_gt_address`
- `smsc_service_center_gt_address`
- جميع معلمات `map_client_m3ua.*` (8)

وضع SMSc:

- `smsc_api_base_url`
- `smsc_service_center_gt_address`
- `auto_flush_dest_smsc` (إذا تم تمكين التفريغ التلقائي)
- جميع معلمات `map_client_m3ua.*` (8)

وضع STP:

- `sctp_handler.point_code` (إذا تم تمكين معالج SCTP)
- `sctp_handler.local_ip`
- `sctp_handler.local_port`

وضع CAMEL GW:

- `cgrates_url` (إذا تم تمكين CGrateS)

- جميع معلمات `cap_client_m3ua.*` (8)

البنية التحتية:

- `license_client.license_server_api_urls`
- `license_client.licensee`

الوثائق ذات الصلة

- **دليل HLR** - تكوين خاص بـ HLR
- **دليل SMSc** - تكوين خاص بـ SMSc
- **دليل STP** - تكوين توجيه STP
- **دليل واجهة برمجة التطبيقات** REST - مرجع واجهة برمجة التطبيقات
- **دليل بوابة USSD** - تكوين بوابة USSD وبروتوكول رد الاتصال HTTP
- **دليل واجهة الويب** - وثائق واجهة الويب

دليل ترجمة عنوان الشبكة العالمية (NAT)

نظرة عامة

ترجمة عنوان الشبكة العالمية (GT NAT) هي ميزة تسمح لـ OmniSS7 بالاستجابة بعناوين GT الخاصة بالجهة المتصلة بها، أو مزيج GT الخاصة بالجهة المتصلة، أو بادئة GT مختلفة بناءً على بادئة متعددة والحاجة إلى ضمان أن تستخدم GT من كليهما. هذا أمر ضروري عند العمل مع عناوين GT الصحيح بناءً على الشبكة أو النظير الذي يتصل و/أو GT الاستجابات الذي تم الاتصال به.

ما هو الجديد (GT NAT المحسن)

تم تحسين ميزة GT NAT مع قدرات جديدة قوية:

الميزات الجديدة

1. **مطابقة بادئة الجهة المتصلة**: يمكن الآن أن تتطابق القواعد على `called_prefix` بالإضافة إلى `calling_prefix`
2. **المطابقة المجمعة**: يمكن للقواعد أن تتطابق على كل من بادئات الاتصال والجهة المتصلة بها في وقت واحد
3. **أولوية قائمة على الوزن**: تستخدم القواعد الآن حقل `weight` (أقل = أولوية) بدلاً من مجرد طول البادئة (أعلى)
4. **مطابقة مرنة**: يمكنك الآن إنشاء قواعد مع:
   - بادئة الاتصال فقط
   - بادئة الجهة المتصلة بها فقط
   - كل من بادئات الاتصال والجهة المتصلة بها
   - لا شيء (قاعدة بديلة/عامة)

تنسيق القاعدة الجديدة

الحقول المطلوبة:

- `weight`: أولوية عددية (أقل = أولوية أعلى)
- `response_gt`: GT للاستجابة به

الحقول الاختيارية (يوصى بواحد على الأقل للمطابقة المحددة):

- `calling_prefix`: المطابقة على بادئة GT الخاصة بالجهة المتصلة
- `called_prefix`: المطابقة على بادئة GT الخاصة بالجهة المتصلة بها

مثال:

```
gt_nat_rules: [
  # قاعدة محددة مع كلا البادئتين - أعلى أولوية
  %{calling_prefix: "8772", called_prefix: "555", weight: 1,
response_gt: "111111"},

  # قواعد محددة - أولوية متوسطة
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"},
  %{called_prefix: "555", weight: 10, response_gt: "333333"},

  # بديل عام - أقل أولوية
  %{weight: 100, response_gt: "999999"}
]
```

حالات الاستخدام

التشغيل عبر شبكات متعددة

عندما يكون لديك شبكات نظير متعددة وكل منها يتوقع استجابات من GT محدد:

- **الشبكة A** تتصل بـ GT `111111` وتتوقع استجابات من `111111` الخاص بك
- **الشبكة B** تتصل بـ GT `222222` وتتوقع استجابات من `222222` الخاص بك

يمكن لمثيل واحد من GT NAT، ستحتاج إلى مثيلات منفصلة أو توجيه معقد. مع GT NAT، بدون التعامل مع ذلك بذكاء OmniSS7.

سيناريوهات التجوال

عند العمل كـ HLR أو SMSc مع اتفاقيات تجوال:

- GT `555000` مشتركو **الشبكة المنزلية** يستخدمون
- GT `555001` يستخدم **الشريك التجوال 1**
- GT `555002` يستخدم **الشريك التجوال 2**

يضمن GT NAT أن يتلقى كل شريك استجابات من GT الصحيح الذي تم تكوينه للتوجيه إليه.

الاختبار والترحيل

أثناء ترحيل الشبكة أو الاختبار:

- ترحيل حركة المرور تدريجياً من GT القديم إلى GT الجديد
- الحفاظ على كلا GTs خلال فترة الانتقال
- توجيه الاستجابات بناءً على GT الذي استخدمه المتصل

كيف يعمل

تدفق ترجمة العنوان

1. **طلب وارد**: يتلقى OmniSS7 رسالة SCCP مع:
   - GT `55512341112` الجهة المتصلة: (GT الخاص بك)
   - GT `877234567` الجهة المتصلة: (GT الخاص بهم)
2. **بحث GT NAT**: يتحقق النظام من GT المتصل `877234567` مقابل قواعد البادئة المكونة
3. **مطابقة البادئة**: يجد أطول بادئة مطابقة (على سبيل المثال، `8772` تتطابق مع `877234567`)
4. **اختيار GT للاستجابة**: يستخدم `response_gt` من القاعدة المطابقة (على سبيل المثال، `55512341112`)
5. **إرسال الاستجابة**: تستخدم استجابة SCCP:

- GT `877234567` :الجهة المتصلة (معكوس - GT الخاص بهم)
- GT `55512341112` :الجهة المتصلة (GT الذي تم ترجمته)

أنواع الاستجابة المتأثرة

ينطبق GT NAT على عدة طبقات من مكدس SS7:

طبقة SCCP (جميع الاستجابات)

- عناوين GT للجهة المتصلة/المتصل في جميع رسائل الاستجابة
- تأكيدات ISD (InsertSubscriberData)
- استجابات UpdateLocation
- استجابات الخطأ

طبقة MAP (محددات التشغيل)

- **استجابات SRI-for-SM**: `networkNode-Number` (عنوان GT لـ SMSc)
- **UpdateLocation**: `hlr-Number` في الاستجابات
- **InsertSubscriberData**: GT لـ HLR في رسائل ISD

التكوين

التكوين الأساسي

أضف إلى `config/runtime.exs`:

```
config :omniss7,
  # تفعيل GT NAT
  gt_nat_enabled: true,

  # تعريف قواعد GT NAT
  gt_nat_rules: [
    # القاعدة 1: المكالمات من البادئة "8772" تحصل على استجابة من
55512341112""
    %{calling_prefix: "8772", response_gt: "55512341112"},

    # القاعدة 2: المكالمات من البادئة "8773" تحصل على استجابة من
55512341111""
    %{calling_prefix: "8773", response_gt: "55512341111"},

    # القاعدة الافتراضية (بادئة فارغة تتطابق مع كل شيء)
    %{calling_prefix: "", response_gt: "55512311555"}
  ]
```

معلمات التكوين

للحصول على مرجع تكوين كامل، انظر مع��مات GT NAT في مرجع التكوين.

المعلمة	النوع	مطلوب	الوصف
`gt_nat_enabled`	Boolean	نعم	تفعيل/تعطيل ميزة GT NAT
`gt_nat_rules`	قائمة من الخرائط	نعم (إذا تم التفعيل)	قائمة بقواعد مطابقة البادئة

تنسيق القاعدة

كل قاعدة هي خريطة تحتوي على المفاتيح التالية:

```
%{
  calling_prefix: "8772",      # (اختياري) بادئة للمطابقة مع GT
المتصل
  called_prefix: "555",        # (اختياري) بادئة للمطابقة مع GT
الجهة المتصلة بها
  weight: 10,                  # (مطلوب) قيمة الأولوية (أقل = أولوية أعلى)
  response_gt: "55512341112"   # (مطلوب) GT للاستخدام في الاستجابات
}
```

حقول القاعدة:

- **`calling_prefix`** (اختياري): بادئة سلسلة للمطابقة مع GT المتصل الوارد
  - يتم المطابقة كمطابقة بادئة سلسلة
  - أو `""` السلسلة الفارغة `nil` تعمل كبديل (متصل GT تتطابق مع أي)
  - يمكن حذفها لمطابقة أي متصل GT
- **`called_prefix`** (اختياري): بادئة سلسلة للمطابقة مع GT الجهة المتصلة بها الواردة
  - يتم المطابقة كمطابقة بادئة سلسلة
  - أو `""` السلسلة الفارغة `nil` تعمل كبديل (جهة متصلة GT تتطابق مع أي بها)
  - يمكن حذفها لمطابقة أي جهة متصلة بها GT
- **`weight`** (مطلوب): قيمة أولوية عددية
  - وزن أقل = أولوية أعلى (يتم معالجته أولاً)
  - يجب أن تكون 0 =<
  - تستخدم كمعيار فرز أساسي لقواعد المطابقة
- **`response_gt`** (مطلوب): عنوان GT العالمي للاستخدام في الاستجابات
  - يجب أن يكون سلسلة رقمية صالحة من نوع E.164
  - يجب أن تتطابق مع أحد GTs المكونة لديك

يجب تحديد واحد على الأقل من `calling_prefix` أو `called_prefix` للمطابقة المحددة. يمكن حذف كلاهما لقاعدة بديلة/عامة.

منطق مطابقة القاعدة

يتم تقييم القواعد بواسطة **الوزن أولاً (تصاعديًا)، ثم بواسطة خصوصية البادئة المجمعة**:

خوارزمية المطابقة:

1. تصفية القواعد حيث تتطابق جميع البادئات المحددة
   - إذا تم تعيين `calling_prefix`، يجب أن تتطابق مع GT المتصل
   - إذا تم تعيين `called_prefix`، يجب أن تتطابق مع GT الجهة المتصلة بها
   - إذا تم تعيين كلاهما، يجب أن تتطابق كلاهما
   - إذا لم يتم تعيين أي منهما، تعمل القاعدة كبديل
2. فرز القواعد المطابقة بواسطة:
   - **أساسي**: الوزن (تصاعديًا - القيم الأقل أولاً)
   - **ثانوي**: طول البادئة المجمعة (تنازليًا - الأطول = الأكثر تحديدًا)
3. إرجاع أول قاعدة مطابقة

أمثلة:

```
# أمثلة القواعد
gt_nat_rules: [
  # الوزن 1: أعلى أولوية - تتطابق مع كلا البادئتين
  %{calling_prefix: "8772", called_prefix: "555", weight: 1,
response_gt: "111111"},

  # الوزن 10: أولوية متوسطة - قواعد محددة
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"},  #
الاتصال فقط
  %{called_prefix: "555", weight: 10, response_gt: "333333"},    #
الجهة المتصلة بها فقط

  # الوزن 100: أقل أولوية - بديل عام
  %{weight: 100, response_gt: "444444"}  # تتطابق مع كل شيء
]

# أمثلة المطابقة:
# الاتصال: "877234567"، الجهة المتصلة بها: "555123" -> "111111"
(الوزن 1، كلاهما يتطابق)
# الاتصال: "877234567"، الجهة المتصلة بها: "999999" -> "222222"
(الوزن 10، الاتصال فقط)
# الاتصال: "999999999"، الجهة المتصلة بها: "555123" -> "333333"
(الوزن 10، الجهة المتصلة بها فقط)
# الاتصال: "999999999"، الجهة المتصلة بها: "888888" -> "444444"
(الوزن 100، بديل)
```

أمثلة

المثال 1: شريكان في الشبكة

السيناريو: أنت تدير SMSc مع شريكين في الشبكة. كل منهما يتوقع استجابات من GT مختلف.

```
config :omniss7,
  gt_nat_enabled: true,

  # GT الافتراضي لـــ SMSc (يستخدم عند تعطيل GT NAT أو عدم تطابق أي قاعدة)
  smsc_service_center_gt_address: "5551000",

  # قواعد GT NAT للشركاء
  gt_nat_rules: [
    # الشريك A (بادئة 4412) يتوقع استجابات من GT 5551001
    %{calling_prefix: "4412", weight: 10, response_gt: "5551001"},

    # الشريك B (بادئة 4413) يتوقع استجابات من GT 5551002
    %{calling_prefix: "4413", weight: 10, response_gt: "5551002"},

    # استخدام: الافتراضي GT القياسي لـــ SMSc (بديل عام)
    %{weight: 100, response_gt: "5551000"}
  ]
```

تدفق الحركة:

```
طلب وارد SRI-for-SM 44121234567 من:
  GT 5551001 :الجهة المتصلة بها (GT الخاص بك الذي يستخدمه الشريك A)
  GT 44121234567 :المتصل (GT الخاص بالشريك A)

بحث GT NAT:
  "44121234567" 4412" تتطابق مع بادئة"
  GT 5551001" :للاستجابة المحدد"

استجابة SRI-for-SM 44121234567 إلى:
  GT (معكوس) 44121234567 :الجهة المتصلة بها
  GT 5551001 :المتصل (GT الذي تم ترجمته)
  networkNode-Number: 5551001 (في استجابة MAP)
```

المثال 2: HLR مع GTs إقليمية

السيناريو: HLR وطني مع GTs مختلفة لكل منطقة.

```
config :omniss7,
  gt_nat_enabled: true,
  hlr_service_center_gt_address: "555000",  # GT الافتراضي لـ HLR

  gt_nat_rules: [
    # VLRs (5551 بادئة) المنطقة الشمالية
    %{calling_prefix: "5551", weight: 10, response_gt: "555100"},

    # VLRs (5552 بادئة) المنطقة الجنوبية
    %{calling_prefix: "5552", weight: 10, response_gt: "555200"},

    # VLRs (5553 بادئة) المنطقة الغربية
    %{calling_prefix: "5553", weight: 10, response_gt: "555300"},

    # (بديل عام) الافتراضي للمناطق الأخرى
    %{weight: 100, response_gt: "555000"}
  ]
```

المثال 3: سيناريو الترحيل

الترحيل من **السيناريو**: GT القديم إلى GT الجديد تدريجياً.

```
config :omniss7,
  gt_nat_enabled: true,
  hlr_service_center_gt_address: "123456789",  # GT القديم
(الافتراضي)

  gt_nat_rules: [
    # (قامت بتحديث تكويناتها بالفعل) الشبكات المهاجرة
    %{calling_prefix: "555", weight: 10, response_gt:
"987654321"},  # GT الجديد
    %{calling_prefix: "666", weight: 10, response_gt:
"987654321"},  # GT الجديد

    # الجميع لا يزال يستخدم GT (بديل عام) القديم
    %{weight: 100, response_gt: "123456789"}  # GT القديم
  ]
```

المثال 4: مطابقة بادئة الجهة المتصلة بها (جديد)

السيناريو: لديك عدة GTs لخدمات مختلفة، وتريد الاستجابة بـ GT الصحيح بناءً على GT الذي تم الاتصال به.

```
config :omniss7,
  gt_nat_enabled: true,

  gt_nat_rules: [
    # الرسائل القصيرة الخاص بك GT عندما يتصلون بـ (5551xxx)، استجب بتلك GT
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},

    # الصوت الخاص بك GT عندما يتصلون بـ (5552xxx)، استجب بتلك GT
    %{called_prefix: "5552", weight: 10, response_gt: "555200"},

    # البيانات الخاص بك GT عندما يتصلون بـ (5553xxx)، استجب بتلك GT
    %{called_prefix: "5553", weight: 10, response_gt: "555300"},

    # بديل عام
    %{weight: 100, response_gt: "555000"}
  ]
```

تدفق الحركة:

```
طلب وارد إلى GT 555100 :الجهة المتصلة بها (الرسائل القصيرة GT الخاص بك)
المتصل: 441234567 (أي متصل) GT

بحث GT NAT:
  GT 5551" تتطابق مع بادئة "555100" الجهة المتصلة بها"
  GT 555100" :للاستجابة المحدد"

المتصل: 555100 (تتطابق مع ما اتصلوا به) GT تستخدم الاستجابة
```

المثال 5: مطابقة بادئة الاتصال + الجهة المتصلة بها (متقدم)

السيناريو: يتصل شركاء مختلفون بـ GTs مختلفة، وتريد تحكمًا دقيقًا.

```
config :omniss7,
  gt_nat_enabled: true,

  gt_nat_rules: [
    # الرسائل القصيرة الخاصة بك - أعلى أولوية GT يتصل بـ الشريك A
(الوزن 1)
    %{calling_prefix: "4412", called_prefix: "5551", weight: 1,
response_gt: "555101"},

    # الرسائل القصيرة الخاصة بك - أعلى أولوية GT يتصل بـ الشريك B
(الوزن 1)
    %{calling_prefix: "4413", called_prefix: "5551", weight: 1,
response_gt: "555102"},

    # الرسائل القصيرة الخاصة بك - أولوية متوسطة GT أي شخص يتصل بـ
(الوزن 10)
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},

    # الشريك A يتصل بأي GT - (الوزن 10) أولوية متوسطة
    %{calling_prefix: "4412", weight: 10, response_gt: "555200"},

    # (الوزن 100) بديل عام - أولوية منخفضة
    %{weight: 100, response_gt: "555000"}
  ]
```

أمثلة المطابقة:

```
# الرسائل القصيرة GT يتصل بـ A الشريك
"الاتصال: "441234567"، الجهة المتصلة بها: "555100
"تتطابق مع قاعدة الوزن 1 (كل البادئتين) → "555101 →

# الصوت GT يتصل بـ A الشريك
"الاتصال: "441234567"، الجهة المتصلة بها: "555200
"تتطابق مع قاعدة الوزن 10 (الاتصال فقط) → "555200 →

# الرسائل القصيرة GT متصل غير معروف يتصل بـ
"الاتصال: "999999999"، الجهة المتصلة بها: "555100
"تتطابق مع قاعدة الوزن 10 (الجهة المتصلة بها فقط) → "555100 →

# الصوت GT متصل غير معروف يتصل بـ
"الاتصال: "999999999"، الجهة المتصلة بها: "555200
"تتطابق مع قاعدة الوزن 100 البديلة → "555000 →
```

أوضاع التشغيل

يعمل GT NAT عبر جميع أوضاع التشغيل لـ OmniSS7:

وضع HLR

يؤثر GT NAT على:

- استجابات UpdateLocation (GT لـ HLR في الاستجابة)
- رسائل InsertSubscriberData (GT لـ HLR كجهة متصلة)
- استجابات SendAuthenticationInfo
- استجابات Cancel Location

لمزيد من المعلومات حول عمليات HLR، انظر دليل تكوين HLR.

التكوين:

```
config :omniss7,
  hlr_mode_enabled: true,
  hlr_service_center_gt_address: "5551234567",  # GT الافت��اضي لـ
HLR

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "331", weight: 10, response_gt:
"5551234568"},  # فرنسا
    %{calling_prefix: "44", weight: 10, response_gt:
"5551234569"},   # المملكة المتحدة
    %{weight: 100, response_gt: "5551234567"}  # بديل عام
  ]
```

وضع SMSc

يؤثر GT NAT على:

- استجابات SRI-for-SM (حقل `networkNode-Number`) - انظر تفاصيل SRI-for-SM
- تأكيدات MT-ForwardSM

لمزيد من المعلومات حول عمليات SMSc، انظر دليل تكوين SMSc.

التكوين:

```
config :omniss7,
  smsc_mode_enabled: true,
  smsc_service_center_gt_address: "5559999",  # GT الافتراضي لـ
SMSc

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "1", weight: 10, response_gt: "5559991"},
# أمريكا الشمالية
    %{calling_prefix: "44", weight: 10, response_gt: "5559992"},
# المملكة المتحدة
    %{calling_prefix: "86", weight: 10, response_gt: "5559993"},
# الصين
    %{weight: 100, response_gt: "5559999"}  # بديل عام
  ]
```

وضع بوابة CAMEL

يؤثر GT NAT على:

- جميع استجابات مستوى SCCP (GT ـ gsmSCF كجهة متصلة)
- استجابات عمليات CAMEL/CAP (InitialDP، EventReportBCSM، إلخ)
- تأكيدات RequestReportBCSMEvent
- استجابات ApplyCharging
- استجابات Continue

التكوين:

```
config :omniss7,
  camelgw_mode_enabled: true,
  camel_gsmscf_gt_address: "55512341112",  # GT الافتراضي لـ gsmSCF

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "555", weight: 10, response_gt:
"55512341111"},  # الشبكة A
    %{calling_prefix: "666", weight: 10, response_gt:
"55512311555"},  # الشبكة B
    %{weight: 100, response_gt: "55512341112"}  # بديل عام
  ]
```

حالة الاستخدام: عند العمل كـ gsmSCF (وظيفة التحكم في الخدمة) لعدة شبكات، قد يتوقع كل gsmSSF استجابات من GT محدد لـ gsmSCF. يضمن GT NAT استخدام GT الصحيح بناءً على أي GT يتصل gsmSSF.

السجلات واستكشاف الأخطاء

تفعيل سجلات GT NAT

يتضمن GT NAT سجلات تلقائية لجميع الترجمات:

```
# في السجلات، سترى:
[info] GT NAT [استجابة SRI-for-SM]: GT 877234567 المتصل -> GT
55512341112 الاستجابة
[info] GT NAT [UpdateLocation ISD]: GT 331234567 المتصل -> GT
55512341111 الاستجابة
[info] GT NAT [استجابة MAP BEGIN]: GT 441234567 المتصل -> GT
55512311555 الاستجابة
```

يوضح حقل السياق المكان الذي تم فيه تطبيق NAT:

- `"استجابة SRI-for-SM"` - في معالج SRI-for-SM
- `"UpdateLocation ISD"` - في رسائل InsertSubscriberData
- `"UpdateLocation END"` - في استجابة UpdateLocation END
- `"استجابة MAP BEGIN"` - استجابات MAP BEGIN العامة
- `"ISD ACK"` - تأكيد ISD
- `"استجابة خطأ HLR"` - استجابة خطأ من HLR
- `"استجابة CAMEL"` - استجابات عمليات CAMEL/CAP (gsmSCF)

التحقق

يتحقق النظام من تكوين GT NAT عند بدء التشغيل:

```
# تحقق من تكوين GT NAT
iex> GtNat.validate_config()
{:ok, [
  %{calling_prefix: "8772", weight: 10, response_gt:
"55512341112"},
  %{calling_prefix: "8773", weight: 10, response_gt:
"55512341111"}
]}

# تحقق مما إذا كان مفعلًا
iex> GtNat.enabled?()
true

# الحصول على جميع القواعد
iex> GtNat.get_rules()
[
  %{calling_prefix: "8772", weight: 10, response_gt:
"55512341112"},
  %{calling_prefix: "8773", weight: 10, response_gt:
"55512341111"}
]
```

اختبار GT NAT

اختبر منطق GT NAT برمجيًا:

```
# اختبار الترجمة مع GT المتصل فقط (called_gt هو nil)
iex> GtNat.translate_response_gt("877234567", nil, "default_gt")
"55512341112"

# اختبار الترجمة مع كلا من GT المتصل والجهة المتصلة بها
iex> GtNat.translate_response_gt("877234567", "555123",
"default_gt")
"55512341112"

# اختبار مع السجلات (GT الجهة المتصلة بها هو nil)
iex> GtNat.translate_response_gt_with_logging("877234567", nil,
"default_gt", "test")
# السجلات: GT NAT [test]: GT 877234567 المتصل -> GT الاستجابة
55512341112
"55512341112"

# اختبار مع السجلات (كلا GTs)
iex> GtNat.translate_response_gt_with_logging("877234567",
"555123", "default_gt", "test")
# السجلات: GT NAT [test]: GT 877234567 المتصل، GT الجهة المتصلة بها
555123 -> GT 55512341112 الاستجابة
"55512341112"

# اختبار عدم المطابقة (يعود إلى الافتراضي)
iex> GtNat.translate_response_gt("999999999", "888888",
"default_gt")
"default_gt"
```

استكشاف الأخطاء

المشكلة: GT NAT لا يعمل

التحقق 1: هل هو مفعل؟

```
iex> Application.get_env(:omniss7, :gt_nat_enabled)
true  # يجب أن يكون true
```

التحقق 2: هل تم تكوين القواعد؟

```
iex> Application.get_env(:omniss7, :gt_nat_rules)
[%{calling_prefix: "8772", response_gt: "55512341112"}, ...]  # يجب أن تعيد قائمة
```

في السجلات لمعرفة ما إذا كانت "GT NAT" ابحث عن **التحقق 3: تحقق من السجلات** الترجمات تحدث.

المشكلة: GT خاطئ في الاستجابات

العرض: تستخدم الاستجابات عنوان GT غير المتوقع

السبب: قد تكون مطابقة بادئة القاعدة واسعة جدًا أو أن القاعدة الافتراضية تلتقط الحركة

الحل: راجع أوزان القواعد والبادئات:

```
# سيء: بديل مع وزن منخفض (يلتقط كل شيء أولاً)
gt_nat_rules: [
  %{weight: 1, response_gt: "111111"},              # هذا يتطابق مع كل شيء أولاً!
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"}  # لا يتم الوصول إليه أبدًا
]

# جيد: قواعد محددة مع وزن أقل، بديل مع وزن أعلى
gt_nat_rules: [
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"},  # محدد، وزن منخفض
  %{weight: 100, response_gt: "111111"}  # بديل، وزن عالٍ (بديل)
]
```

المشكلة: GT NAT لم يُطبق على نوع رسالة محدد

العرض: بعض الاستجابات تستخدم GT الذي تم ترجمته، والبعض الآخر لا

التغطية الحالية:

- ✅ SCCP (جميع الاستجابات) المتصل GT
- ✅ SRI-for-SM (networkNode-Number) استجابات

- □ رسائل UpdateLocation ISD (GT لـ HLR)
- □ استجابات UpdateLocation END
- □ تأكيدات ISD
- □ استجابات MAP BEGIN

إذا لم يستخدم نوع رسالة محدد GT NAT، تحقق من الشيفرة. فقد لا يكون قد تم تنفيذه بعد. اتصل بالدعم أو راجع المصدرية.

اعتبارات الأداء

أداء البحث

يستخدم GT NAT مطابقة بادئة بسيطة مع تعقيد O(n) حيث n هو عدد القواعد.

نصائح الأداء:

- احتفظ بعدد القواعد أقل من 100 للحصول على أفضل أداء
- استخدم بادئات محددة لتقليل عدد القواعد
- يجب أن تكون القاعدة الافتراضية (بادئة فارغة) في النهاية

الاختبار (نظام نموذجي):

- 10 1 > :قواعد لكل بحث µs
- 50 5 > :قاعدة لكل بحث µs
- 100 10 > :قاعدة لكل بحث µs

استخدام الذاكرة

تتطلب كل قاعدة ~100 بايت من الذاكرة:

- 10 1 ≈ قواعد KB
- 100 10 ≈ قاعدة KB

أفضل الممارسات

1. دائمًا تضمين قاعدة بديلة عامة

```
gt_nat_rules: [
  %{calling_prefix: "8772", weight: 10, response_gt: "111111"},
  %{calling_prefix: "8773", weight: 10, response_gt: "222222"},
  %{weight: 100, response_gt: "default_gt"}  # دائمًا لديك بديل بوزن عالٍ
]
```

2. استخدم بادئات وأوزان ذات معنى

```
# جيد: بادئات واضحة ومحددة مع أوزان مناسبة
%{calling_prefix: "331", weight: 10, response_gt: "..."}  # فرنسا
%{calling_prefix: "44", weight: 10, response_gt: "..."}   # المملكة المتحدة

# سيء: بادئات واسعة جدًا أو أوزان مربكة
%{calling_prefix: "3", weight: 5, response_gt: "..."}     # العديد من البلدان
%{calling_prefix: "331", weight: 100, response_gt: "..."}  # الوزن يجب أن يكون أقل للقواعد المحددة
```

3. وثق قواعدك

```
gt_nat_rules: [
  # الشريك XYZ - الشبكة في المملكة المتحدة (نطاق GT: 4412xxxxxxx)
  # الوزن 10: أولوية الشريك القياسية
  %{calling_prefix: "4412", weight: 10, response_gt: "5551001"},

  # الشريك ABC - الشبكة في فرنسا (نطاق GT: 33123xxxxxx)
  # الوزن 10: أولوية الشريك القياسية
  %{calling_prefix: "33123", weight: 10, response_gt: "5551002"}
]
```

4. اختبر قبل النشر

```
# اختبر في iex قبل النشر
iex> GtNat.translate_response_gt("44121234567", nil, "default")
"5551001"  # النتيجة المتوقعة

# اختبار مع GT بها المتصلة الجهة
iex> GtNat.translate_response_gt("44121234567", "555123",
"default")
"5551001"  # النتيجة المتوقعة
```

5. راقب السجلات

قم بتمكين تسجيل مستوى INFO لرؤية جميع ترجمات GT NAT في الإنتاج.

التكامل مع ميزات أخرى

وضع STP

يعمل GT NAT بشكل مستقل عن توجيه STP. يقوم STP بتوجيه GTs بناءً على رموز النقاط و GT NAT الوجهة، بينما يتعامل مع عنوان الاستجابة GT NAT.

لمزيد من المعلومات حول توجيه STP، انظر دليل تكوين STP.

تكامل CAMEL

GT NAT مع عمليات CAMEL/CAP **مدمج بالكامل**:

طبقة SCCP:

- الجهة المتصلة بها في جميع استجابات GT CAMEL
- يتم تطبيقه تلقائيًا بناءً على GT الوارد لـ gsmSSF

التكوين:

- `camel_gsmscf_gt_address` - GT الافتراضي لـ gsmSCF (اختياري)
- الجهة المتصلة بها من الطلب الوارد GT إذا لم يتم تكوينه، يستخدم

- الافتراضية بناءً على بادئة الجهة المتصلة بها GT NAT تتجاوز قواعد

مثال:

```
# عندما يتصل gsmSSF 555123456 بـ gsmSCF الخاص بك
# الوارد: الجهة المتصلة بها=55512341112، المتصل=555123456
# بحث GT NAT: "555" -> GT 55512341111"="الاستجابة"
# الاستجابة: الجهة المتصلة بها=555123456، المتصل=55512341111
```

توازن الحمل

يمكن دمج GT NAT مع توازن الحمل M3UA لإدارة حركة المرور المتقدمة.

دليل الترحيل

تفعيل GT NAT على نظام موجود

1. **تحضير التكوين**

```
# أضف إلى runtime.exs (احتفظ به معطل في البداية)
config :omniss7,
  gt_nat_enabled: false,  # ابدأ معطلاً
  gt_nat_rules: [
    # قواعدك هنا مع الأوزان
    %{calling_prefix: "877", weight: 10, response_gt:
"111111"},
    %{weight: 100, response_gt: "999999"}  # بديل عام
  ]
```

2. **اختبار التكوين**

```
# تحقق من أن التكوين يتجمع
mix compile

# اختبار في iex
iex -S mix
iex> GtNat.validate_config()
```

3. **تفعيل في البيئة التجريبية**

```
gt_nat_enabled: true  # تغيير إلى true
```

4. **راقب السجلات**

```
tail -f log/omniss7.log | grep "GT NAT"
```

5. **نشر إلى الإنتاج**

    - نشر خلال نافذة الصيانة
    - راقب الساعات الـ 24 الأولى عن كثب
    - احصل على خطة للعودة جاهزة (تعيين `gt_nat_enabled: false`)

الدعم

للمشاكل أو الأسئلة:

- تحقق من السجلات للحصول على رسائل "GT NAT"
- تحقق من التكوين باستخدام `GtNat.validate_config()`
- راجع قسم استكشاف الأخطاء في هذا الدليل
- اتصل بدعم OmniSS7 مع مقتطفات السجل

دليل تكوين HLR

← العودة إلى الوثائق الرئيسية

يوفر هذا الدليل تكوينًا لاستخدام OmniSS7 كـ **سجل الموقع المنزلي (HLR/HSS)** مع **OmniHSS** كقاعدة بيانات المشتركين الخلفية.

تكامل OmniHSS

OmniHSS، وهو خادم **SS7 كواجهة إشارة OmniSS7 في HLR يعمل وضع** تتفاعل مع مشترك منزلي (HSS) كامل الميزات. تفصل هذه البنية بين الاهتمامات:

- **OmniSS7 (واجهة HLR)**: تتعامل مع جميع إشارات بروتوكول SS7/MAP، وتوجيه SCCP، والتواصل الشبكي
- **OmniHSS (الخلفية HSS)**: تدير بيانات المشتركين، والمصادقة، والتزويد، والميزات المتقدمة

لماذا OmniHSS؟

يوفر OmniHSS إدارة مشتركة على مستوى الناقل مع ميزات تشمل:

- **دعم متعدد IMSI**: يمكن لكل مشترك أن يكون لديه عد IMSIs مرتبطة برقم MSISDN واحد للتجوال الدولي، وتبديل الشبكات، وتزويد eSIM
- **مصادقة مرنة**: دعم لكل من خوارزميات المصادقة Milenage (3G/4G/5G) وCOMP128 (2G)
- **تتبع جلسات الدائرة والحزمة**: تتبع مستقل لتسجيلات الشبكة CS (الدائرة المبدلة) وPS (الحزمة المبدلة)
- **تزويد متقدم**: ملفات تعريف الخدمة القابلة للتخصيص، والخدمات التكميلية، وبيانات اشتراك CAMEL
- **تصميم يعتمد على API**: واجهة برمجة تطبيقات HTTP RESTful للتكامل مع أنظمة الفوترة، وإدارة علاقات العملاء، والتزويد
- **تحديثات في الوقت الحقيقي**: تتبع الموقع، وإدارة الجلسات، وتوليد متجهات المصادقة

تُخزن جميع بيانات المشتركين، وبيانات المصادقة، وتكوينات الخدمة وتُدار في OmniHSS. يستعلم OmniSS7 عن OmniHSS عبر استدعاءات API HTTPS للرد على عمليات MAP مثل UpdateLocation، SendAuthenticationInfo، وSendRoutingInfo.

هو **واجهة إشارة فقط**. يتم التعامل مع جميع منطق إدارة OmniSS7 في HLR **مهم**: وضع المشتركين، وخوارزميات المصادقة، وقواعد التزويد، وعمليات قاعدة البيانات بواسطة OmniHSS. يغطي هذا الدليل تكوين بروتوكول SS7/MAP في OmniSS7. لمعلومات حول تزويد المشتركين، وتكوين المصادقة، وملفات تعريف الخدمة، والعمليات الإدارية، **يرجى الرجوع إلى وثائق OmniHSS**.

دعم متعدد IMSI

يدعم OmniHSS تكوينات متعددة IMSI (معرف **بشكل أصلي**، مما يسمح لمشترك واحد برقم MSISDN) بامتلاك عدة IMSIs. وهذا يمكّن من:

- **ملفات تعريف التجوال الدولي**: IMSIs مختلفة لمناطق مختلفة لتقليل تكاليف التجوال
- **ملف تعريف متعدد eSIM**: ملفات تعريف شبكة متعددة على جهاز واحد يدعم eSIM
- **تبديل الشبكة**: تبديل سلس بين الشبكات دون تغيير MSISDN
- **تنسيق شريحة مزدوجة**: التنسيق عبر عدة شرائح فعلية أو افتراضية
- **الاختبار والتطوير**: عدة IMSIs اختبارية تشير إلى نفس المشترك

كيف يعمل:

- لكل IMSI بيانات مصادقة خاصة بها (Ki، OPc، الخوارزمية)
- يمكن أن تحتوي كل IMSI على تسجيلات جلسات دائرة وحزمة مستقلة
- يمكن مشاركة خدمات المشتركين وملفات التعريف أو تخصيصها لكل IMSI
- يستعلم OmniSS7 عن OmniHSS حسب IMSI، وتعيد OmniHSS بيانات المشترك المناسبة
- يمكن أن تتعقب أنظمة الفوترة الاستخدام لكل IMSI بينما تربط جميع IMSIs بحساب واحد

مثال سيناريو متعدد IMSI:

```
للمشترك: +1-555-123-4567 رقم MSISDN
├─ IMSI 1: 310260123456789 (شبكة منزلية أمريكية - مصادقة Milenage)
├─ IMSI 2: 208011234567890 (ملف تعريف التجوال في فرنسا - مصادقة
Milenage)
└─ IMSI 3: 440201234567891 (ملف تعريف التجوال في المملكة المتحدة -
مصادقة COMP128)
```

يمكن استخدام جميع IMSIs الثلاثة بشكل مستقل لتسجيل الشبكة، لكنها جميعًا تنتمي إلى نفس حساب المشترك. تدير OmniHSS خريطة IMSI والتزويد وتضمن المصادقة إلى المشترك المناسبين لكل IMSI.

جدول المحتويات

1. امل��ت OmniHSS
2. دعم متعدد IMSI
3. ما هو وضع HLR؟
4. تمكين وضع HLR
5. قاعدة بيانات المشتركين

6. متجهات المصادقة
7. تحديثات الموقع
8. تكامل CAMEL
9. معالجة المشتركين في التجوال
10. عمليات HLR

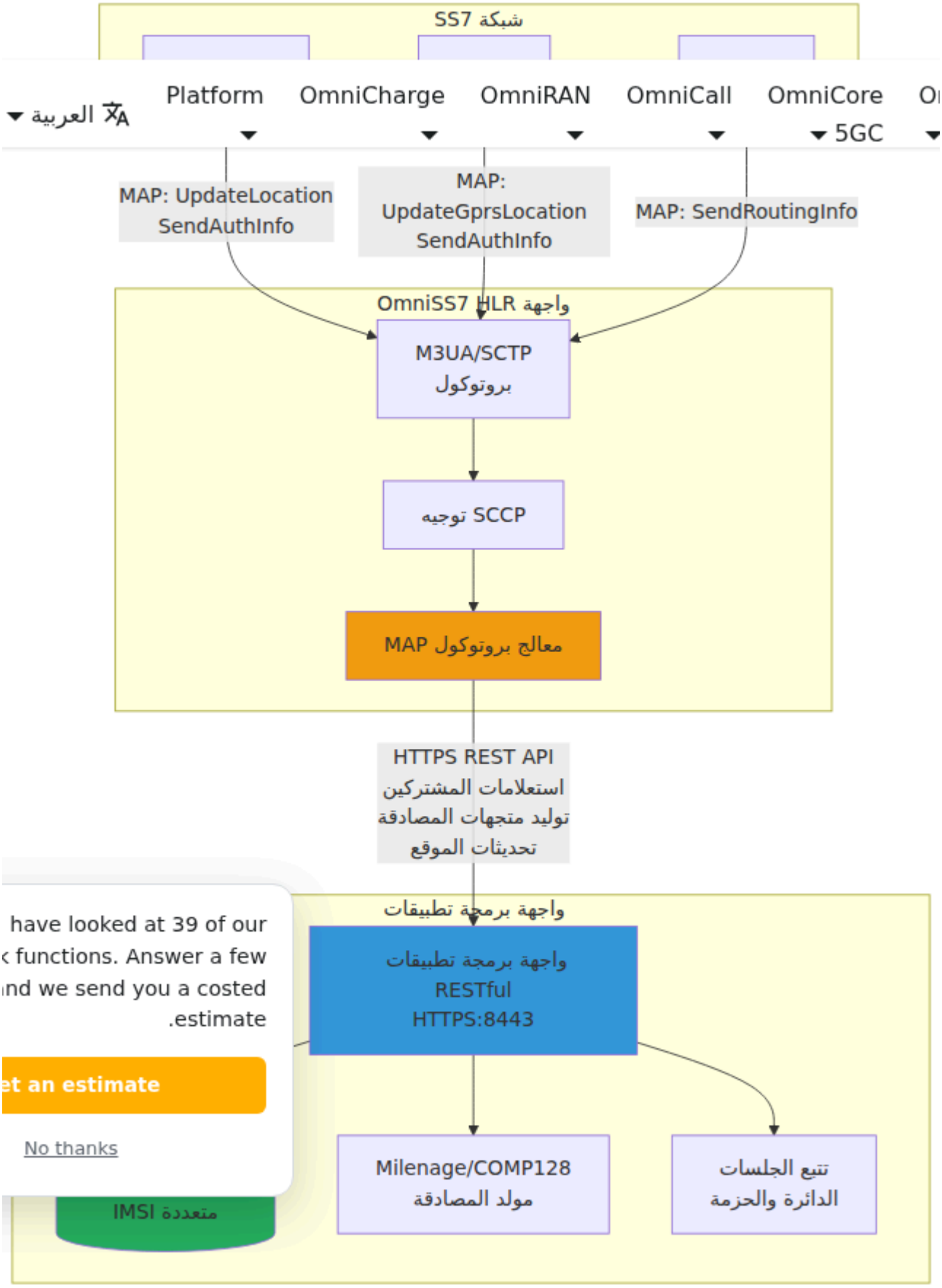
- تعيين حقول الاستجابة
  - SendRoutingInfo (SRI)
  - UpdateLocation / ISD
  - SendRoutingInfoForSM
- ملخص مصدر الحقول

ما هو وضع HLR؟

يمكّن OmniSS7 لـ **وضع HLR** من العمل كسجل موقع منزلي:

- **إدارة المشتركين**: تخزين وإدارة بيانات المشتركين
- **المصادقة**: توليد متجهات المصادقة للوصول إلى الشبكة
- **تتبع الموقع**: معالجة تحديثات الموقع من VLRs
- **معلومات التوجيه**: توفير معلومات التوجيه للمكالمات والرسائل القصيرة

هيكل HLR


شبكة SS7
العربية
Platform
OmniCharge
OmniRAN
OmniCall
OmniCore
5GC
MAP: UpdateLocation
SendAuthInfo
MAP:
UpdateGprsLocation
SendAuthInfo
MAP: SendRoutingInfo
واجهة OmniSS7 HLR
M3UA/SCTP
بروتوكول
توجيه SCCP
معالج بروتوكول MAP
HTTPS REST API
استعلامات المشتركين
توليد متجهات المصادقة
تحديثات الموقع
واجهة برمجة تطبيقات
واجهة برمجة تطبيقات
RESTful
HTTPS:8443
Milenage/COMP128
مولد المصادقة
تتبع الجلسات
الدائرة والحزمة
متعددة IMSI
have looked at 39 of our
functions. Answer a few
nd we send you a costed
.estimate
et an estimate
No thanks

تمكين وضع HLR

يمكن لـ OmniSS7 العمل في أوضاع مختلفة (STP، HLR، SMSc). يتم اختيار الوضع من خلال ملف التكوين، **`config/config.exs`** مجموعة من علامات الوضع والتكوين المرتبط في الافتراضي للتطبيق.

التبديل إلى وضع HLR

لتشغيل OmniSS7 كـ HLR، قم بتعيين علامات الوضع بحيث يتم تمكين ميزات HLR وتعطيل ميزات SMSc، ثم أضف التكوين الخاص بـ HLR:

1. **افتح** `config/config.exs`
2. قم **بتعيين** علامات الوضع تحت `config :omniss7`:
   - `map_client_enabled: true`: مطلوب لإرسال واستقبال عمليات MAP
   - `hlr_mode_enabled: true`: يمكّن معالجة HLR المحددة
   - `smsc_mode_enabled: false`: يعطل ميزات SMSc
3. **أضف** معلمات تكوين HLR (نقطة نهاية API، عناوين GT، إعدادات ISD/CAMEL، اتصال M3UA) الموضحة أدناه
4. **خصص** المعلمات لنشر التطبيق الخاص بك
5. **أعد تشغيل** التطبيق لتفعيل التغييرات

> **ملاحظة**: `config/runtime.exs` هو نموذج بسيط يستخدم للاختبار الآلي فقط. لا يحتوي على تعريفات وضع التشغيل. جميع تكوينات الوضع و HLR موجودة في `config/config.exs` (مع تجاوزات خاصة بالبيئة في ملفات مثل `config/test.exs`). توفر عمليات النشر في الإنتاج ملفها الخاص `config/config.exs`.

تكوين وضع HLR

يبدو تكوين HLR الكامل (في `config/config.exs`) كما يلي:

```
config :omniss7,
  # تمكين ميزات - علامات الوضع فقط HLR
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false,

  # تكوين واجهة برمجة تطبيقات OmniHSS الخلفية
  hlr_api_base_url: "https://omnihss.example.com:8443",
  hlr_api_key: "<مفتاح API الخاص بـ OmniHSS مع دور
hlr_interworking>",

  # عنوان GT لمركز خدمة HLR لعمليات SMS
  hlr_service_center_gt_address: "1234567890",

  # تكوين خريطة MSISDN ↔ IMSI
  # انظر: قسم خريطة MSISDN ↔ IMSI للحصول على التفاصيل
  hlr_imsi_plmn_prefix: "50557",
  hlr_msisdn_country_code: "61",
  hlr_msisdn_nsn_offset: 0,
  hlr_msisdn_nsn_length: 9,

  # تكوين InsertSubscriberData
  # وضع الوصول إلى الشبكة: :packetAndCircuit، :packetOnly، أو
:circuitOnly
  isd_network_access_mode: :packetAndCircuit,

  # إرسال ISD #2 (بيانات الخدمات التكميلية)
  isd_send_ss_data: true,

  # إرسال ISD #3 (بيانات حظر المكالمات)
  isd_send_call_barring: true,

  # تكوين CAMEL (لردود SendRoutingInfo)
  # مفتاح الخدمة لبدء خدمة CAMEL
  camel_service_key: 11_110,

  # نقطة اكتشاف الزناد CAMEL
  # الخيارات: :termAttemptAuthorized، :tBusy، :tNoAnswer، :tAnswer
  camel_trigger_detection_point: :termAttemptAuthorized,

  # بادئات VLR المنزلية
  # قائمة بادئات عنوان VLR منزلية "شبكة" التي تعتبر
  # إذا بدأ VLR المشترك بأحد هذه البادئات، استخدم استجابة SRI
```

```
القياسية
  # للحصول PRN خلاف ذلك، يكون المشترك في التجوال ونحتاج إلى إرسال
على MSRN
  # إذا (تعامل كالتجوال VLRs لا توجد بادئات = جميع) [] الافتراضي هو
تم حذفها
  home_vlr_prefixes: ["555123"],

  # تكوين اتصال M3UA
  # الاتصال كـ ASP لاستقبال عمليات MAP (UpdateLocation،
SendAuthInfo، إلخ.)
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    # نقطة النهاية المحلية (نظام HLR)
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    # نقطة النهاية البعيدة STP
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }
```

معلمات التكوين للتخصيص

للحصول على مرجع كامل لجميع معلمات التكوين، انظر مرجع التكوين.

المعلمة	النوع	الافتراضي
`hlr_api_base_url`	سلسلة	*مطلوب*
`hlr_api_key`	سلسلة	*مطلوب*
`hlr_service_center_gt_address`	سلسلة	*مطلوب*
`smsc_service_center_gt_address`	سلسلة	*مطلوب*
`hlr_smsc_alert_gts`	قائمة	`[]`
`hlr_alert_location_expiry_seconds`	عدد صحيح	`172800`
`hlr_imsi_plmn_prefix`	سلسلة	`"50557"`

المعلمة	النوع	الافتراضي
`hlr_msisdn_country_code`	سلسلة	`"61"`
`hlr_msisdn_nsn_offset`	عدد صحيح	`0`
`hlr_msisdn_nsn_length`	عدد صحيح	`9`
`isd_network_access_mode`	ذرة	`:packetAndCircuit`
`isd_send_ss_data`	منطقي	`true`
`isd_send_call_barring`	منطقي	`true`
`camel_service_key`	عدد صحيح	`11_110`
`camel_trigger_detection_point`	ذرة	`:termAttemptAuthorized`

المعلمة	النوع	الافتراضي
`hlr_camel_subscriptions`	قائمة	`[]`
`home_vlr_prefixes`	قائمة	`[]`
`local_ip`	مجموعة	*مطلوب*
`local_port`	عدد صحيح	`2905`
`remote_ip`	مجموعة	*مطلوب*
`remote_port`	عدد صحيح	`2905`
`routing_context`	عدد صحيح	`1`

ماذا يحدث عند تمكين وضع HLR

عند `hlr_mode_enabled: true`، ستظهر واجهة الويب:

- **أحداث SS7** - تسجيل الأحداث
- **عميل SS7** - اختبار عمليات MAP
- **الأقران** (أقران M3UA/SCTP) - حالة الاتصال

- 🔗 **روابط HLR** - إدارة ←لمشتركين محدد *لـ* + HLR حالة واجهة برمجة تطبيقات *HLR*
- 📊 **الموارد** - مراقبة النظام
- ⚙️ **التكوين** - عارض التكوين

SMSc ستكون علامات التبويب **التوجيه**، **اختبار التوجيه**، و**روابط** مخفية.

ملاحظات مهمة

- **التكوين المطلوب**: تعتبر المعلمة `hlr_service_center_gt_address` **إلزامية**. سيفشل التطبيق في البدء إذا لم يتم تكوينها.
- **الخلفية OmniHSS**: يجب أن تكون واجهة برمجة تطبيقات OmniHSS الخلفية قابلة للوصول عند `hlr_api_base_url` المكون.
- **مفتاح واجهة برمجة تطبيقات OmniHSS**: `hlr_api_key` إلزامي. يرسله OmniHSS، وأيضًا عند في كل طلب لـ `x-omnihss-key` في رأس OmniSS7 التحقق من حالة روابط HLR. إذا كان المفتاح مفقودًا أو فارغًا، يفشل كل طلب لـ OmniHSS بخطأ يذكر `hlr_api_key`. لا يرسل OmniSS7 إذا طلبًا بدون المفتاح. كان المفتاح غير صحيح، تعيد OmniHSS HTTP 401.
- **مهلة طلب واجهة برمجة التطبيقات**: جميع طلبات واجهة برمجة تطبيقات لديها **مهلة ثابتة مدتها 5 ثوانٍ** OmniHSS
- **مهلة طلب MAP**: جميع طلبات MAP (SRI، UpdateLocation، SendAuthInfo، إلخ.) لديها **مهلة ثابتة مدتها 10 ثوانٍ**
- **مهلة ISD**: تحتوي كل رسالة InsertSubscriberData (ISD) في تسلسل UpdateLocation على **مهلة ثابتة مدتها 10 ثوانٍ**
- MAP لاستقبال عمليات STP إلى M3UA يتطلب الاتصال بـ
- بعد تغيير الأوضاع، يجب إعادة تشغيل التطبيق لتفعيل التغييرات
- **واجهة الويب**: انظر دليل واجهة الويب للحصول على معلومات حول استخدام الواجهة الويب
- **الوصول إلى واجهة برمجة التطبيقات**: انظر دليل واجهة برمجة التطبيقات لوثائق Swagger UI والوصول إلى واجهة REST واجهة برمجة التطبيقات

قاعدة بيانات المشتركين

يدير OmniHSS بما في ذلك الهويات، وبيانات المصادقة، وملفات **جميع بيانات المشتركين** هذه البيانات عبر استدعاءات واجهة برمجة OmniSS7 تعريف الخدمة، ومعلومات الموقع. يسترجع التطبيقات RESTful.

OmniHSS نموذج مشترك

يخزن OmniHSS معلومات شاملة عن المشتركين:

- **عدة IMSIs لكل مشترك**: دعم لتكوينات متعددة IMSI (eSIM، ملفات تعريف التجوال، تبديل الشبكات)
- **بيانات المصادقة**: Ki، OPc، واختيار الخوارزمية (Milenage أو COMP128)
- **ملفات تعريف الخدمة**: فئة المشترك، الخدمات المسموح بها، معلمات جودة الخدمة
- **تتبع الموقع**: تتبع مستقل لـ VLR/MSC و SGSN/GGSNالحالي (جلسة الدائرة) (جلسة الحزمة)
- **بيانات اشتراك CAMEL**: مفاتيح الخدمة، نقاط الزناد، وعناوين gsmSCF
- **الخدمات التكميلية** CLIP/CLIR: إعادة توجيه المكالمات، الحظر، الانتظار، تكوينات
- **الحالة الإدارية**: مفعل/معطل، قيود الخدمة، تواريخ انتهاء الصلاحية

متجهات المصادقة

توليد متجهات المصادقة

يولد OmniHSS **متجهات المصادقة** باستخدام خوارزميات Milenage أو COMP128 بناءً على طريقة المصادقة المكونة لكل مشترك. عندما يتلقى OmniSS7 طلبات MAP **sendAuthenticationInfo**:

1. يستخرج OmniSS7 IMSI من طلب MAP
2. يستدعي OmniSS7 واجهة برمجة تطبيقات OmniHSS لتوليد متجهات المصادقة
3. يسترجع OmniHSS بيانات Ki وOPc الخاصة بالمشترك
4. يولد OmniHSS العدد المطلوب من المتجهات (RAND، XRES، CK، IK، AUTN)
5. يقوم OmniSS7 بترميز المتجهات إلى تنسيق MAP ويعيدها إلى VLR/SGSN المطلوب

تكامل واجهة برمجة تطبيقات OmniHSS

يتواصل OmniSS7 مع OmniHSS عبر واجهة برمجة تطبيقات HTTPS REST لاسترجاع معلومات المشتركين، وتحديث بيانات الموقع، وتوليد متجهات المصادقة:

```
config :omniss7,
  hlr_api_base_url: "https://omnihss-server:8443",
  hlr_api_key: "<مفتاح API الخاص بـ OmniHSS مع دور
hlr_interworking>"
```

يحتوي كل طلب إلى OmniHSS على رأس `x-omnihss-key` مع قيمة `hlr_api_key`. تستخدم OmniHSS المفتاح للعثور على دور العميل. يجب أن يكون مفتاح HLR لديه دور `hlr_interworking`. يمنح هذا الدور الوصول إلى نقاط النهاية الخاصة بالبحث عن المشتركين، وتعيين المفاتيح، ومتجهات المصادقة، وجلسات الدائرة التي يستخدمها HLR.

عندما يتلقى OmniSS7 عمليات MAP من شبكة SS7، يستعلم عن OmniHSS لـ:

- حسب MSISDN أو IMSI **استرجاع بيانات المشترك**
- المخزنة Ki/OPc باستخدام بيانات **توليد متجهات المصادقة**
- UpdateLocation عندما يقوم المشتركون بإجراء **تحديث موقع جلسة الدائرة**
- وحقوق الخدمة **التحقق من حالة المشترك**

تحديثات الموقع

معالجة تحديث الموقع

عند استلام طلبات MAP **updateLocation**، يتعاون OmniSS7 مع OmniHSS لتسجيل المشترك في VLR جديد:

1. من طلب UpdateLocation (IMSI، GT VLR الجديد، GT MSC الجديد) **استخراج معلومات الموقع**
2. **استعلام OmniHSS** للتحقق من وجود المشترك وتمكينه
3. في OmniHSS بموقع VLR/MSC الجديد **تحديث جلسة الدائرة**
4. **إرسال رسائل InsertSubscriberData (ISD)** لتزويد المشترك في VLR الجديد

5. **إرجاع استجابة UpdateLocation** إلى VLR (تشمل GT HLR من `hlr_service_center_gt_address`)
6. **إرسال alertServiceCenter** إلى GTs الخاصة بـ SMSc المكونة (إذا كانت `hlr_smsc_alert_gts` مملوءة)

ملاحظة: تحدد المعلمة `hlr_service_center_gt_address` عنوان GT العالمي لـ HLR الذي يتم إرجاعه في ردود UpdateLocation. وهذا يسمح لـ VLR/MSC بتحديد وتوجيه الرسائل مرة أخرى إلى هذا HLR.

تكامل مركز الخدمة التنبيهي

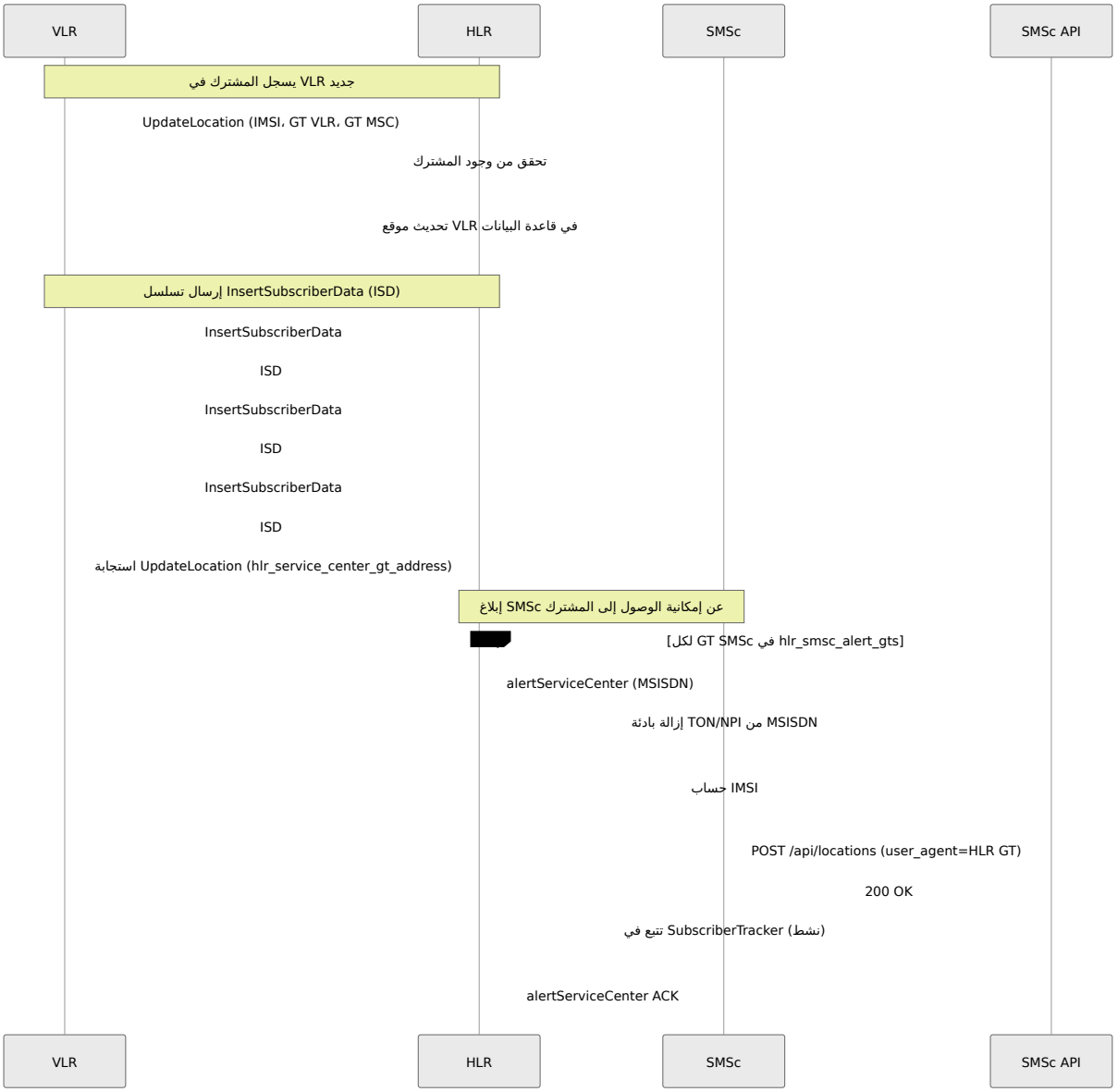
بعد نجاح UpdateLocation، يمكن لـ HLR تلقائيًا إبلاغ أنظمة SMSc بأن المشترك أصبح الآن قابلًا للوصول من خلال إرسال رسائل **alertServiceCenter** (عملية MAP 64). لمعلومات حول كيفية تعامل SMSc مع هذه التنبيهات، انظر دليل معالجة مركز الخدمة التنبيهي في SMSc.

التكوين

قم بتكوين قائمة عناوين GTs الخاصة بـ SMSc لإبلاغها:

```
config :omniss7,
  # قائمة GTs الخاصة بـ SMSc لإرسال alertServiceCenter بعد
UpdateLocation
  hlr_smsc_alert_gts: [
    "15559876543",
    "15559876544"
  ],

  # وقت انتهاء الموقع عندما يتلقى SMSc alertServiceCenter
(الافتراضي: 48 ساعة)
  hlr_alert_location_expiry_seconds: 172800
```

مخطط التدفق



السلوك

عندما يقوم المشترك بإجراء UpdateLocation:

1. يرسل HLR alertServiceCenter **كل** إلى GT SMSc في قائمة `hlr_smsc_alert_gts`
2. تتضمن الرسالة MSISDN الخاص بالمشترك
3. يستخدم HLR `hlr_service_center_gt_address` كـ GT للمتصل
4. توجيه SCCP: SSN 6=المتصل (HLR)، SSN 8=المدعو (SMSc)

يتلقى SMSc التنبيه و:

- **يزيل بادئة TON/NPI** من MSISDN (→ "19123123213"، على سبيل المثال "123123213"")

- يحدد المشترك على أنه قابل للوصول في قاعدة بيانات موقعه (عبر POST إلى /api/locations)
- **يحدد حقل** `user_agent` إلى GT HLR عند استدعاء واجهة برمجة التطبيقات (للتتبع أي HLR أرسل التنبيه)
- يحدد وقت انتهاء الموقع بناءً على `hlr_alert_location_expiry_seconds`
- يتتبع المشترك في SMSc Subscriber Tracker للمراقبة

الاختبار

استخدم صفحة **المشتركين النشطين** في واجهة الويب لإرسال رسائل alertServiceCenter يدويًا للاختبار:

1. انتقل إلى علامة التبويب "المشتركين النشطين"
2. ابحث عن قسم "اختبار مركز خدمة التنبيه"
3. أدخل MSISDN و SMSc GT و HLR GT (القيم الافتراضية مُعبأة مسبقًا من التكوين)
   - SMSc GT الافتراضي هو أول إدخال في `hlr_smsc_alert_gts`
   - HLR GT الافتراضي هو `hlr_service_center_gt_address`
4. انقر على "إرسال alertServiceCenter"

هذا مفيد لاختبار معالجة تنبيهات SMSc دون الحاجة إلى تدفق UpdateLocation كامل. يستخدم النموذج التحقق من `phx-blur` لتجنب عرض الأخطاء أثناء الكتابة.

تكوين InsertSubscriberData (ISD)

بعد نجاح UpdateLocation، يرسل HLR إلى VLR بيانات توفير المشترك باستخدام رسائل **InsertSubscriberData** (ISD). يسمح لك تكوين ISD بتخصيص البيانات المرسلة وكيفية ذلك.

لتكوين، راجع تكوين ISD في مرجع التكوين لمرجع معلمات ��.

تسلسل ISD

يمكن أن يرسل HLR ما يصل إلى 3 رسائل ISD متتالية:

1. **ISD #1** (يتم إرساله دائمًا) - بيانات المشترك الأساسية:
   - IMSI
   - MSISDN

- فئة المشترك
- حالة المشترك (serviceGranted)
- قائمة خدمات الناقل
- قائمة خدمات الهاتف
- وضع الوصول إلى الشبكة

2. **ISD #2** (اختياري) - بيانات الخدمات التكميلية (SS):
   - إعدادات تحويل المكالمات (غير مشروطة، مشغول، عدم الرد، غير متاح)
   - انتظار المكالمات
   - تعليق المكالمات
   - خدمة متعددة الأطراف
   - حالة وميزات الخدمة التكميلية
3. **ISD #3** (اختياري) - بيانات حظر المكالمات:
   - حظر جميع المكالمات الصادرة (BAOC)
   - حظر المكالمات الدولية الصادرة (BOIC)
   - بيانات قيود الوصول

خيارات التكوين

```
# تكوين InsertSubscriberData
# وضع الوصول إلى الشبكة: :packetAndCircuit، :packetOnly، أو
:circuitOnly
isd_network_access_mode: :packetAndCircuit,

# إرسال ISD #2 (بيانات الخدمات التكميلية)
isd_send_ss_data: true,

# إرسال ISD #3 (بيانات حظر المكالمات)
isd_send_call_barring: true,
```

وضع الوصول إلى الشبكة

يحدد معلمة `isd_network_access_mode` نوع الوصول إلى الشبكة المسموح به للمشترك:

القيمة	الوصف	حالة الاستخدام
`:packetAndCircuit`	كل من تبديل الحزم (GPRS/LTE) وتبديل الدوائر (الصوت)	الافتراضي - مشتركون بخدمات كاملة
`:packetOnly`	تبديل الحزم فقط (البيانات/LTE)	بطاقات SIM للبيانات فقط، أجهزة IoT
`:circuitOnly`	تبديل الدوائر فقط (الصوت/SMS)	أجهزة قديمة، خطط صوتية فقط

التحكم في رسائل ISD

يمكنك التحكم في الرسائل ISD المرسلة بناءً على متطلبات الشبكة الخاصة بك:

إرسال جميع ISDs (افتراضي - مجموعة ميزات كاملة):

```
isd_send_ss_data: true,
isd_send_call_barring: true,
```

إرسال بيانات المشترك الأساسية فقط (توفير الحد الأدنى):

```
isd_send_ss_data: false,
isd_send_call_barring: false,
```

إرسال الأساسية + الخدمات التكميلية (بدون حظر المكالمات):

```
isd_send_ss_data: true,
isd_send_call_barring: false,
```

مثال على تدفق ISD

عند استلام UpdateLocation:

```
VLR → HLR: UpdateLocation (BEGIN)
HLR → VLR: InsertSubscriberData #1 (CONTINUE) - البيانات الأساسية
VLR → HLR: ISD #1 ACK (CONTINUE)
HLR → VLR: InsertSubscriberData #2 (CONTINUE) - بيانات SS [إذا
كانت مفعلة]
VLR → HLR: ISD #2 ACK (CONTINUE)
HLR → VLR: InsertSubscriberData #3 (CONTINUE) - حظر المكالمات [إذا
كانت مفعلة]
VLR → HLR: ISD #3 ACK (CONTINUE)
HLR → VLR: UpdateLocation Response (END)
```

إذا كانت `isd_send_ss_data` أو `isd_send_call_barring` مضبوطة على `false`، يتم تخطي تلك الرسائل ISD، ويتم إرسال UpdateLocation END في وقت أقرب.

أفضل الممارسات

- **تكوين افتراضي**: استخدم `:packetAndCircuit` وقم بتمكين جميع ISDs لتحقيق أقصى توافق
- **IoT/M2M**: استخدم `:packetOnly` وقم بتعطيل بيانات SS/حظر المكالمات للأجهزة التي تعمل بالبيانات فقط
- **التشغيل البيني**: قد لا تدعم بعض VLRs القديمة جميع الخدمات التكميلية - قم بتعطيل `isd_send_ss_data` إذا واجهت مشكلات
- **الأداء**: تعطيل ISDs غير المستخدمة يقلل من الحمل الزائد للرسائل ويسرع تحديثات الموقع

تكامل CAMEL

تكوين CAMEL لإرسال معلومات التوجيه

يمكن لـ HLR، عند الرد على طلبات **SendRoutingInfo** (SRI) من GMSC (Gateway MSC)، توجيه GMSC لاستدعاء خدمات CAMEL في الخدمة لتوجيه المكالمات الذكي والتحكم.

لمرجع معلمات التكوين، راجع تكوين CAMEL في مرجع التكوين.

ما هو CAMEL؟

CAMEL (تطبيقات مخصصة لشبكة الهاتف المحمول المنطق المعزز) هو بروتوكول يمكّن خدمات الشبكة الذكية في شبكات GSM/UMTS. يسمح لمشغلي الشبكات بتنفيذ خدمات ذات قيمة مضافة مثل:

- الفوترة المدفوعة مسبقًا
- تصفية المكالمات وحظرها
- الشبكات الخاصة الافتراضية (VPN)
- خدمات الأسعار المميزة
- تحويل المكالمات مع منطق مخصص
- خدمات تعتمد على الموقع

خيارات التكوين

```
# تكوين CAMEL (لردود SendRoutingInfo)
# مفتاح الخدمة لبدء خدمة CAMEL
camel_service_key: 11_110,

# نقطة اكتشاف الزناد CAMEL
# الخيارات: :termAttemptAuthorized، :tBusy، :tNoAnswer، :tAnswer
camel_trigger_detection_point: :termAttemptAuthorized,
```

مفتاح الخدمة

يحدد `camel_service_key` أي خدمة CAMEL يجب استدعاؤها في gsmSCF (وظيفة التحكم في الخدمة). هذا هو معرف رقمي مُعد في شبكتك:

مفتاح الخدمة	حالة الاستخدام النموذجية
`11_110`	التحكم في المكالمات المدفوعة مسبقًا (افتراضي)
`100`	خدمة مدفوعة مسبقًا للمكالمات الصادرة
`200`	تحويل المكالمات مع منطق مخصص
`300`	الشبكة الخاصة الافتراضية (VPN)
مخصص	خدمات محددة من قبل المشغل

مثال على التكوين:

```
# للتحكم في المكالمات المدفوعة مسبقًا
camel_service_key: 11_110,

# لخدمة VPN
camel_service_key: 300,
```

نقطة اكتشاف الزناد

أثناء إعداد CAMEL متى يجب تشغيل خدمة `camel_trigger_detection_point` تحدد المكالمة:

متى يتم تشغيلها	الوصف	نقطة الاكتشاف
قبل توجيه المكالمة إلى المشترك	محاولة المكالمة مصرح بها (افتراضي)	`:termAttemptAuthorized`
عندما يكون المشترك مشغولًا	مشغول نهائي	`:tBusy`
عندما لا يرد المشترك	عدم الرد النهائي	`:tNoAnswer`
عندما يرد المشترك على المكالمة	الرد النهائي	`:tAnswer`

أمثلة على التكوين:

التحكم المدفوع مسبقًا القياسي (تشغيل قبل التوجيه):

```
camel_trigger_detection_point: :termAttemptAuthorized,
```

معالجة مشغول مخصصة (تشغيل عند المشغول):

```
camel_trigger_detection_point: :tBusy,
```

(تشغيل عند الرد) **الفوترة المعتمدة على الرد**:

```
camel_trigger_detection_point: :tAnswer,
```

استجابة SRI مع CAMEL

عند التكوين، تتض��ن ردود SendRoutingInfo معلومات اشتراك CAMEL:

```
GMSC → HLR: SendRoutingInfo (BEGIN)
HLR → GMSC: SRI Response (END) مع:
  - IMSI
  - رقم VLR
  - حالة المشترك
  - معلومات توجيه CAMEL:
    * مفتاح الخدمة: 110_11
    * عنوان gsmSCF: <العنوان المُعد>
    * نقطة اكتشاف الزناد: termAttemptAuthorized
    * معالجة المكالمات الافتراضية: continueCall

يتصل GMSC بـ gsmSCF عند نقطة الزناد لتنفيذ خدمة CAMEL
```

أفضل الممارسات

- **الشبكات الإنتاجية**: استخدم مفاتيح الخدمة القياسية المتفق عليها مع مزود gsmSCF الخاص بك
- **الاختبار**: استخدم `:termAttemptAuthorized` لأكثر اختبارات شمولية
- **الخدمات المدفوعة مسبقًا**: مفتاح الخدمة `110_11` هو معيار صناعي شائع للمكالمات المدفوعة مسبقًا النهائية
- **معالجة النسخ الاحتياطي**: يضمن `defaultCallHandling: :continueCall` استمرار المكالمات إذا كان gsmSCF غير متاح

اشتراكات CAMEL في InsertSubscriberData

تحمل InsertSubscriberData واحدة أخرى من HLR، يرسل UpdateLocation أثناء المشترك مع قاعدة في IMSI عندما يتطابق `vlrCamelSubscriptionInfo` للمكالمات MSC O-CSI القاعدة المطابقة الأولى تفوز. تستخدم `hlr_camel_subscriptions`. الصادرة (DP collectedInfo) و VT-CSI للمكالمات الواردة (DP termAttemptAuthorized)،

وتفتح حوار CAP نحو `gsmscf_address` العنوان العالمي، بحيث يتم اختيار gsmSCF بواسطة HLR، وليس بواسطة تكوين MSC.

```
config :omniss7,
  hlr_camel_subscriptions: [
    %{
      imsi_prefix: "3133800000009",
      o_csi: %{service_key: 100, gsmscf_address: "14155550300",
default_call_handling: :release_call},
      t_csi: %{service_key: 101, gsmscf_address: "14155550300",
default_call_handling: :continue_call}
    }
  ]
```

المفتاح	الوصف
`imsi_prefix`	بادئة IMSI التي تنطبق عليها القاعدة
`o_csi` / `t_csi`	على الأقل واحد. `service_key` (مفتاح الخدمة InitialDP)، `gsmscf_address` (gsmSCF E.164 GT)، `default_call_handling` (`:release_call` أو `:continue_call`، يتم تطبيقه بواسطة MSC عندما لا يرد gsmSCF)

يتم إرسال CSIs مع معالجة قدرة CAMEL 2 (CAP v2). تثير قاعدة تحتوي على مفتاح مفقود خطأ باسم المفتاح.

معالجة المشتركين أثناء التجوال

الكشف عن VLR التجوال VLR المحلي مقابل

يحتاج إلى تحديد ما إذا كان المشترك على **SendRoutingInfo** (SRI)، طلب HLR عند استلام تجوال (يزور شبكة أخرى). يختلف السلوك بناءً على هذا VLR محلي" (داخل شبكتك) أو على" VLR التحديد:

لمرجع معلمات التكوين، راجع بادئات VLR المحلية في مرجع التكوين.

- **VLR** إرجاع استجابة **المحلي**: SRI القياسية مع معلومات توجيه CAMEL
- **VLR** إرسال طلب توفير رقم التجوال **التجوال**: (PRN) للحصول على MSRN، ثم إرجاعه في استجابة SRI

التكوين

```
# بادئات VLR المحلية
# قائمة بادئات عنوان VLR شبكة محلية "التي تعتبر"
# إذا بدأ عنوان VLR استخدم هذه البادئات، الخاص بالمشترك بأحد
استجابة SRI القياسية
# خلاف ذلك، يكون المشترك في التجوال ونحتاج إلى إرسال PRN للحصول على
MSRN
home_vlr_prefixes: ["555123"],
```

مثال على التكوين:

```
# شبكة محلية واحدة
home_vlr_prefixes: ["555123"],

# شبكات محلية متعددة (مثل، مناطق أو فروع مختلفة)
home_vlr_prefixes: ["555123", "555124", "555125"],
```

كيف يعمل

1. تدفق المشترك المحلي (قياسي)

عندما يبدأ عنوان VLR الخاص بالمشترك ببادئة محلية مُعدة:

```
GMSC → HLR: SendRoutingInfo (MSISDN: "1234567890")
HLR يستعلم عن بيانات المشترك من واجهة برمجة التطبيقات الخلفية
HLR يتحقق من عنوان VLR: "5551234567"
HLR يحدد: VLR شبكة محلية → "555123" يبدأ بـ
HLR → GMSC: استجابة SRI مع معلومات توجيه CAMEL:
  - IMSI
  - رقم VLR: "5551234567"
  - عنوان gsmSCF (MSC): "5551234501"
  - مفتاح خدمة CAMEL: 11_110
  - نقطة اكتشاف الزناد: termAttemptAuthorized
```

2. تدفق المشترك في التجوال (PRN مطلوب)

عندما لا يتطابق عنوان VLR الخاص بالمشترك مع أي بادئة محلية:

```
GMSC → HLR: SendRoutingInfo (MSISDN: "1234567890")
HLR يستعلم عن بيانات المشترك من واجهة برمجة التطبيقات الخلفية
HLR يتحقق من ��نوان VLR: "49170123456"
HLR يحدد: VLR تجوال → "555123" لا يبدأ بـ
HLR → MSC: ProvideRoamingNumber (PRN):
  - MSISDN: "1234567890"
  - IMSI: "999999876543210"
  - رقم MSC: "49170123456"
  - عنوان GMSC: "5551234501"
MSC → HLR: استجابة PRN مع MSRN: "49170999888777"
HLR → GMSC: استجابة SRI مع معلومات التوجيه:
  - IMSI
  - رقم VLR: "49170123456"
  - رقم التجوال (MSRN): "49170999888777"
```

اختلافات هيكل الاستجابة

استجابة SRI للمشترك المحلي

```
%{
  imsi: "999999876543210",
  extendedRoutingInfo: {
    :camelRoutingInfo, %{
      gmscCamelSubscriptionInfo: %{
        "t-CSI": %{
          serviceKey: 11_110,
          "gsmSCF-Address": "5551234501",
          defaultCallHandling: :continueCall,
          "t-BcsmTriggerDetectionPoint": :termAttemptAuthorized
        }
      }
    }
  },
  subscriberInfo: %{
    locationInformation: %{"vlr-number": "5551234567"},
    subscriberState: {:notProvidedFromVLR, :NULL}
  }
}
```

استجابة SRI للمشترك في التجوال

```
%{
  imsi: "999999876543210",
  extendedRoutingInfo: {
    :routingInfo, %{
      roamingNumber: "49170999888777"  # MSRN من PRN
    }
  },
  subscriberInfo: %{
    locationInformation: %{"vlr-number": "49170123456"},
    subscriberState: {:notProvidedFromVLR, :NULL}
  }
}
```

عملية توفير رقم التجوال (PRN)

هيكل طلب PRN

يتضمن طلب PRN المرسل إلى MSC/VLR:

الحقل	المصدر	الوصف
MSISDN	طلب SRI	رقم هاتف المشترك
IMSI	واجهة برمجة التطبيقات HLR	IMSI الخاص بالمشترك
رقم MSC	واجهة برمجة التطبيقات HLR	MSC الذي يخدم المشترك في التجوال (`serving_msc`)
عنوان GMSC	طلب SRI	GMSC الذي يقوم بإجراء طلب SRI الأصلي
رقم مرجع المكالمة	ثابت	معرف مرجع المكالمة
مراحل CAMEL المدعومة	ثابت	مراحل CAMEL المدعومة من GMSC

معالجة استجابة PRN

يتوقع HLR استجابة PRN على تحتوي:

- **MSRN** (رقم التجوال لمحطة الهاتف المحمول): رقم مؤقت مُخصص بواسطة الشبكة الزائرة لتوجيه المكالمة

معالجة الأخطاء:

- إذا انتهى وقت PRN → في استجابة (المشترك غائب) 27 يتم إرجاع الخطأ SRI
- إذا فشل PRN → في استجابة (المشترك غائب) 27 يتم إرجاع الخطأ SRI
- إذا لم يمكن استخراج MSRN → في استجابة (المشترك غائب) 27 يتم إرجاع الخطأ SRI

أمثلة على التكوين

مشغل شبكة محلية واحدة

```
# جميع عناوين VLR تعتبر محلية "555123" التي تبدأ بـ
home_vlr_prefixes: ["555123"],
```

- VLR `5551234567` → محلي (استجابة CAMEL)
- VLR `5551235001` → محلي (استجابة CAMEL)
- VLR `49170123456` → تجوال (PRN + استجابة MSRN)

مشغل متعدد المناطق

```
# شبكات محلية متعددة عبر مناطق مختلفة
home_vlr_prefixes: ["555123", "555124", "555125"],
```

- VLR `5551234567` → محلي (المنطقة 1)
- VLR `5552341234` → محلي (المنطقة 2)
- VLR `5553411111` → محلي (المنطقة 3)
- VLR `44201234567` → تجوال (دولي)

تكوين الاختبار

لاختبار وظيفة PRN، قم بتعيين قائمة فارغة لمعالجة جميع VLRs كـ تجوال:

```
# يتم التعامل مع جميع VLRs كـ تجوال (لاختبار تدفق PRN)
home_vlr_prefixes: [],
```

أفضل الممارسات

- **اختيار البادئات**: استخدم أقصر بادئة فريدة تحدد VLRs الخاصة بشبكتك (مثل، رمز الدولة + رمز الشبكة)
- **بادئات متعددة**: تضمين جميع بادئات VLR في شبكتك، بما في ذلك مناطق وفروع مختلفة
- **اتفاقيات التجوال**: تأكد من دعم PRN بشكل صحيح من قبل الشبكات الشريكة في التجوال
- **الاختبار**: اختبر سيناريوهات المحلية والتجوال بدقة قبل نشر الإنتاج
- **المراقبة**: راقب معدلات انتهاء وقت PRN لتحديد مشكلات الاتصال مع الشركاء في التجوال

استكشاف الأخطاء وإصلاحها

الأعراض: يتم التعامل مع جميع المشتركين كـ تجوال

- **السبب**: لم يتم تكوين `home_vlr_prefixes` أو لا تتطابق البادئات مع عناوين VLR
- **الحل**: تحقق من عناوين VLR في قاعدة بياناتك وقم بتحديث البادئات وفقًا لذلك

الأعراض: طلبات PRN تنتهي وقتها

- **السبب**: مشكلات الاتصال بالشبكة مع MSC/VLR الشريكة في التجوال
- **الحل**: تحقق م�� توجيه M3UA/SCCP إلى عناوين MSC البعيدة

الأعراض: SRI غير صالح في استجابة MSRN

- **السبب**: تنسيق استجابة PRN من الشريك في التجوال لا يتطابق مع الهيكل المتوقع
- **الحل**: راجع سجلات استجابة PRN وقم بتعديل `extract_msrn_from_prn/1` إذا لزم الأمر

عمليات HLR

عمليات MAP المدعومة

تقوم OmniSS7 بتوزيع عمليات MAP الواردة حسب opcode (بغض النظر عن مرحلة TCAP). يتم التعامل مع العمليات الواردة التالية:

Opcode	العملية	الغرض
2	`updateLocation`	تسجيل موقع VLR (يؤدي إلى تسلسل ISD + alertServiceCenter اختياري)
23	`updateGprsLocation`	تسجيل موقع SGSN (مبدل الحزم)
3	`cancelLocation`	إلغاء التسجيل من VLR القديم
7	`insertSubscriberData`	تلقي ملف تعريف المشترك (مثل عند العمل كجانب VLR)
22	`sendRoutingInfo` (SRI)	توفير MSRN/CAMEL للتوجيه للمكالمات (محلي مقابل تجوال)
45	`sendRoutingInfoForSM` (SRI-for-SM)	توفير IMSI + SMSC GT لتوجيه SMS
44	`MT-Forward-SM`	تسليم SMS مُنتهي إلى الهاتف المحمول
46	`MO-Forward-SM`	استقبال SMS مُنطلق من الهاتف المحمول
47	`reportSM-DeliveryStatus`	تقرير حالة تسليم SMS-GMSC
64	`alertServiceCenter`	تنبيه قابل للوصول إلى المشترك إلى SMSc
66	`readyForSM`	تشير إلى أن المشترك MSC/SGSN جاهز لـ SMS
56	`sendAuthenticationInfo`	توليد متجهات المصادقة
57	`restoreData`	استعادة بيانات المشترك في VLR
67	`purgeMS`	حذف سجل مشترك في HLR

Opcode	العملية	الغرض
59	`processUnstructuredSS-Request`	USSD (يتم التعامل معه عبر بوابة USSD عند التمكين)
71	`anyTimeInterrogation` (ATI)	استجواب موقع/حالة المشترك
10	`registerSS`	تسجيل الخدمة التكميلية (مثل تحويل المكالمات غير المشروطة)

تُجاب العمليات غير المعالجة بخطأ `facilityNotSupported` (21).

رسم خرائط حقول الاستجابة

تتناول هذه القسم أين تأتي كل حقل في استجابات HLR.

استجابة SendRoutingInfo (SRI)

الغرض: يوفر معلومات التوجيه للمكالمات الواردة إلى مشترك.

يوفر HLR على ما إذا كان المشترك بناءً نوعين مختلفين من الاستجابة في VLR أو في محلي التجوال:

استجابة المشترك المحلي (توجيه CAMEL)

تستخدم VLR بادئة المشترك الخاص بعندما يبدأ عنوان `home_vlr_prefixes` مُعدة.

هيكل الاستجابة:

الحقل	المصدر	الوصف
IMSI	واجهة برمجة التطبيقات OmniHSS	IMSI الخاص بالمشترك من قاعدة بيانات OmniHSS
رقم VLR	واجهة برمجة التطبيقات OmniHSS	VLR الحالي الذي يخدم المشترك (`circuit_session.assigned_vlr`)
حالة المشترك	ثابت	دائمًا `notProvidedFromVLR`
extendedRoutingInfo	-	النوع: `camelRoutingInfo`
عنوان gsmSCF	واجهة برمجة التطبيقات OmniHSS	MSC الذي يخدم المشترك (`circuit_session.assigned_msc`)
مفتاح الخدمة	config.exs	معرف خدمة CAMEL (`camel_service_key`)
نقطة اكتشاف الزناد	config.exs	متى يتم تشغيل CAMEL (`camel_trigger_detection_point`)
معالجة قدرة CAMEL	ثابت	مستوى دعم مرحلة CAMEL
معالجة المكالمات الافتراضية	ثابت	غير gsmSCF النسخة الاحتياطية إذا كان متاح

استجابة المشترك في التجوال (توجيه MSRN)

`home_vlr_prefixes` الخاص بالمشترك مع أي بادئة VLR تستخدم عندما لا يتطابق عنوان مُعدة.

هيكل الاستجابة:

الحقل	المصدر	الوصف	
IMSI	واجهة برمجة التطبيقات OmniHSS	IMSI الخاص بالمشترك من قاعدة بيانات OmniHSS	"9
رقم VLR	واجهة برمجة التطبيقات OmniHSS	VLR الحالي الذي يخدم المشترك (`circuit_session.assigned_vlr`)	"4
حالة المشترك	ثابت	دائمًا `notProvidedFromVLR`	:n
extendedRoutingInfo	-	النوع: `routingInfo`	-
رقم التجوال (MSRN)	استجابة PRN	MSRN الذي تم الحصول عليه من طلب ProvideRoamingNumber	"4

منطق قرار التوجيه:

```
1. تتلقى OmniSS7 طلب SendRoutingInfo
2. تستعلم OmniSS7 عن بيانات المشترك من واجهة برمجة التطبيقات
OmniHSS
3. تتحقق OmniSS7 من عنوان VLR مقابل home_vlr_prefixes:

   إذا بدأ VLR بالبادئة المحلية:
     → إرجاع معلومات توجيه CAMEL (تدفق المشترك المحلي)

   إذا لم يتطابق VLR مع أي بادئة محلية:
     → إرسال ProvideRoamingNumber (PRN) إلى MSC
     → استخراج MSRN من استجابة PRN
     → إرجاع معلومات التوجيه مع MSRN (تدفق المشترك في التجوال)
```

تدفق البيانات:

- تستعلم OmniSS7 عن معلومات المشترك من OmniHSS
- تعيد OmniHSS IMSI، موقع VLR/MSC الحالي، وحالة المشترك
- تستخدم OmniSS7 هذه البيانات لبناء استجابة MAP

متطلبات التكوين:

```
# في config/config.exs
home_vlr_prefixes: ["555123"],  # قائمة بادئات VLR المحلية
```

استجابات الأخطاء:

- يتم إرجاع الخطأ 27 (المشترك: `null` `serving_msc` و `serving_vlr` إذا كانت غائب)
- إذا لم يتم العثور على المشترك: يتم إرجاع الخطأ 1 (مشترك غير معروف)
- يتم إرجاع الخطأ 27 (المشترك غائب) :(حالة التجوال) PRN إذا انتهى وقت طلب
- غير صالحة (حالة التجوال): يتم إرجاع الخطأ 27 (المشترك PRN إذا كانت استجابة غائب)

استجابة UpdateLocation مع InsertSubscriberData

الغرض: تسجيل المشترك في VLR جديد وتوفير بيانات المشترك.

استجابة UpdateLocation END

الحقل	المصدر	الوصف	المثال
رقم HLR	config.exs	عنوان العنوان العالمي لـ HLR (`hlr_service_center_gt_address`)	`"5551234568"`
نوع رسالة TCAP	ثابت	الاستجابة النهائية بعد جميع ISDs	`END`

InsertSubscriberData #1 (بيانات المشترك الأساسية)

الحقل	المصدر	الوصف	المثال
IMSI	الطلب	من طلب UpdateLocation	`"999999876543210"`
MSISDN	OmniHSS API	رقم هاتف المشترك من OmniHSS	`"555123456"`
الفئة	ثابت	فئة المشترك	`"\n"` (0x0A)
حالة المشترك	ثابت	حالة الخدمة	`:serviceGranted`
قائمة خدمات الناقل	ثابت	خدمات الناقل المدعومة	`[<<31>>]`
قائمة الخدمات التليفونية	ثابت	الخدمات التليفونية المدعومة	`[<<17>>, "!", "\""]`
وضع الوصول إلى الشبكة	config.exs	الوصول إلى الحزمة/الدائرة (`isd_network_access_mode`)	`:packetAndCircuit`

InsertSubscriberData #2 (الخدمات التكميلية) - اختياري

الحقل	المصدر	الوصف	يتحكم فيه
SS المجهزة	ثابت	بيانات الخدمات التكميلية	`isd_send_ss_data: true`
إعادة توجيه المكالمات	ثابت	تكوينات إعادة التوجيه (غير مشروط، مشغول، لا رد، غير قابل للوصول)	تم تمكين التكوين
انتظار المكالمات	ثابت	حالة خدمة انتظار المكالمات	تم تمكين التكوين
خدمة متعددة الأطراف	ثابت	دعم المكالمات الجماعية	تم تمكين التكوين

ISD #2 تشمل:

- إعادة توجيه المكالمات غير المشروطة (رمز SS 21)
- إعادة توجيه المكالمات عند الانشغال (رمز SS 41)
- إعادة توجيه المكالمات عند عدم الرد (رمز SS 42)
- إعادة توجيه المكالمات عند عدم الوصول (رمز SS 62)
- انتظار المكالمات (رمز SS 43)
- خدمة متعددة الأطراف (رمز SS 51)
- خدمات CLIP/CLIR

InsertSubscriberData #3 (حظر المكا��مات) - اختياري

الحقل	المصدر	الوصف	يتحكم فيه
معلومات حظر المكالمات	ثابت	تكوينات حظر المكالمات	`isd_send_call_barring: true`
BAOC	ثابت	حظر جميع المكالمات الصادرة (رمز SS 146)	تم تمكين التكوين
BOIC	ثابت	حظر المكالمات الدولية الصادرة (رمز SS 147)	تم تمكين التكوين
بيانات قيود الوصول	ثابت	قيود الوصول إلى الشبكة	تم تمكين التكوين

تحكم تسلسل ISD:

- ISD #1: يحتوي على بيانات المشترك الأساسية - **يتم إرساله دائمًا**
- ISD #2: يتم إرساله فقط إذا كان `isd_send_ss_data: true` في config/config.exs
- ISD #3: يتم إرساله فقط إذا كان `isd_send_call_barring: true` في config/config.exs

استجابة SendRoutingInfoForSM (SRI-for-SM)

الغرض: توفير معلومات توجيه MSC/SMSC لتسليم SMS. عندما يحتاج SMSc إلى تسليم SMS إلى مشترك، فإنه يرسل طلب SRI-for-SM إلى HLR لتحديد مكان توجيه الرسالة.

في الطلب: تجيب HLR في أحد وضعين، يتم تحديدهما بواسطة **الطرف المتصل SCCP GT**

1. التوجيه المنزلي (افتراضي: كل منشئ خارجي). يعيد هذا الوضع IMSI اصطناعي أبدًا عن HLR كرقم عقدة الشبكة. لا تكشف IP-SM-GW الخاص بـ GT بالإضافة إلى IMSIs الحقيقية والعقد الخدمية لمشغلين خارجيين. يعود IP-SM-GW، الذي يتولى MT-ForwardSM إلى بعد ذلك التسليم. هذا هو السلوك الصحيح لحركة المرور بين الشبكات والتجوال.

الحقل	المصدر	الوصف	كيف تم إنشاؤه	
IMSI	محسوب	IMSI اصطناعي مشتق من MSISDN	`PLMN_PREFIX + zero_padded_MSISDN`	`"00100`
رقم عقدة الشبكة	config.exs	عنوان GT لـ SMSC لتوجيه SMS	`smsc_service_center_gt_address`	`"55512`

2. التوجيه المباشر على الشبكة (فقط عندما يأتي الطلب من أحد عقد SMSc / IP-SM-GW الخاصة بنا). عندما يتطابق GT المتصل مع قائمة `home_smsc_gts` المكونة، تبحث **الخدمية الحقيقية** كرقم **MSC الحقيقي** و **IMSI** عن المشترك في قاعدة بياناتها وتعيد HLR عقدة الشبكة، حتى يتمكن SMSc الخاص بنا من إرسال MT-ForwardSM مباشرة إلى MSC. إذا لم يكن هناك خدمية متاحة لذلك المشترك، فإنها تعود إلى التوجيه المنزلي MSC.

الحقل	المصدر	الوصف	المثال
IMSI	قاعدة بيانات HLR	IMSI الحقيقي للمشترك	`"505570123456789"`
رقم عقدة الشبكة	قاعدة بيانات HLR	`serving_msc` الحقيقي من آخر UpdateLocation	`"5559876543"`

معلمات التكوين (من `config/config.exs`):

```
# عنوان GT لمركز الخدمة (يتم إرجاعه في استجابات SRI-for-SM الموجهة
إلى المنزل)
# هذا يخبر SMSc المطلوب أين يرسل رسائل MT-ForwardSM
smsc_service_center_gt_address: "5551234567",  # مطلوب

# التوجيه المباشر على الشبكة: GTs المتصلة SCCP من SMSc / IP-SM-GW
الخاصة بنا.
# يحصل SRI-for-SM الخدمية الحقيقية بدلاً من MSC من أي من هؤلاء على
التوجيه المنزلي.
# يتم توجيه الجميع إلى المنزل. اترك فارغًا ([]) لتوجيه كل شيء إلى
المنزل.
home_smsc_gts: ["5551234567"],                # اختياري (افتراضي
([]

# تكوين ربط MSISDN ↔ IMSI
# بادئة PLMN: MCC (001 = شبكة اختبار) + MNC (01 = مشغل اختبار)
hlr_imsi_plmn_prefix: "001001",                # المعلمة الوحيدة
المطلوبة!
```

> **ملاحظة:** يعمل التوجيه المباشر على الشبكة فقط إذا كانت HLR تحتفظ فعليًا بـ `serving_msc` للمشترك. أي أن MSC الخدمية قامت بإجراء MAP `updateLocation` إلى HLR هذه. إذا قام MSC بالتسجيل عبر آلية أخرى (على سبيل المثال، GSUP إلى HLR)، فإن الخلفية لا تحتوي على الموقع ويتم توجيه الاستعلام إلى المنزل (منفصل).

ربط MSISDN ↔ IMSI

معلمات التكوين:

تتحكم هذه المعلمات في كيفية توليد MSISDNs لاستجابات SRI-for-SM الاصطناعية من IMSIs لـ OmniSS7:

- **`hlr_imsi_plmn_prefix`**: بادئة MCC+MNC لاستخدامها عند بناء IMSIs الاصطناعية (على سبيل المثال، "50557" لـ MCC=505، MNC=57)
- **`hlr_msisdn_country_code`**: رمز الدولة للإضافة عند القيام بربط عكسي IMSI→MSISDN (على سبيل المثال، "61" لأستراليا، "1" للولايات المتحدة/كندا)
- **`hlr_msisdn_nsn_offset`**: موضع الحرف الذي يبدأ فيه رقم المشترك الوطني (NSN) داخل MSISDN (رمز الدولة، أو طول رمز MSISDN عادة 0 إذا لم يتضمن الدولة إذا كان يتضمنه)

- `hlr_msisdn_nsn_length`: عدد الأرقام المستخرجة من MSISDN كـ NSN

للحصول على تفاصيل إضافية حول التكوين، راجع ربط MSISDN ↔ IMSI في مرجع التكوين.

لماذا يحتاج ربط MSISDN إلى IMSI؟

يتطلب بروتوكول MAP لـ **SendRoutingInfoForSM** (SRI-for-SM) أن تعيد HLR **IMSI** المطلوب فقط SMSc في استجابتها. ومع ذلك، يعرف (هوية المشترك الدولي المحمول) **MSISDN** (رقم الهاتف) للمشترك.

في شبكة تقليدية:

- يرسل SMSc SRI-for-SM مع MSISDN (على سبيل المثال، الوجهة "5551234567")
- يجب على HLR البحث عن المشترك في قاعدة بياناتها للعثور على IMSI الخاص بهم
- تعيد HLR IMSI في استجابة SRI-for-SM
- ثم يستخدم SMSc هذا IMSI عند إرسال MT-ForwardSM إلى MSC/VLR

نهج OmniSS7 - IMSIs الاصطناعية:

بدلاً من الاحتفاظ بقاعدة بيانات كاملة للمشتركين مع ربط MSISDN إلى IMSI، تستخدم OmniSS7 نظام ترميز بسيط لـ **حساب** IMSIs الاصطناعية مباشرة من MSISDN. يوفر هذا النهج فائدتين رئيسيتين:

1. **الخصوصية**: لا يتم الكشف عن IMSIs الحقيقية للمشتركين المخزنة في قاعدة بيانات HLR أبدًا في استجابات SRI-for-SM المرسلة عبر شبكة SS7
2. **البساطة**: لا حاجة للاستعلام عن قاعدة بيانات HLR لعمليات البحث عن IMSI أثناء عمليات SRI-for-SM - يتم حساب IMSI في الوقت الفعلي من MSISDN

كيف يعمل:

يتم ترميز MSISDNs مباشرة في الجزء الخاص بالمشترك من IMSI (بعد الأرقام MCC+MNC):

```
IMSI = PLMN_PREFIX + zero_padded_MSISDN
```

حيث:

- **PLMN_PREFIX**: MCC + MNC (على سبيل المثال، "001001" لشبكة الاختبار)
- **MSISDN**: جميع الأرقام الرقمية من رقم الهاتف

- **Zero Padding**: يتم حشوها من اليسار بالأصفار لملء IMSI إلى 15 رقمًا بالضبط

مثال خطوة بخطوة:

```
# التكوين
plmn_prefix = "001001"  # MCC 001 + MNC 01

# الإدخال: MSISDN من طلب SRI-for-SM (تم فك تشفير TBCD)
msisdn = "555123456"    # 9 أرقام

# الخطوة 1: حساب المساحة المتاحة لرقم المشترك
subscriber_digits = 15 - String.length("001001")  # = 9 أرقام

# الخطوة 2: حشو MSISDN المشترك بالأصفار من اليسار لملء الجزء الخاص بالمشترك
padded_msisdn = String.pad_leading("555123456", 9, "0")  # =
"555123456" (لا حاجة للحشو)

# الخطوة 3: دمج بادئة PLMN + MSISDN المحشو
imsi = "001001" <> "555123456"  # = "001001555123456" (15 رقمًا
بالضبط)
```

أمثلة كاملة:

MSISDN المدخل	بادئة PLMN	الأرقام المتاحة للمشترك	MSISDN المحشو	IMSI النهائي
`"555123456"`	`"001001"` (6)	9	`"555123456"`	`"001001555123456`
`"99"`	`"001001"` (6)	9	`"000000099"`	`"001001000000099`
`"999999999"`	`"001001"` (6)	9	`"999999999"`	`"001001999999999`
`"91123456789"`	`"001001"` (6)	9	`"555123456"`	`"001001555123456`

معالجة الحالات الحدودية:

- **MSISDNs القصيرة**: حشو بالأصفار من اليسار (على سبيل المثال، `"99"` → `"000000099"`)
- **MSISDNs الطويلة**: يتم الاحتفاظ بأقصى الأرقام من اليمين، ويتم تقليم الأرقام من اليسار (على سبيل المثال، `"91123456789"` → `"555123456"`)
- **طول IMSI**: دائمًا 15 رقمًا بالضبط

الربط العكسي (IMSI → MSISDN):

يمكن لـ SMSc عكس هذا الربط لتحويل IMSIs مرة أخرى إلى MSISDNs:

```
# الإدخال: IMSI من استجابة SRI-for-SM
imsi = "001001555123456"

# الخطوة 1: إزالة بادئة PLMN
plmn_prefix = "001001"
subscriber_portion = String.slice(imsi, 6, 9)  # = "555123456"

# الخطوة 2: إزالة الأصفار من اليسار للحصول على MSISDN الفعلي
msisdn = String.replace_leading(subscriber_portion, "0", "")  # =
"555123456"
```

أمثلة الربط العكسي:

المدخل IMSI	بادئة PLMN	الجزء الخاص بالمشترك	إزالة الأصفار من اليسار	MSISDN النهائي
`"001001555123456"`	`"001001"`	`"555123456"`	`"555123456"`	`"55512345`
`"001001000000099"`	`"001001"`	`"000000099"`	`"99"`	`"99"`
`"001001999999999"`	`"001001"`	`"999999999"`	`"999999999"`	`"99999999`

خصائص هذا الربط:

- ✅ **حتمي**: دائمًا ينتج نفس MSISDN نفس IMSI
- ✅ **قابل للعكس**: يمكن تحويله مرة أخرى من MSISDN إلى IMSI
- ✅ **تكوين بسيط**: يتطلب فقط `hlr_imsi_plmn_prefix`
- ✅ **يحافظ على الخصوصية**: لا يتم الكشف عن IMSIs الحقيقية أبدًا
- ✅ **لا حاجة للاستعلام عن قاعدة البيانات**: حساب سريع، لا حاجة لاستدعاءات API
- ✅ **دائمًا 15 رقمًا**: IMSI دائمًا 15 رقمًا بالضبط

معالجة إدخال MSISDN:

عندما تتلقى HLR طلب SRI-for-SM، يخضع MSISDN لفك تشفير TBCD:

1. **فك تشفير TBCD**: تحويل TBCD الثنائي إلى سلسلة (قد يتضمن بادئة TON/NPI مثل "91")

2. **استخراج الأرقام**: الاحتفاظ بالأرقام الرقمية فقط، وإزالة أي أحرف غير رقمية
3. **تطبيع**: إذا كان أطول من المساحة المتاحة، يتم أخذ الأرقام من اليمين؛ إذا كان أقصر، يتم حشوه بالأصفار من اليسار
4. **ترميز**: دمج بادئة PLMN + MSISDN المعالج

اعتبارات الأمان:

تُستخدم IMSIs الاصطناعية المعادة في استجابات SRI-for-SM لأغراض التوجيه فقط. إنها ليست IMSIs الحقيقية المخزنة في قاعدة بيانات المشتركين في HLR. يوفر ذلك طبقة إضافية من حماية IMSIs الخصوصية، حيث يتم الكشف عن IMSIs الحقيقية فقط عند الضرورة المطلقة (على سبيل المثال، أثناء عمليات SendAuthenticationInfo أو UpdateLocation التي تتطلب متجهات مصادقة حقيقية).

تدفق الاستجابة:

```
1. SMSc → HLR: طلب SRI-for-SM
   - MSISDN (TBCD): "91123456789" (يتضمن TON/NPI)

2. معالجة HLR:
   - فك تشفير TBCD: "91123456789"
   - استخراج الأرقام: "91123456789" (11 رقمًا)
   - ملاءمة إلى 9 أرقام: "555123456" (أقصى 9 من اليمين)
   - إضافة PLMN: "001001" + "555123456" = "001001555123456"
   - الحصول على GT SMSC من التكوين: "5551234567"

3. HLR → SMSc: استجابة SRI-for-SM
   - IMSI: "001001555123456" (اصطناعي، دائمًا 15 رقمًا)
   - رقم عقدة الشبكة: "5551234567" (أين يجب إرسال MT-ForwardSM)

4. يرسل SMSc MT-ForwardSM مع "5551234567" إلى IMSI
"001001555123456"
```

التكوين:

تستخدم المعلمات التالية في `config/config.exs`:

```
# بادئة PLMN: MCC (001 = شبكة اختبار) + MNC (01 = مشغل اختبار)
hlr_imsi_plmn_prefix: "001001",

# استخراج NSN (إذا كانت MSISDNs تتضمن رمز الدولة)
hlr_msisdn_country_code: "1",      # مستخدم للربط العكسي (IMSI→MSISDN)
hlr_msisdn_nsn_offset: 1,          # تخطي رمز الدولة المكون من رقم واحد
hlr_msisdn_nsn_length: 10          # استخراج NSN المكون من 10 أرقام
```

تكوين استخراج NSN:

إذا كانت MSISDNs تتضمن رمز الدولة (على سبيل المثال، `"99988000088"` بدلاً من مجرد `"88000088"` NSN)، يجب عليك تكوين استخراج NSN:

- **`hlr_msisdn_nsn_offset`**: الموضع الذي يبدأ فيه NSN (عادةً طول رمز الدولة لديك)
- **`hlr_msisdn_nsn_length`**: عدد الأرقام في NSN

أمثلة:

المثال	رمز الدولة	مثال MSISDN	nsn_offset	nsn_length	NSN المستخرج
رمز الدولة المكون من رقم واحد	`"9"`	`"95551234567"`	`1`	`10`	`"555123456`
رمز الدولة المكون من رقمين	`"99"`	`"99412345678"`	`2`	`9`	`"412345678`
رمز الدولة المكون من ثلاثة أرقام	`"999"`	`"99988000088"`	`3`	`8`	`"88000088"`

كيف يعمل:

1. **MSISDN → IMSI**: استخراج NSN من MSISDN، حشوه بالأصفار من اليسار، دمجه مع بادئة PLMN

```
MSISDN: "99988000088"
NSN: String.slice("99988000088", 3, 8) = "88000088"
NSN (أرقام 9) "088000088" :المحشو
IMSI: "547050" + "088000088" = "547050088000088"
```

2. **IMSI → MSISDN**: إزالة بادئة PLMN، إزالة الأصفار من اليسار، إضافة رمز الدولة

```
IMSI: "547050088000088"
088000088" :الجزء الخاص بالمشترك: "
88000088" :إزالة الأصفار: "
MSISDN: "+999" + "88000088" = "+99988000088"
```

متطلبات API: يستخدم SRI-for-SM فقط. القيم المحسوبة والتكوين للخلفية API لا تتطلب أي استدعاءات.

ملخص مصادر الحقول

نوع المصدر	الوصف	الأمثلة
OmniHSS API	بيانات ديناميكية من قاعدة بيانات المشتركين في OmniHSS	IMSI، MSISDN، VLR/MSC من الخدمية circuit_session
config.exs	معلمات تكوين OmniSS7	`smsc_service_center_gt_address`، `camel_service_key`، `isd_network_access_mode`
ثابت	قيم مشفرة في مولد الاستجابة	SS حالة المشترك، خدمات الناقل، رموز
طلب	الحقول المستخرجة من الطلب الوارد MAP	IMSI من UpdateLocation، MSISDN من SRI
محسوب	القيم المستمدة باستخدام المنطق	IMSI الاصطناعي في SRI-for-SM (hlr_imsi_prefix + NSN)

تبعيات التكوين

مطلوب في config/config.exs:

- `hlr_service_center_gt_address` - مستخدم في استجابات UpdateLocation
- `smsc_service_center_gt_address` - مستخدم في استجابات SRI-for-SM (أين يجب توجيه MT-ForwardSM)

اختياري في config/config.exs (مع القيم الافتراضية):

- `camel_service_key` - `110_11` :الافتراضي
- `camel_trigger_detection_point` - الافتراضي: `:termAttemptAuthorized`
- `isd_network_access_mode` - الافتراضي: `:packetAndCircuit`
- `isd_send_ss_data` - الافتراضي: `true`
- `isd_send_call_barring` - الافتراضي: `true`
- `hlr_imsi_plmn_prefix` - `001001"` :الافتراضي" (بادئة PLMN لربط MSISDN↔IMSI)

مطلوب من OmniHSS:

يجب على OmniHSS توفير نقاط نهاية REST API لـ:

- البحث عن المشترك حسب MSISDN و IMSI
- تحديث مواقع جلسات الدائرة (تعيين VLR/MSC)
- توليد متجهات المصادقة
- استعلامات حالة المشترك وملف الخدمة

الوثائق ذات الصلة

وثائق OmniSS7:

- ← العودة إلى الوثائق الرئيسية
- دليل الميزات الشائعة
- دليل عميل MAP
- المرجع الفني

- مرجع التكوين

وثائق OmniHSS: لإدارة المشتركين، والتوفير، وتكوين المصادقة، والعمليات الإدارية، راجع **وثائق منتج OmniHSS**. تحتوي OmniHSS على جميع منطق قاعدة بيانات المشتركين، وخوارزميات المصادقة، وقواعد توفير الخدمة، وقدرات إدارة Multi-IMSI.

OmniSS7 بواسطة Omnitouch Network Services

MAP دليل تكوين عميل

← العودة إلى الوثائق الرئيسية

يغطي هذا الدليل كيفية تشغيل OmniSS7 كـ **عميل** **MAP**. في هذا الدور، يتصل OmniSS7 بشبكة SS7 كعملية خادم تطبيق (ASP). يقوم MAP بإصدار عمليات تجاه (جزء التطبيق المحمول) وعناصر الشبكة الأساسية الأخرى. يتبع سلوك البروتوكول GMLCs وSGSNs وMSC/VLRs وHLRs 3GPP TS 29.002.

جدول المحتويات

1. ما هو وضع عميل MAP؟
2. تمكين وضع عميل MAP
3. مرجع عمليات MAP
4. أمثلة على العمليات الشائعة
5. إرسال الطلبات عبر API
6. المقاييس والمراقبة

7. استك��اف الأخطاء وإصلاحها

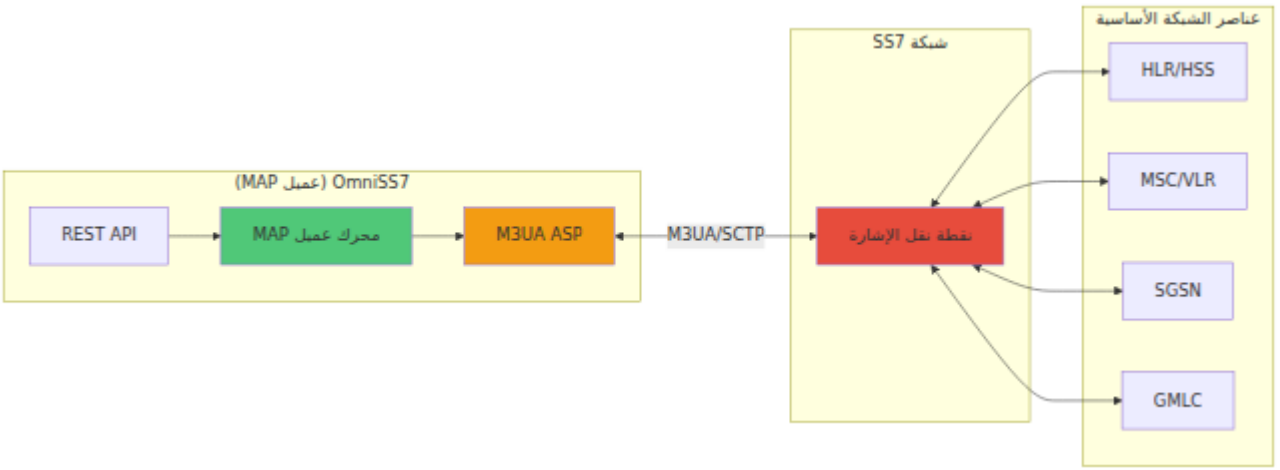
ما هو وضع عميل MAP؟

إلى نظير **(ASP)** بالاتصال كـ **عملية خادم تطبيق** يسمح لـ OmniSS7 **وضع عميل MAP**. يتم عرض كل عملية كنقطة نهاية HTTP. وتبادل رسائل **MAP** عبر SCTP عبر (STP أو SGP) M3UA. يقوم API بتشفير الطلب كحوار TCAP/MAP وينقله عبر SCCP وM3UA إلى شبكة SS7. ثم يقوم بفك تشفير استجابة الشبكة وإعادتها بشكل متزامن إلى المتصل OmniSS7.

تشمل الاستخدامات النموذجية:

- **إدارة التنقل**: تحديث الموقع، إلغاء الموقع، تطهير MS، توفير رقم التجوال
- **معالجة المكالمات**: إرسال معلومات التوجيه، توفير معلومات المشترك
- **SMS**: SRI-for-SM، MO/MT-ForwardSM، تقارير حالة التسليم
- **المصادقة**: إرسال معلومات المصادقة
- **بيانات المشترك**: إدراج/حذف بيانات المشترك، استجواب في أي وقت
- **الخدمات التكميلية**: تسجيل/مسح/تفعيل/استجواب SS، معالجة كلمة المرور
- **خدمات الموقع (LCS) وتوجيه GPRS**

بنية الشبكة



MAP تمكين وضع عميل

يتم تكوين وضع عميل MAP في `config/runtime.exs`. للحصول على مرجع كامل لارتباط M3UA، راجع معلمات اتصال M3UA.

```
config :omniss7,
  # تمكين وضع عميل MAP
  map_client_enabled: true,

  # اتصال M3UA لعميل MAP (يتصل كـ ASP بنقطة STP/SGP بُعد عن)
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :map_client_asp,
    local_ip: {10, 0, 0, 100},
    local_port: 2905,
    remote_ip: {10, 0, 0, 1},
    remote_port: 2905,
    routing_context: 1
  }
```

معلمات `map_client_m3ua`

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`mode`	سلسلة	نعم	-	دور M3UA. `"ASP"` لوضع العميل (يتصل بنقطة STP/SGP).
`callback`	مجموعة	نعم	-	`{Module, Function, Args}` يتم استدعاؤه لكل حمولة M3UA واردة. بالنسبة لعميل MAP، يكون هذا هو `{MapClient, :handle_payload, []}`.
`process_name`	ذرة	نعم	-	الاسم المسجل لعملية M3UA ASP.
`local_ip`	مجموعة	نعم	-	عنوان ربط SCTP المحلي كزوج IP، مثل `{10, 0, 0, 100}`.
`local_port`	عدد صحيح	لا	2905	منفذ SCTP المحلي. 2905 هو منفذ M3UA المسجل لدى IANA.
`remote_ip`	مجموعة	نعم	-	عنوان IP لنقطة STP/SGP عن بُعد كزوج.
`remote_port`	عدد صحيح	لا	2905	منفذ SCTP لنقطة STP/SGP عن بُعد.
`routing_context`	عدد صحيح	لا	-	قيمة سياق توجيه M3UA، عندما يتطلب النظير واحدًا.

> المفاتيح العليا: `map_client_enabled` (Boolean، الافتراضي `false`) يقوم بتشغيل عميل MAP؛ `map_client_m3ua` (خريطة) يحتفظ بالاتصال أعلاه.

تقوم SCCP بتكوين عنوان (عناوين الأطراف المتصلة/المتصل بها) للعمليات الصادرة، رموز SSNs، النقاط بشكل منفصل. راجع مرجع التكوين.

مرجع عمليات MAP

رموز العمليات هي رموز `POST /api/<endpoint>` كل عملية أدناه يمكن الوصول إليها كـ. يقوم: الطلبات متزامنة API بحظر حتى تصل استجابة SS7 من 3GPP TS 29.002 عمليات MAP أو تنتهي مهلة الطلب (انظر رموز الاستجابة).

إدارة التنقل

العملية	نقطة النهاية	رمز العملية	الوصف
تحديث الموقع	`/api/updateLocation`	2	تسجيل VLR الخادم لمشترك مع HLR.
إلغاء الموقع	`/api/cancelLocation`	3	توجيه VLR لحذف سجل مشترك.
توفير رقم التجوال	`/api/prn`	4	الحصول على MSRN من MSC الخادم.
تطهير MS	`/api/purgeMS`	67	وضع علامة على مشترك على أنه تم تطهيره في HLR.
إرسال التعريف	`/api/sendIdentification`	55	استرجاع بيانات IMSI/ المصادقة من VLR السابق.
إعادة تعيين	`/api/reset`	37	إعادة تعيين initiated من HLR تجاه VLR.
استعادة البيانات	`/api/restoreData`	57	استعادة بيانات VLR يطلب المشترك من HLR.

معالجة المكالمات وبيانات المشترك

العملية	نقطة النهاية	رمز العملية	الوصف
إرسال معلومات التوجيه	`/api/sri`	22	استعلام HLR لمعلومات توجيه المكالمات الصوتية.
توفير معلومات المشترك	`/api/provideSubscriberInfo`	70	طلب حالة/موقع المشترك من VLR.
حذف بيانات المشترك	`/api/deleteSubscriberData`	8	إزالة بيانات المشترك في VLR.
استجواب في أي وقت	`/api/anyTimeInterrogation`	71	يستعلم gsmSCF عن معلومات/موقع المشترك في HLR.
استجواب بيانات الاشتراك في أي وقت	`/api/anyTimeSubscriptionInterrogation`	62	استعلام بيانات الاشتراك في HLR.
تعديل في أي وقت	`/api/anyTimeModification`	65	تعديل بيانات

العملية	نقطة النهاية	رمز العملية	الوصف
			الاشتراك في HLR.
ملاحظة تعديل بيانات المشترك	`/api/noteSubscriberDataModified`	5	يقوم HLR بإخطار gsmSCF بالبيانات المتغيرة.
ملاحظة حدث MM	`/api/noteMM-Event`	89	الإبلاغ عن حدث إدارة التنقل إلى gsmSCF.

SMS

العملية	نقطة النهاية	رمز العملية	الوصف
إرسال معلومات التوجيه لـ SM	`/api/sri-for-sm`	45	استعلام HLR عن العقدة الخادمة لتسليم SMS.
MT-Forward SM	`/api/MT-forwardSM`	44	تسليم SM منتهي إلى MSC/SGSN الخادم.
MO-Forward SM	`/api/forwardSM`	46	تقديم SM منشأ من الهاتف إلى SMSC.
إرسال SM (PDU)	`/api/sendSM`, `/api/deliverPDU`	44	بناء وتسليم PDU SMS-DELIVER.
الإبلاغ عن حالة تسليم SM	`/api/reportSM-DeliveryStatus`	47	الإبلاغ عن نتيجة تسليم SM إلى HLR.
جاهز لـ SM	`/api/readyForSM`	66	إخطار HLR بأن MS قابل للوصول لتسليم SMS.
تنبيه مركز الخدمة	`/api/alertServiceCentre`	64	تنبيه SMSC بأن مشترك متاح.

ا��مصادقة والمعدات

العملية	نقطة النهاية	رمز العملية	الوصف
إرسال معلومات المصادقة	`/api/send-auth-info`	56	استرجاع متجهات المصادقة من HLR.
إرسال IMSI	`/api/sendIMSI`	58	حل MSISDN إلى IMSI الخاص به في HLR.
تحقق من IMEI	`/api/checkIMEI`	43	استعلام حالة المعدات في EIR.

الخدمات التكميلية

العملية	نقطة النهاية	رمز العملية	الوصف
تسجيل SS	`/api/registerSS`	10	تسجيل خدمة تكميلية (مثل إعادة توجيه المكالمات / CFU).
مسح SS	`/api/eraseSS`	11	مسح خدمة تكميلية.
تفعيل SS	`/api/activateSS`	12	تفعيل خدمة تكميلية.
إلغاء تفعيل SS	`/api/deactivateSS`	13	إلغاء تفعيل خدمة تكميلية.
استجواب SS	`/api/interrogateSS`	14	استجواب حالة خدمة تكميلية.
تسجيل كلمة المرور	`/api/registerPassword`	17	تسجيل كلمة مرور SS.
الحصول على كلمة المرور	`/api/getPassword`	18	استرجاع/التح��ق من كلمة مرور SS.
إرسال إشارة التحقق من SS	`/api/forwardCheckSS-Indication`	38	تحقق غير مؤكد تجاه VLR.

خدمات الموقع (LCS) وGPRS

العملية	نقطة النهاية	رمز العملية	الوصف
إرسال معلومات التوجيه لـ LCS	`/api/sendRoutingInfoForLCS`	85	يستعلم GMLC عن HLR لتوجيه LCS.
توفير موقع المشترك	`/api/provideSubscriberLocation`	83	طلب موقع مشترك من العقدة الخادمة.
تقرير موقع المشترك	`/api/subscriberLocationReport`	86	الإبلاغ عن موقع مشترك إلى GMLC.
إرسال معلومات التوجيه لـ GPRS	`/api/sendRoutingInfoForGprs`	24	استعلام HLR عن توجيه GPRS (اختيار GGSN).
تحديث موقع GPRS	`/api/updateGprsLocation`	23	تسجيل SGSN الخادم لمشترك مع HLR.

> **نطاق LCS:** عمليات خدمات الموقع هي **صادرة (طالبة) فقط**. يقوم عميل MAP بتفعيلها *لن تستجيب لطلب* LCS لا تعمل العقدة كخادم GMLC. أو العقدة الخادمة أو HLR تجاه *SubscriberLocationReport الوارد. ProvideSubscriberLocation الوارد أو تعالج تقرير* تحمل الترميزات العناصر المعلوماتية الإلزامية بالإضافة إلى الخيارات الأساسية المذكورة أعلاه. لم يتم الكشف عن مجموعة معلمات LCS الأوسع (QoS، أشكال الأولوية، GAD، كلمة المرور، رقم المرجع) بعد.

CAP / CAMEL وحقن البيانات الخام

العملية	نقطة النهاية	الوصف
CAP InitialDP	`/api/cap/initialDP`	تفعيل حوار CAMEL تجاه gsmSCF.
CAP Connect / Continue / ReleaseCall / ApplyCharging	`/api/cap/connect`، `/api/cap/continue`، `/api/cap/releaseCall`، `/api/cap/applyCharging`	عمليات التحكم في المكالمات CAMEL. راجع دليل بوابة CAMEL.
Raw SCCP	`/api/raw/sccp`	حقن حمولة SCCP مبنية يدويًا (اختبار/تشخيص).
Raw M3UA	`/api/raw/m3ua`	حقن حمولة M3UA مبنية يدويًا (اختبار/تشخيص).

أمثلة على العمليات الشائعة

إرسال معلومات التوجيه لـ SM (SRI-for-SM)

يستعلم HLR لمعرفة كيفية معالجة جانب SMS. عن العقدة الخادمة المستخدمة لتسليم HLR لذلك، راجع SRI-for-SM في دليل HLR.

موجهًا على SRI-for-SM بشكل افتراضي، يتم إرسال **عنوان SCCP (مسار تسليم SMS).** الاحتفاظ بإدخال STP مما يتطلب من ،(المسمى SSN 6/HLR) **الوجهة MSISDN إلى GT** بدلاً من ذلك بتوجيه SMSc لهذا النطاق الرقمي. بالنسبة للمشتركين **داخل الشبكة**، يقوم GTT الذي يمكن) HLR المكون. ثم يقوم بتوجيه على عنوان النظير الخاص بـ **HLR GT** مباشرة إلى SRI الخادم الحقيقي. يتم اختيار ذلك HLR MSC لكل نطاق، ويعيد GTT لذا لا حاجة إلى ،(توجيهه بالفعل لكل وجهة:

التكوين (`:omniss7`)	الغرض	المثال
`hlr_gt_address`	GT لـ HLR المنزلي لتوجيه SRIs داخل الشبكة إليه	`"61455501111"`
`home_msisdn_prefixes`	بادئات MSISDN داخل الشبكة؛ تطابق يوجه إلى HLR GT	`["6777"]`

(`hlr_gt_address` مع تعيين) `home_msisdn_prefixes` يتم إرسال وجهة تتطابق مع إدخال GT المطلوب؛ كل شيء آخر يحتفظ بالافتراضي الموجه على SSN 6 مع HLR GT إلى MSISDN. يقبل طلب SRI-for-SM `sccp_opts` (`:called_party`, `:called_ssn`) لتجاوز الطرف المتصل SCCP لأي متصل.

نقطة النهاية: `POST /api/sri-for-sm`

الطلب:

```
{
  "msisdn": "447712345678",
  "serviceCenter": "447999123456"
}
```

الاستجابة:

```
{
  "result": {
    "imsi": "234509876543210",
    "locationInfoWithLMSI": {
      "networkNode-Number": "447999555111"
    }
  }
}
```

cURL:

```
curl -X POST http://localhost/api/sri-for-sm \
  -H "Content-Type: application/json" \
  -d '{"msisdn": "447712345678", "serviceCenter": "447999123456"}'
```

إرسال معلومات التوجيه (SRI)

نقطة النهاية: `POST /api/sri`

الطلب:

```
{
  "msisdn": "447712345678",
  "gmsc": "447999123456"
}
```

توفير رقم التجوال (PRN)

نقطة النهاية: `POST /api/prn`

الطلب:

```
{
  "msisdn": "447712345678",
  "gmsc": "447999123456",
  "msc_number": "447999555111",
  "imsi": "234509876543210"
}
```

إرسال معلومات المصادقة

نقطة النهاية: `POST /api/send-auth-info`

الطلب:

```
{
  "imsi": "234509876543210",
  "vectors": 5
}
```

الاستجابة:

```
{
  "result": {
    "authenticationSetList": [
      {
        "rand": "0123456789ABCDEF0123456789ABCDEF",
        "xres": "ABCDEF0123456789",
        "ck": "0123456789ABCDEF0123456789ABCDEF",
        "ik": "FEDCBA9876543210FEDCBA9876543210",
        "autn": "0123456789ABCDEF0123456789ABCDEF"
      }
    ]
  }
}
```

تحديث الموقع

الذي يستجيب بتسلسل إدراج بيانات المشترك. راجع HLR، الخادم لمشترك مع VLR يسجل تحديثات الموقع في دليل HLR.

نقطة النهاية: `POST /api/updateLocation`

الطلب:

```
{
  "imsi": "234509876543210",
  "vlr": "447999555111"
}
```

استجواب في أي وقت (ATI)

نقطة النهاية: `POST /api/anyTimeInterrogation`

الطلب:

```
{
  "hlr_gt": "447999123456",
  "msisdn": "447712345678"
}
```

فئات العمليات



إرسال الطلبات عبر API

باستخدام واجهة Swagger

توفر واجهة Swagger واجهة تفاعلية لكل نقطة نهاية.

1. انتقل إلى `http://your-server/swagger`.
2. اختر نقطة نهاية (مثل `/api/sri-for-sm`) وانقر على **جربها**.
3. املأ جسم الطلب وانقر على **تنفيذ**.
4. يتم عرض استجابة SS7 المفككة أدناه.

المستند الخام لـ OpenAPI متاح على `http://your-server/swagger.json`.

رموز استجابة API

الرمز	المعنى
200	نجاح: النتيجة المفككة موجودة في جسم الاستجابة.
400	طلب غير صحيح: معلمات غير صالحة أو مفقودة.
504	مهلة الطلب (الافتراضي 10 ثواني) ضمن مهلة SS7 مهلة ال��وابة: لا توجد استجابة.

بعض العمليات غير مؤكدة (إرسال ونسيان) وتعود على الفور بمجرد إرسال الرسالة.

مقاييس عميل MAP

تُصدّر جميع المقاييس على نقطة نهاية Prometheus (`/metrics`).

النوع: عداد **الوصف:** إجمالي طلبات MAP المرسلة. **المقياس:** `map_requests_total` **التسميات:** `operation`: على سبيل المثال `sri`, `sri_for_sm`, `prn`, `authentication_info`, `update_location`.

النوع: عداد **الوصف:** إجمالي طلبات MAP التي نتج عنها خطأ أو مهلة. **المقياس:** `map_request_errors_total` **التسميات:** `operation`.

النوع: هيستوجرام **الوصف:** توزيع أوقات جولة طلبات MAP. **المقياس:** `map_request_duration_milliseconds` **التسميات:** `operation`.

النوع: مقياس **الوصف:** طلبات MAP التي تنتظر حاليًا استجابة. **المقياس:** `map_pending_requests`

أمثلة على استعلامات Prometheus

```
# طلبات SRI-for-SM في الساعة الماضية
increase(map_requests_total{operation="sri_for_sm"}[1h])

# متوسط وقت استجابة SRI
rate(map_request_duration_milliseconds_sum{operation="sri"}[5m])
  / rate(map_request_duration_milliseconds_count{operation="sri"}
[5m])

# معدل الأخطاء لكل عملية
sum(rate(map_request_errors_total[5m])) by (operation)

# الطلبات المعلقة الحالية
map_pending_requests
```

استكشاف الأخطاء وإصلاحها في عميل MAP

انتهاء مهلة الطلبات (504)

الأعراض: تعيد API 504 من HLR/MSC؛ لا توجد استجابة.

الأسباب المحتملة:

- ارتباط M3UA إلى STP/SGP ليس نشطًا.
- لا يوجد مسار لـ STP إلى عنوان Global Title الوجهة.
- عنوان SCCP غير صحيح (GT للأطراف المتصلة/المتصل بها، SSN) أو سياق التوجيه.

الحل:

1. تأكد من حالة ارتباط M3UA على لوحة المعلومات أو عبر دليل STP لعرض الحالة.
2. تحقق من الاتصال الشبكي (SCTP) إلى STP.
3. تأكد من أن عنوان Global Title الوجهة يمكن توجيهه وأن النظام الفرعي الوجهة (مثل HLR SSN 6) في الخدمة.

4. راجع السجلات للرسائل المرتجعة من SCCP (انظر أدناه).

رسائل العودة / الخطأ من SCCP

الأعراض: تظهر السجلات رسائل العودة لخدمة وحدة SCCP (UDTS).

الأسباب الشائعة للعودة:

- **لا يوجد ترجمة للعنوان**: عنوان Global Title غير موجود في جدول توجيه STP.
- **فشل النظام الفرعي**: النظام الفرعي الوجهة (مثل HLR SSN 6) غير متاح.
- **فشل الشبكة / الازدحام**: مشكلة مؤقتة في الشبكة.

الحل:

- تأكد من تكوين توجيه/ترجمة STP لعنوان GT الوجهة.
- تحقق من أن النظام الفرعي الوجهة يعمل.
- أعد المحاولة بعد انقضاء الازدحام.

الوثائق ذات الصلة

- ← العودة إلى الوثائق الرئيسية
- مرجع التكوين: جميع معلمات التكوين
- دليل الميزات الشائعة، API، المراقبة: واجهة الويب
- دليل STP: التوجيه وترجمة Global Title
- دليل HLR: معالجة العمليات من جانب HLR
- دليل مركز SMS: تسليم SMS
- دليل بوابة CAMEL: عمليات CAP/CAMEL

OmniSS7 بواسطة Omnitouch Network Services

دليل تكوين مركز الرسائل القصيرة (SMSc)

← العودة إلى الوثائق الرئيسية

يوفر هذا الدليل تكوينًا مفصلًا لاستخدام OmniSS7 كواجهة أمامية لمركز الرسائل القصيرة (SMSc) مع **OmniMessage** كمنصة تخزين الرسائل والتسليم الخلفية.

تكامل OmniMessage

يعمل وضع OmniSS7 SMSc كواجهة أمامية لإشارات SS7 تتفاعل مع **OmniMessage**، وهي منصة رسائل SMS بمستوى الناقل. تفصل هذه البنية بين الاهتمامات بمستوى الناقل:

- **OmniSS7 (واجهة SMSc الأمامية)**: تتعامل مع جميع إشارات بروتوكول SS7/MAP، وتوجيه SCCP، والتواصل الشبكي
- **OmniMessage (الخلفية SMS)**: تدير تخزين الرسائل، والتخزين المؤقت، ومنطق إعادة المحاولة، وتتبع التسليم، وقرارات التوجيه

لماذا OmniMessage؟

يوفر OmniMessage قدرات رسائل SMS بمستوى الناقل مع ميزات تشمل:

- **إدارة قائمة الانتظار للرسائل**: تخزين دائم مع منطق إعادة المحاولة القابل للتكوين وأولوية التخزين المؤقت
- **تتبع التسليم**: حالة التسليم في الوقت الحقيقي، وتقارير التسليم (DLR)، وتتبع أسباب الفشل
- **دعم متعدد لمراكز الرسائل القصيرة**: يمكن لعدة مثيلات أمامية الاتصال بخلفية OmniMessage واحدة لتحقيق التوازن في الحمل والموثوقية
- **ذكاء التوجيه**: قواعد توجيه متقدمة بناءً على الوجهة، والمرسل، ومحتوى الرسالة، ووقت اليوم
- **تحديد المعدلات**: تحكم TPS (المعاملات في الثانية) لكل مسار لمنع الازدحام الشبكي
- **تصميم يعتمد على API أولاً**: واجهة برمجة تطبيقات HTTP REST للتكامل مع أنظمة الفوترة، وبوابات العملاء، والتطبيقات الخارجية

- **التحليلات والتقارير**: إحصائيات حجم الرسائل، ومعدلات نجاح التسليم، ومقاييس الأداء

تُخ��ن جميع بيانات الرسائل، وحالة التسليم، وتكوينات التوجيه وتُدار في OmniMessage. يستعلم OmniSS7 من OmniMessage عبر استدعاءات API عبر HTTPS لاسترداد الرسائل المعلقة، وتحديث حالة التسليم، والتسجيل كواجهة أمامية نشطة.

هو **واجهة أمامية للإشارات فقط**. يتم التعامل مع جميع منطق OmniSS7 SMSc **مهم**: وضع توجيه الرسائل، وإدارة قوائم الانتظار، وخوارزميات إعادة المحاولة، وتتبع التسليم، والقواعد التجارية بواسطة OmniMessage. يغطي هذا الدليل تكوين بروتوكول SS7/MAP في OmniSS7. لمزيد من المعلومات حول توجيه الرسائل، وتكوين القوائم، وتقارير التسليم، وتحديد المعدلات، والتحليلات، **يرجى الرجوع إلى وثائق OmniMessage**.

جدول المحتويات

1. تكامل OmniMessage
2. ما هو وضع مركز الرسائل القصيرة؟
3. تمكين وضع SMSc
4. تكوين واجهة برمجة التطبيقات HTTP
5. تدفقات رسائل SMS
6. معالجة مركز الخدمة التنبيهية
7. منع الحلقات
8. تتبع المشتركين في SMSc
9. تخزين عنوان VLR
10. تكوين التفريغ التلقائي
11. المقاييس والمراقبة
12. استكشاف الأخطاء وإصلاحها

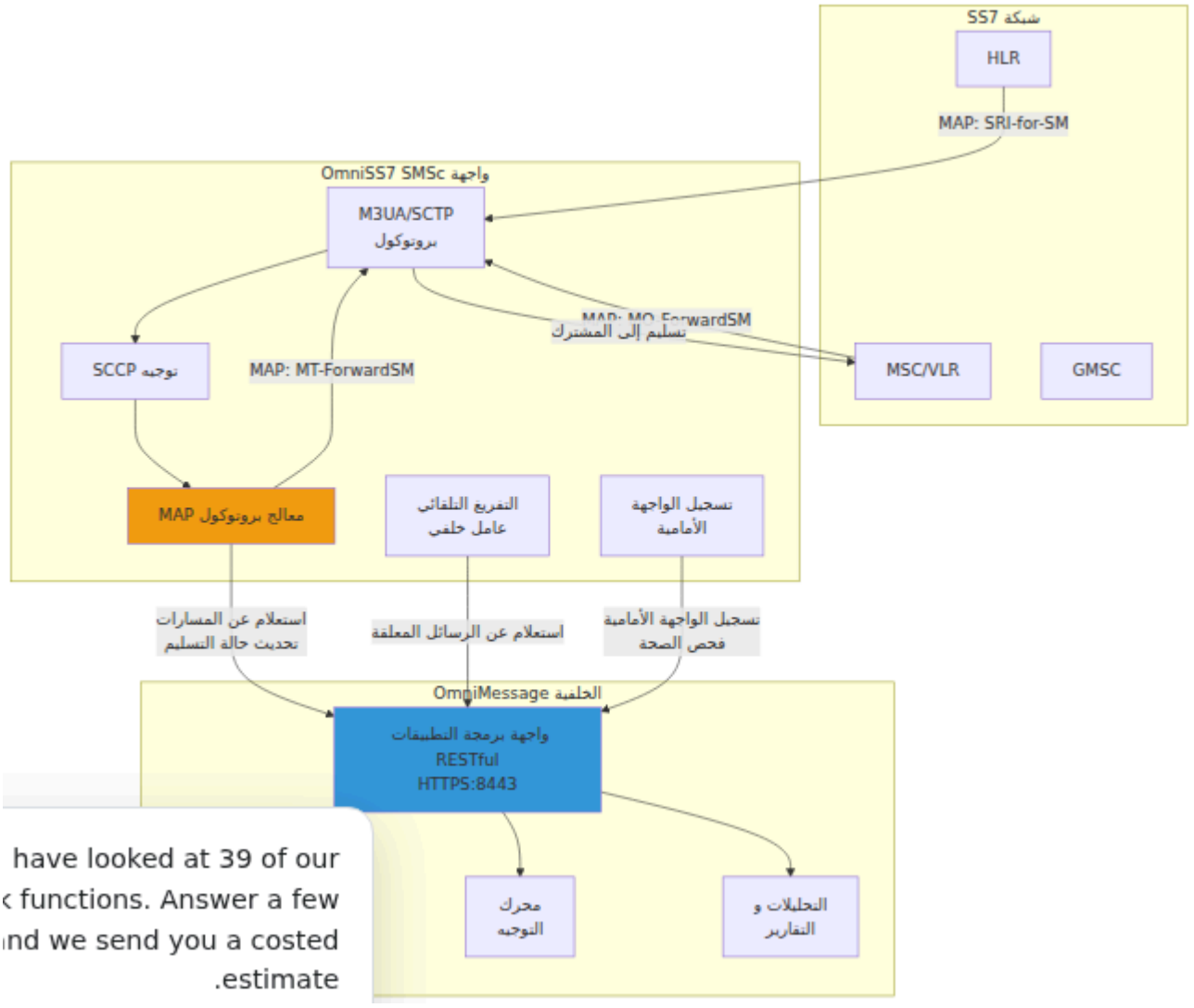
ما هو وضع مركز الرسائل القصيرة؟

ملاحظة: يغطي هذا القسم تكوين إشارات SS7 لـ OmniSS7. لمزيد من المعلومات حول قواعد توجيه الرسائل، وإدارة قوائم الانتظار، وتتبع التسليم، وتكوين المنطق التجاري، انظر **وثائق منتج OmniMessage**.

يمكّن **وضع مركز الرسائل القصيرة** OmniSS7 من العمل كـ SMSc لـ:

- **تسليم MT-SMS**: تسليم الرسائل القصيرة إلى المشتركين
- **معالجة MO-SMS**: استقبال الرسائل القصيرة المنشأة من الهاتف المحمول وتوجيهها
- **تخزين الرسائل في قائمة الانتظار**: قائمة انتظار الرسائل المدعومة بقاعدة بيانات مع منطق إعادة المحاولة
- **التفريغ التلقائي**: تسليم الرسائل القصيرة تلقائيًا من قائمة الانتظار
- **تقارير التسليم**: تتبع حالة تسليم الرسائل

بنية مركز الرسائل القصيرة



have looked at 39 of our
k functions. Answer a few
nd we send you a costed
.estimate

تمكين وضع SMSc

يمكن أن يعمل OmniSS7 في أوضاع مختلفة (STP، HLR، SMSc). يتم اختيار الوضع من خلال ملف التكوين، `config/config.exs` مجموعة من علامات الوضع والتكوين المرتبط في الافتراضي للتطبيق.

التبديل إلى وضع SMSc

لتشغيل OmniSS7 كـ SMSc، قم بتعيين علامات الوضع بحيث يتم تمكين ميزات SMSc، ثم أضف تكوين SMSc المحدد:

1. **افتح** `config/config.exs`
2. قم **بتعيين** علامات الوضع تحت `config :omniss7`:
   - `map_client_enabled: true`: مطلوب لتمكين قدرات التوجيه/MAP
   - `smsc_mode_enabled: true`: يمكّن معالجة SMSc المحددة
   - `hlr_mode_enabled: false`: يعطل ميزات HLR
3. **أضف** معلمات تكوين SMSc (نقطة نهاية API، اسم SMSc، عنوان GT، إعدادات التفريغ التلقائي، اتصال M3UA) الموضحة أدناه
4. **خصص** المعلمات لنشر تطبيقك
5. **أعد تشغيل** التطبيق لتصبح التغييرات سارية

> **ملاحظة**: `config/runtime.exs` هو نموذج بسيط يستخدم للاختبار الآلي فقط. لا يحتوي على تعريفات وضع التشغيل. جميع تكوينات الوضع و SMSc موجودة في `config/config.exs` (مع تجاوزات خاصة بالبيئة في ملفات مثل `config/test.exs`). توفر عمليات النشر الإنتاجية ملفها الخاص `config/config.exs`.

تكوين وضع SMSc

يبدو تكوين SMSc الكامل (في `config/config.exs`) كما يلي:

```
config :omniss7,
  # تمكين ميزات الوضع - علامات الوضع STP + SMSc
  # ملاحظة: map_client_enabled هو true لأن SMSc يحتاج إلى قدرات التوجيه
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: true,

  # تكوين واجهة برمجة التطبيقات الخلفية OmniMessage
  smsc_api_base_url: "https://omnimessage.example.com:8443",
  # تعريف SMSc للتسجيل مع الخلفية
  smsc_name: "ipsmgw",
  # عنوان مركز الخدمة GT لعمليات SMS
  smsc_service_center_gt_address: "5551234567",

  # تكوين التفريغ التلقائي (معالجة قائمة انتظار SMS في الخلفية)
  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: "ipsmgw",
  auto_flush_tps: 10,

  # تكوين اتصال M3UA
  # الاتصال كـ ASP لإرسال/استقبال عمليات MAP SMS
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :stp_client_asp,
    # نقطة النهاية المحلية (نظام SMSc)
    local_ip: {10, 179, 4, 12},
    local_port: 2905,
    # نقطة النهاية البعيدة STP
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }
```

تُسجل صفحات واجهة الويب بشكل منفصل تحت `config :control_panel` في `config/config.exs`. تسجل قائمة `use_additional_pages` جميع صفحات الافتراضية بتصفية الصفحات التي لا تنطبق على الوضع النشط.(إدخالًا 16~) OmniSS7 يقوم وقت التشغيل ؛ لاحظ أن صفحة حالة النظير تُسمى **"Peers"**:

```
config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "أحداث SS7"},
    {SS7.Web.TestClientLive, "/client", "عميل SS7"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "Peers"},
    {SS7.Web.SCTPConnectionsLive, "/sctp", "اتصالات SCTP"},
    {SS7.Web.RoutingLive, "/routing", "التوجيه"},
    {SS7.Web.RoutingTestLive, "/routing_test", "اختبار التوجيه"},
    {SS7.Web.SmscLinksLive, "/smsc_links", "روابط SMSc"},
    {SS7.Web.SmscSubscribersLive, "/smsc_subscribers", "مشتركي
SMSc"},
    {SS7.Web.HlrLinksLive, "/hlr_links", "روابط HLR"},
    # ... صفحات إضافية (المشتركين النشطين، CAMEL، اختبار السجلات،
MSISDN/IMSI، إلخ.)
  ],
  page_order: [
    "/application", "/configuration", "/events", "/logs",
"/client", "/m3ua",
    "/routing", "/routing_test", "/smsc_links",
"/smsc_subscribers", "/hlr_links",
    # ... المسارات المتبقية المرتبة
  ]
```

معلمات التكوين للتخصيص

للحصول على مرجع كامل لجميع معلمات التكوين، انظر مرجع التكوين.

المعلمة	النوع	الافتراضي	صف
`smsc_api_base_url`	سلسلة	*مطلوب*	**ة برمجة طبيقات** الخلفية لـ OmniMe
`smsc_name`	سلسلة	`"{hostname}_SMSc"`	SM معرف لخاص بك للتسجيل
`smsc_service_center_gt_address`	سلسلة	*مطلوب*	GT عنوان الخدمة
`auto_flush_enabled`	Boolean	`true`	ن معالجة ة الانتظار التلقائية
`auto_flush_interval`	عدد صحيح	`10_000`	رة معالجة ة الانتظار ملي ثانية
`auto_flush_dest_smsc`	سلسلة	`nil`	فلتر اسم هة للتفريغ) التلقائي لجة جميع ل المعلقة
`auto_flush_tps`	عدد صحيح	`10`	ل معالجة الرسائل لات/ثانية)
`local_ip`	مجموعة	*مطلوب*	IP عنوان ص SMSc بك

المعلمة	النوع	الافتراضي	صف
`local_port`	عدد صحيح	`2905`	المحلي لـ SCTP
`remote_ip`	مجموعة	*مطلوب*	IP عنوان للاتصال بـ
`remote_port`	عدد صحيح	`2905`	ذ البعيد لـ SCTP
`routing_context`	عدد صحيح	`1`	رف سياق التوجيه M

ماذا يحدث عند تمكين وضع SMSc

عند تعيين `smsc_mode_enabled: true` و `map_client_enabled: true`، ستظهر واجهة الويب:

- **أحداث SS7** - تسجيل الأحداث
- **عميل SS7** - اختبار عمليات MAP
- **Peers** - حالة الاتصال (نظائر M3UA/SCTP)
- **التوجيه** - إدارة جدول التوجيه (تم تمكين STP)
- **اختبار التوجيه** - اختبار التوجيه (تم تمكين STP)
- **روابط SMSc** - حالة واجهة برمجة التطبيقات SMSc + إدارة قائمة انتظار SMS ← *محدد لـ SMSc*
- **الموارد** - مراقبة النظام
- **التكوين** - عارض التكوين

روابط ستكون علامة تبويب **HLR** مخفية.

ملاحظات مهمة

- يتطلب وضع SMSc `map_client_enabled: true` لتمكين قدرات التوجيه
- **الخلفية OmniMessage**: يجب أن تكون واجهة برمجة التطبيقات الخلفية لـ OmniMessage قابلة للوصول عند المكون `smsc_api_base_url`

- كل OmniMessage **تسجيل الواجهة الأمامية**: يقوم النظام تلقائيًا بالتسجيل مع `SMS.FrontendRegistry` عبر وحدة (مللي ثانية 85,000 = `@registration_interval`) **~85 ثانية**
- **مهلة طلب API**: جميع طلبات واجهة برمجة التطبيقات لـ OmniMessage لها **مهلة ثابتة مدتها 5 ثوانٍ**
- **مهلة طلب MAP**: تحتوي SRI-for-SM على مهلة حوار مدتها 10 ثوانٍ. تحتوي MT-ForwardSM البطيئة VLR على **مهلة حوار مدتها 30 ثانية** لاستيعاب استجابات G/CS على الشبكات 2
- في الخلفية تلقائيًا SMS يقوم التفريغ التلقائي بمعالجة قائمة انتظار
- MAP SMS لإرسال/استقبال عمليات STP بـ M3UA يتطلب اتصال
- بعد تغيير الأوضاع، يجب عليك إعادة تشغيل التطبيق لتصبح التغييرات سارية
- **واجهة الويب**: انظر دليل واجهة الويب لمعلومات حول استخدام الواجهة الويب
- **الوصول إلى API**: انظر دليل API لوثائق واجهة برمجة التطبيقات REST والوصول إلى واجهة Swagger

---

تكوين واجهة برمجة التطبيقات HTTP

إعداد الخلفية OmniMessage

يتواصل OmniSS7 مع OmniMessage عبر واجهة برمجة التطبيقات REST HTTPS لإدارة تسليم الرسائل، وتتبع حالة المشترك، والتسجيل كواجهة أمامية نشطة:

```
config :omniss7,
  # عنوان URL الأساسي لواجهة برمجة التطبيقات OmniMessage
  smsc_api_base_url: "https://omnimessage.example.com:8443",
  # معرف اسم SMSC للتسجيل (يستخدم hostname_SMSc إذا كان فارغًا)
  smsc_name: "omni-smsc01",
  # عنوان GT لمركز الخدمة لعمليات SMS
  smsc_service_center_gt_address: "5551234567"
```

معلمات التكوين:

المعلمة	النوع	مطلوب	الافتراضي
`smsc_api_base_url`	سلسلة	نعم	`"https://localhost`
`smsc_name`	سلسلة	لا	`""` (يستخدم `"{hostname}_SMSc"`)
`smsc_service_center_gt_address`	سلسلة	لا	`"5551234567"`

تسجيل الواجهة الأمامية

عند بدء التشغيل و**إعادة التسجيل** OmniMessage ��يقوم النظام تلقائيًا بتسجيل نفسه م
OmniMessage يسمح ذلك لـ `SMS.FrontendRegistry`. **كل ~85 ثانية** عبر وحدة بـ:

- تتبع الواجهات الأمامية النشطة لتحقيق التوازن في الحمل
- مراقبة وقت التشغيل وحالة الصحة
- جمع معلومات التكوين
- إدارة توجيه SMS الموزع عبر واجهات أمامية متعددة

تفاصيل التنفيذ:

- **فترة التسجيل**: ~85 ثانية (`@registration_interval 85_000`، ثابتة)
- **العملية**: تبدأ تلقائيًا عند `smsc_mode_enabled: true`

حمولة التسجيل:

```
{
  "frontend_name": "omni-smsc01",
  "configuration": "{...}",
  "frontend_type": "SS7",
  "hostname": "smsc-server01",
  "uptime_seconds": 12345
}
```

ملاحظة: يتم أخذ اسم الواجهة الأمامية من معلمة تكوين `smsc_name`. إذا لم يتم تعيينه، فإنه يتراجع إلى `"{hostname}_SMSc"`.

التواصل مع واجهة برمجة التطبيقات OmniMessage

عندما يتلقى OmniSS7 عمليات MAP من شبكة SS7 أو يعالج قائمة انتظار الرسائل، يتواصل مع OmniMessage لـ:

- **التسجيل كو��جهة أمامية نشطة** والإبلاغ عن حالة الصحة
- **إرسال الرسائل المنشأة من الهاتف المحمول (MO)** المستلمة من المشتركين
- **استرداد الرسائل الموجهة إلى الهاتف المحمول (MT)** من قائمة الانتظار للتسليم
- **تحديث حالة التسليم** مع تقارير النجاح/الفشل
- **استعلام معلومات التوجيه** لتوجيه الرسائل

نقطة النهاية	الطريقة	الغرض	م الطلب
`/api/frontends`	POST	تسجيل مثيل الواجهة الأمامية	`{"frontend_nam "configuration "frontend_type "hostname": ". "uptime_second`
`/api/messages_raw`	POST	إدراج رسالة SMS جديدة (TPDU الخام)	`{"rp_originato "...", "rp_destinatio "...", "direct "message_body" "send_time": " "smsc_node_nam "source_smsc": "source_type":`
`/api/messages`	GET	الحصول على قائمة انتظار الرسائل	رأس: `smsc: <sm`
`/api/messages/{id}`	PATCH	وضع علامة على الرسالة كتم تسليمها	`{"deliver_time "dest_smsc": "`
`/api/messages/{id}`	PUT	تحديث حالة الرسالة	`{"dest_smsc":`
`/api/locations`	POST	إدراج/تحديث موقع المشترك	`{"msisdn": ".. "...", "locati "ims_capable": "csfb": false, "...", "user_a "...", "ran_lo "...", "imei": "registered":`
`/api/events`	POST	إضافة تتبع الأحداث	`{"message_id": "name": "...",`

نقطة النهاية	الطريقة	الغرض	ل الطلب
			`"description":`
`/api/status`	GET	المسار قابل) فحص الصحة للتكوين عبر `smsc_api_status_path`، الافتراضي `/api/status`؛ يتم استعلامه بواسطة صفحة وليس بواسطة SMSc، روابط `SMSc.APIClient`)	-

API تنسيق استجابة

تستخدم جميع استجابات API تنسيق JSON مع الاتفاقيات التالية:

- **استجابات النجاح**: HTTP 200-201 مع جسم JSON يحتوي على بيانات النتيجة
- **استجابات الخطأ**: HTTP 4xx/5xx مع تفاصيل الخطأ في جسم الاستجابة
- **الطوابع الزمنية**: تنسيق ISO 8601 (مثل، `T12:34:56Z"` `21-10-2025"`)
- **معرفات الرسائل**: معرفات عددية أو نصية

API وحدات عميل

يتكون نظام SMS من ثلاث وحدات رئيسية:

1. SMSc.APIClient

وحدة عميل API الرئيسية التي توفر جميع الاتصالات عبر واجهة برمجة التطبيقات HTTP مع OmniMessage:

- `frontend_register/4` - تسجيل الواجهة الأمامية مع OmniMessage
- `insert_message/1` - إدراج رسالة SMS خام (تأخذ خريطة رسالة واحدة؛ تنشر إلى `/api/messages_raw`)
- `insert_location/9` - إدراج/تحديث بيانات موقع المشترك
- `get_message_queue/2` - استرداد الرسائل المعلقة من قائمة الانتظار
- `mark_dest_smsc/3` - وضع علامة على الرسالة كتم تسليمها أو فشلت

- `add_event/3` - إضافة تتبع الأحداث للرسائل
- `flush_queue/2` - معالجة الرسائل المعلقة (SRI-for-SM + MT-forwardSM)
- `auto_flush/2` - حلقة معالجة قائمة الانتظار المستمرة

2. SMS.FrontendRegistry

يتعامل مع التسجيل الدوري للواجهة الأمامية مع الخلفية:

- يسجل تلقائيًا عند بدء التشغيل
- يعيد التسجيل كل ~85 ثانية
- يستخدم `smsc_name` (يتراجع إلى اسم المضيف) من التكوين
- يجمع معلومات التكوين ووقت التشغيل للنظام

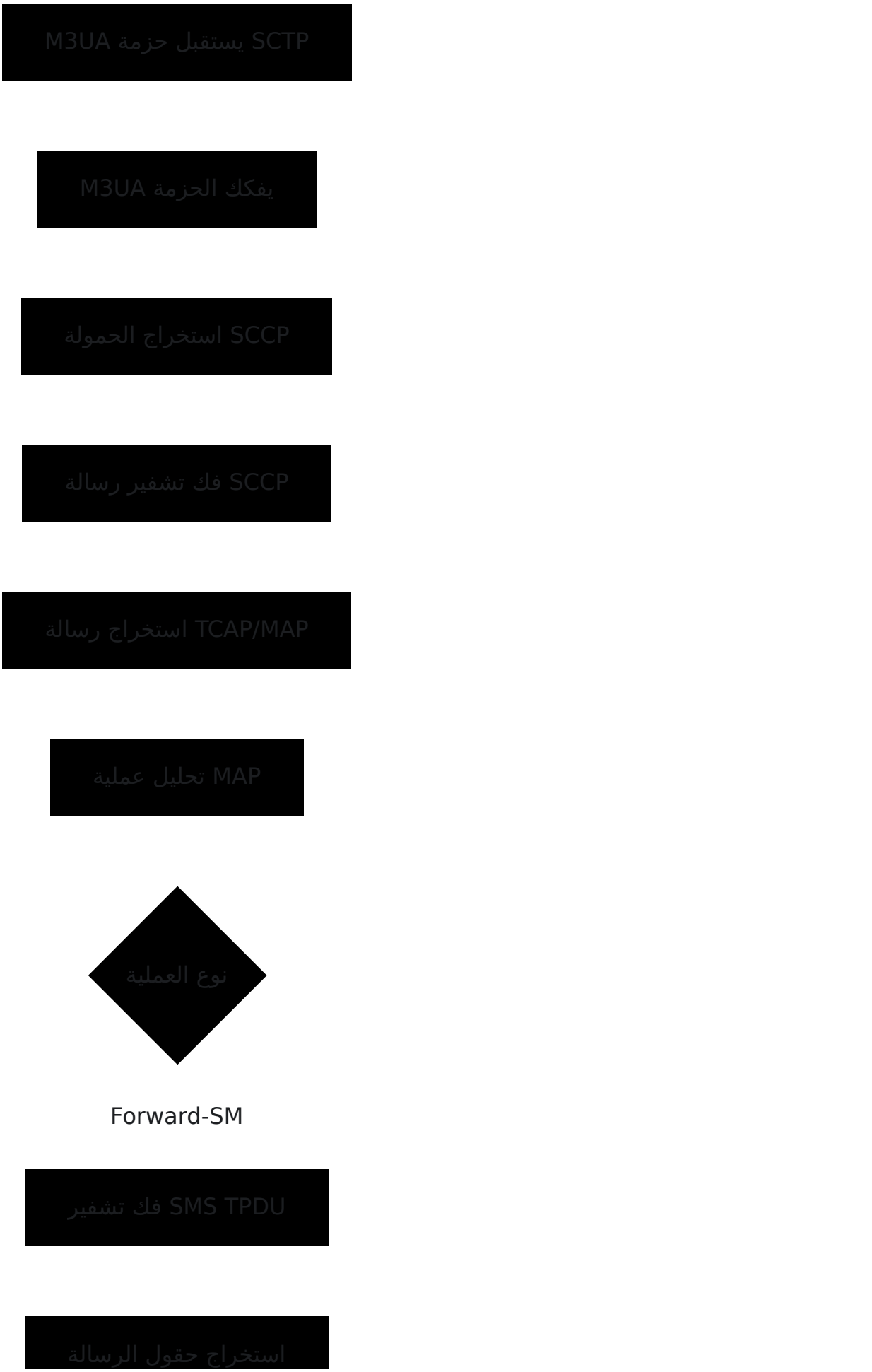
3. SMS.Utils

وظائف مساعدة لعمليات SMS:

- `generate_tp_scts/0` - توليد الطابع الزمني للرسالة القصيرة بتنسيق TPDU

تدفقات رسائل SMS

تدفق SMS (منشأ من الهاتف المحمول) الوارد


SCTP يستقبل حزمة M3UA
M3UA يفكك الحزمة
SCCP استخراج الحمولة
SCCP فك تشفير رسالة
TCAP/MAP استخراج رسالة
MAP تحليل عملية
نوع العملية
Forward-SM
SMS TPDU فك تشفير
استخراج حقول الرسالة





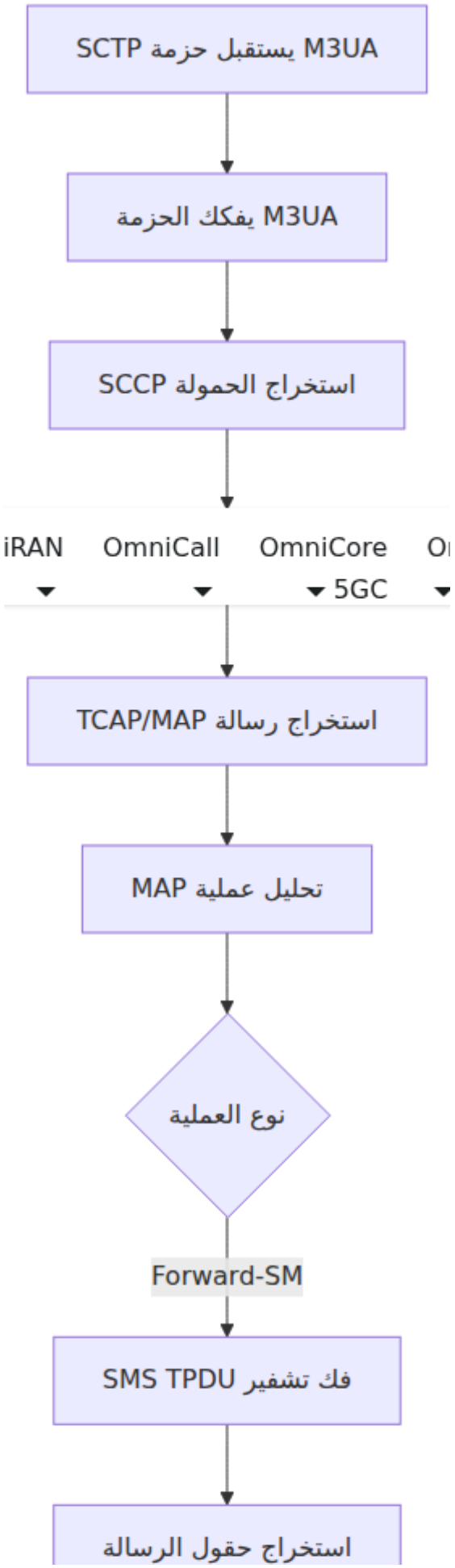


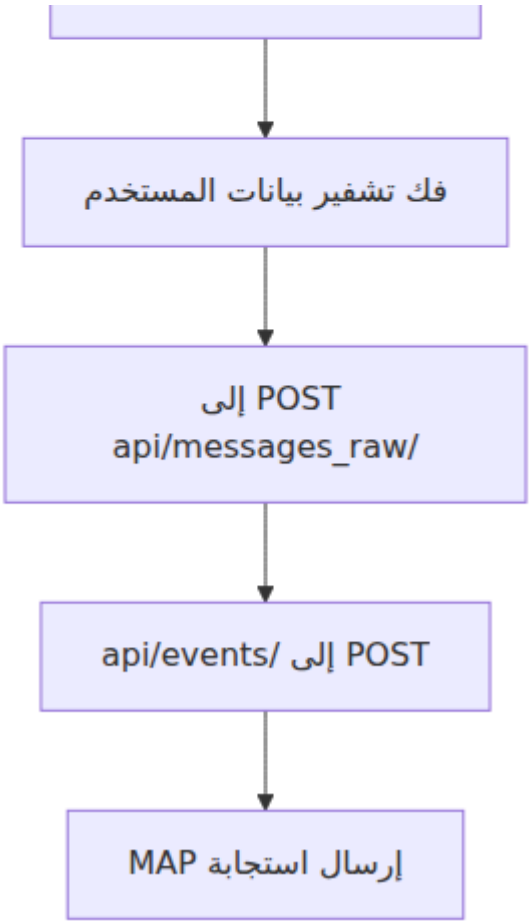




الصادر (موجه إلى الهاتف المحمول) SMS تدفق

عندما يتم تكوين واجهة برمجة التطبيقات HSS/HLR (`hlr_api_base_url`)، يمكن لـ IP-SM-GW تسليم MT-SMS دون جولة من خل��ل SRI-for-SM إلى **المشتركين على الشبكة** الحقيقي للمشترك وعنوان IMSI مباشرة للحصول على HSS استعلام واجهة برمجة التطبيقات في متتبع المشترك عمليات البحث المتكررة **VLR** الخادم. يتجنب **ذاكرة التخزين المؤقت** VLR لنفس المشترك في HSS.


M3UA يستقبل حزمة SCTP
M3UA يفكك الحزمة
SCCP استخراج الحمولة
iRAN
OmniCall
OmniCore
5GC
TCAP/MAP استخراج رسالة
MAP تحليل عملية
نوع العملية
Forward-SM
SMS TPDU فك تشفير
استخراج حقول الرسالة



شرح الخطوات الرئيسية:

- **فحص ذاكرة التخزين المؤقت VLR**: قبل استعلام واجهة برمجة التطبيقات HSS، يتحقق النظام من ذاكرة التخزين المؤقت VLR في متتبع المشترك للعثور على إدخال صالح (IMSI + عنوان VLR ضمن TTL). يتخطى نجاح الذاكرة التخزينية واجهة برمجة التطبيقات HSS تمامًا. انظر تخزين عنوان VLR للحصول على التفاصيل.

- **تسليم مباشر عبر واجهة برمجة التطبيقات HSS (على الشبكة)**: عندما يتم تكوين `hlr_api_base_url`، يستعلم IP-SM-GW واجهة برمجة التطبيقات HSS الوجهة للحصول على MSISDN باستخدام **IMSI وعنوان VLR الحقيقي** للمشترك. إذا كانت كلاهما متاحة، يرسل IP-SM-GW MT-ForwardSM مباشرة إلى **الخادم**. يتجاوز هذا تمامًا SRI-for-SM.

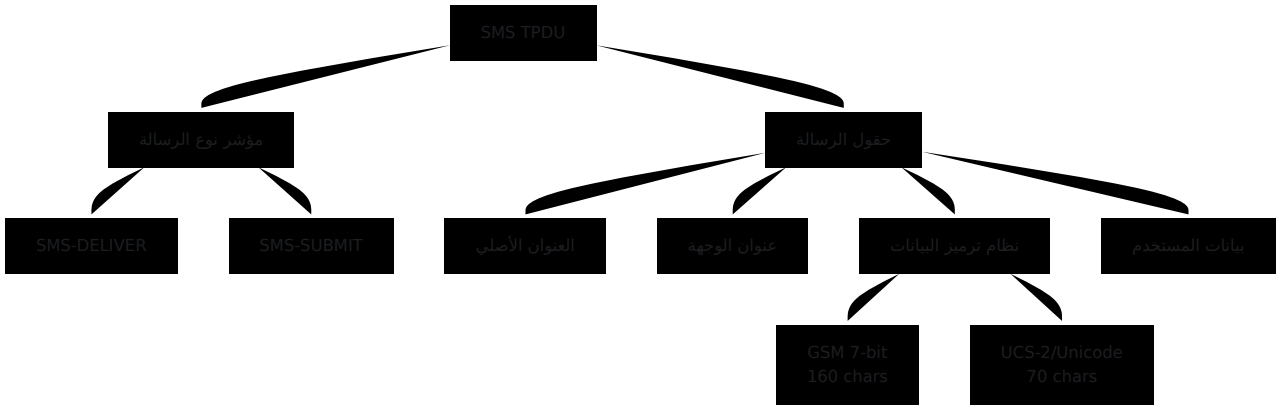
- **مشترك غائب (على الشبكة)**: إذا وجدت واجهة برمجة التطبيقات HSS المشترك خادم، يتم وضع علامة على الرسالة كفاشلة مع إدخال موقع VLR ولكن لم يتم تعيين أي منتهي الصلاحية. يتم إعادة محاولات التسليم عندما يرسل HLR alertServiceCenter بعد إعادة تسجيل المشترك.

- **تراجع SRI-for-SM (خارج الشبكة)**: عندما لا يتم تكوين واجهة برمجة التطبيقات HSS، يتم استخدام تدفق SRI-for-SM أو لم يتم العثور على المشترك في HSS

صناعي ورقم عقدة الشبكة. انظر دليل IMSI برقم HLR القياسي. يستجيب SRI-for-SM في HLR.

- **مسح VLR عند الفشل**: عندما يفشل تسليم MT-ForwardSM (انتهاء المهلة، خطأ SCCP، إلخ)، يتم مسح عنوان VLR المخزن حتى يتم إعادة استعلام واجهة برمجة التطبيقات HSS للحصول على VLR جديد.
- **انتهاء مهلة MT-ForwardSM**: يحتوي حوار MT-ForwardSM على **مهلة قدرها 30 ثانية** لاستيعاب استجابات VLR 2 من البطيئةIG/CS.

هيكل SMS TPDU



معالجة مركز خدمة التنبيه

لتتبع حالة قابلية الوصول من HLR **alertServiceCenter** تلقي رسائل SMSc يمكن لـ للمشتركين.

راجع دليل تكامل alertServiceCenter، للحصول على معلومات حول كيفية إرسال رسائل HLR لـ مركز خدمة التنبيه في HLR.

ما هو alertServiceCenter؟

عندما يقوم المشترك بإجراء UpdateLocation في HLR (مع VLR/MSC جديد)، أي، التسجيل بأن المشترك أصبح الآن قابلًا للوصول عن طريق إرسال رسالة SMSc إبلاغ أنظمة HLR يمكن لـ **alertServiceCenter** (عملية MAP 64).

التكوين

يتم تكوين وقت انتهاء صلاحية الموقع في HLR:

```
config :omniss7,
  # وقت انتهاء صلاحية الموقع عندما يتلقى SMSc alertServiceCenter
(الافتراضي: 48 ساعة)
  hlr_alert_location_expiry_seconds: 172800
```

السلوك

عندما يتلقى SMSc رسالة alertServiceCenter:

1. **فك تشفير MSISDN**: استخراج MSISDN الخاص بالمشترك من الرسالة (تنسيق TBCD)
2. **إزالة بادئة TON/NPI**: إزالة البادئات الشائعة مثل "19"، "11"، "91" (على سبيل المثال، "19123123213" → "123123213")
3. **حل IMSI**: إذا تم تكوين `hlr_api_base_url`، استعلام واجهة برمجة التطبيقات HSS أو لم يتم تكوين HSS **الحقيقي** للمشترك. إذا فشل البحث في **IMSI** لحل HSS اصطناعي باستخدام نفس التعيين IMSI واجهة برمجة التطبيقات، يتم الرجوع إلى توليد IMSI الحقيقي أمرًا حيويًا للتسليم على الشبكة للمشتركين من مثل SRI-for-SM. يعتبر MT-ForwardSM. يتطلب MSC/VLR IMSI الصحيح في 2G/CS.
4. **POST إلى /api/locations**: تحديث قاعدة بيانات الموقع بـ:
   - `msisdn`: رقم هاتف المشترك (منظف)
   - `imsi`: IMSI الاصطناعي
   - `location`: اسم SMSc (على سبيل المثال، "ipsmgw")
   - `expires`: الوقت الحالي + `hlr_alert_location_expiry_seconds`
   - `csfb`: true (المشترك قابل للوصول عبر Circuit-Switched Fallback)
   - `ims_capable`: false (هذا تسجيل 2G/3G CS، وليس IMS/VoLTE)
   - `user_agent`: HLR GT (للتتبع) الذي أرسل التنبيه
   - `ran_location`: "SS7"
5. **التتبع في متتبع المشتركين SMSc**: تسجيل المشترك مع HLR GT، الحالة=نشط، عدادات الرسائل عند 0
6. **إرسال ACK**: الرد على HLR بإقرار alertServiceCenter

معالجة المشترك الغائب

عندما يحاول SMSc أثناء تسليم رسالة ويتلقى خطأ "مشترك غائب" SRI-for-SM (للمزيد حول SRI-for-SM، راجع SRI-for-SM في دليل HLR):

1. تعيد SRI-for-SM خطأ `absentSubscriberDiagnosticSM` **كشف الغياب**:
2. **انتهاء صلاحية الموقع**: POST إلى /api/locations مع `expires=0` لتحديد المشترك على أنه غير قابل للوصول
3. **وكيل المستخدم**: تعيينه إلى "SS7_AbsentSubscriber" لتحديد المصدر
4. SMSc **تحديث المتتبع**: وضع علامة على المشترك على أنه `فشل` في متتبع مشتركين

هذا يضمن أن تعكس قاعدة بيانات الموقع والمتتبع بدقة حالة قابلية الوصول للمشتركين.

مخطط التدفق



نقطة نهاية واجهة برمجة التطبيقات

POST /api/locations

```
{
  "msisdn": "15551234567",
  "imsi": "001010123456789",
  "location": "ipsmgw",
  "ims_capable": false,
  "csfb": true,
  "expires": "2025-11-01T12:00:00Z",
  "user_agent": "15551111111",
  "ran_location": "SS7",
  "imei": "",
  "registered": "2025-10-30T12:00:00Z"
}
```

ملاحظة: يحتوي حقل `user_agent` على HLR GT الذي أرسل alertServiceCenter، مما يسمح لـ SMSc بتتبع HLR الذي يقدم تحديثات الموقع.

بالنسبة للمشتركين الغائبين، يتم تعيين `expires` (انتهاء فوري) إلى الوقت الحالي.

منع الحلقات

تنفذ SMSc **منع الحلقات التلقائي** لتجنب حلقات توجيه الرسائل اللانهائية عندما تنشأ الرسائل من شبكات SS7، مع السماح بتسليم MO→MT على الشبكة الشرعي بين المشتركين.

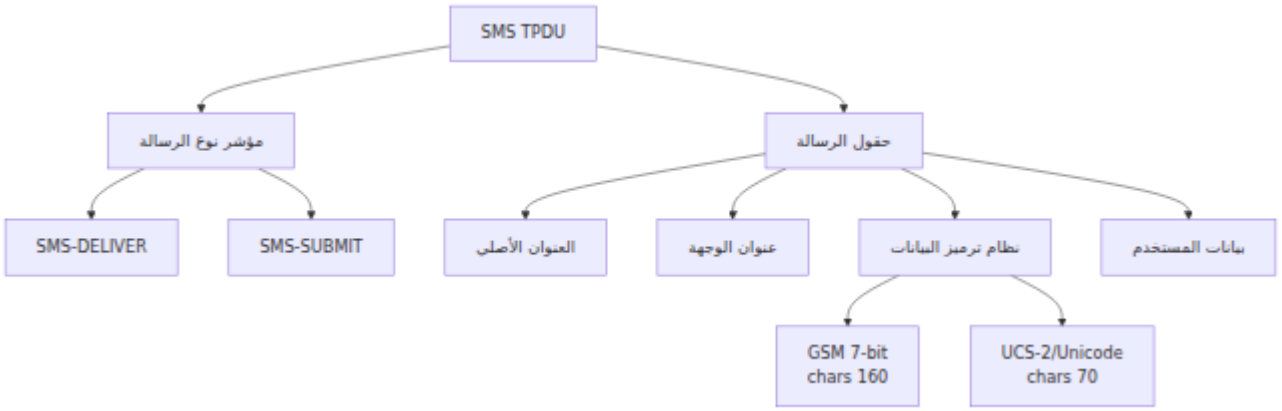
لماذا يعتبر منع الحلقات مهمًا

عندما يتلقى SMSc رسائل SMS منشأة من الهاتف المحمول (MO) من شبكة SS7، فإنه يقوم بإدراجها في قائمة الرسائل مع حقل `source_smsc` الذي يحدد أصلها (على سبيل المثال، `"SS7_MO_15551234567"`). بدون منع الحلقات، يمكن أن تكون هذه الرسائل:

1. تم إدراجها في قائمة الانتظار مع `source_smsc` يحتوي على "SS7" → تم استلامها من شبكة SS7
2. تم استردادها من قائمة الانتظار → تمت معالجتها للتسليم
3. تم إرسالها مرة أخرى إلى شبكة SS7 → مما يخلق حلقة

كيف يعمل

بحظر SMSc تقوم منطق منع الحلقات بتمييز بين الوجهات **على الشبكة** و**خارج الشبكة**. يقوم فقط عندما لا يكون المشترك الوجهة **على الشبكة**. هذا يمنع حلقات توجيه SS7 الرسائل من على سبيل) الشرعي بين المشتركين على نفس الشبكة MO→MT العبور. لا يزال يسمح بتسليم المثال، MSC-to-MSC عبر IP-SM-GW).



كشف الشبكة

يتحقق النظام مما إذا كان المشترك الوجهة على الشبكة قبل تطبيق SS7، عندما تنشأ رسالة من منع الحلقات:

1. **ذاكرة التخزين المؤقت VLR** (الأسرع): تحقق مما إذا كان لدى المشترك عنوان IMSI صالح في متتبع المشتركين مخزن مع VLR
2. **واجهة برمجة التطبيقات HSS** (السلطوية): إذا تم تكوين `hlr_api_base_url` MSISDN. إذا بواسطة HSS وفشلت الذاكرة المؤقتة، استعلام واجهة برمجة التطبيقات HSS، فهو على الشبكة. كان المشترك موجودًا في

إذا لم تجد أي من الفحوصات المشترك، تعتبر الوجهة **خارج الشبكة** ويتم حظر الرسالة.

التنفيذ

عند معالجة الرسائل من قائمة الانتظار، يتحقق SMSc من حقل `source_smsc`:

- **إذا كان `source_smsc` يحتوي على "SS7" والوجهة خارج الشبكة**:
  - يتم تخطي الرسالة
  - تمت إضافة حدث: `"منع الحلقة"` مع وصف يشير إلى أن الوجهة خارج الشبكة

    - يتم وضع علامة على الرسالة على أنها فشلت عبر طلب PUT
    - تم تسجيلها بمستوى تحذير

- **إذا كان `source_smsc` يحتوي على "SS7" والوجهة على الشبكة**:
    - تتم معالجة الرسالة بشكل عادي. هذه تدفق MO→MT شرعي.
    - تستمر عملية التسليم عبر واجهة برمجة التطبيقات HSS / مسار ذاكرة التخزين المؤقت VLR

- **إذا لم يحتوي `source_smsc` على "SS7"**:
    - تتم معالجة الرسالة بشكل عادي (على سبيل المثال، الرسائل من واجهة برمجة التطبيقات على الويب، SMPP، أو مصادر غير SS7 أخرى)

قيم SMSC المصدر

يمكن أن تحتوي الرسائل على قيم `source_smsc` متنوعة:

المصدر	قيمة المثال	الإجراء
شبكة SS7 (MO-FSM)، وجهة خارج الشبكة	`"SS7_MO_15551234567"`	**تم تخطيها** - منع الحلقة
شبكة SS7 (MO-FSM)، وجهة على الشبكة	`"SS7_MO_15551234567"`	**مسموح** - تسليم MO→MT على الشبكة
واجهة برمجة التطبيقات الخارجية/SMPP	`"ipsmgw"` أو `"api_gateway"`	معالجة بشكل عادي
SMSc أخرى	`"smsc-node-01"`	معالجة بشكل عادي

تتبع الأحداث

عندما يتم تخطي رسالة بسبب منع الحلقات، يتم تسجيل حدث:

```
{
  "message_id": 12345,
  "name": "منع الحلقة",
  "description": "تم تخطي الرسالة - source_smsc
'SS7_MO_15551234567' يحتوي على 'SS7' مما يمنع
والوجهة خارج الشبكة،
حلقة الرسالة"
}
```

هذا الحدث مرئي في:

- **واجهة الويب**: صفحة أحداث SS7 (`/events`)
- **قاعدة البيانات**: جدول `events` عبر واجهة برمجة التطبيقات
- **السجلات**: إدخالات سجل بمستوى تحذير

التكوين

يتم **تمكين منع الحلقات دائمًا** ولا يمكن تعطيله. هذه ميزة أمان حيوية لمنع انقطاع الشبكة وإما ذاكرة `smsc_mode_enabled: true` بسبب حلقات الرسائل. يتطلب تجاوز الشبكة كلاً من التخزين المؤقت VLR أو `hlr_api_base_url` ليتم تكوينها.

سيناريوهات المثال

السيناريو 1: MO-SMS من SS7 (محظور) إلى مشترك خارج الشبكة

```
1. الهاتف المحمول → MSC/VLR → IP-SM-GW (عبر MO-ForwardSM)
2. يتلقى IP-SM-GW MO-FSM، يتم إدراجه في قائمة الانتظار: source_smsc
= "SS7_MO_9991900"
3. يقوم Auto-flush باسترداد الرسالة من قائمة الانتظار
4. خارج → HSS أو VLR الوجهة غير موجودة في ذاكرة التخزين المؤقت
الشبكة
5. يكتشف IP-SM-GW "SS7" في source_smsc + تخطي → وجهة خارج الشبكة
6. تم تسجيل الحدث: "منع الحلقة"
7. تم إرسال MT-ForwardSM أو SRI-for-SM لم تم (منع الحلقة)
```

السيناريو 2: MO-SMS من SS7 (مسموح) إلى مشترك على الشبكة

```
1. الهاتف المحمول → MSC/VLR → IP-SM-GW (عبر MO-ForwardSM)
2. يتلقى IP-SM-GW MO-FSM، يتم إدراجه في قائمة الانتظار: source_smsc
= "SS7_MO_9991900"
3. يقوم Auto-flush باسترداد الرسالة من قائمة الانتظار
4. تم العثور على الوجهة في واجهة برمجة التطبيقات HSS → مشترك على
الشبكة
5. يسمح IP-SM-GW بالتسليم على الرغم من الأصل SS7
6. تعيد HSS IMSI الحقيقي + VLR → يتم إرسال MT-ForwardSM مباشرة إلى
VLR
```

SMSc تتبع المشتركين في

يتضمن SMSc **متتبع المشتركين** GenServer الذي يحافظ على حالة الوقت الحقيقي للمشتركين بناءً على رسائل alertServiceCenter ومحاولات تسليم الرسائل.

الغرض

يوفر المتتبع:

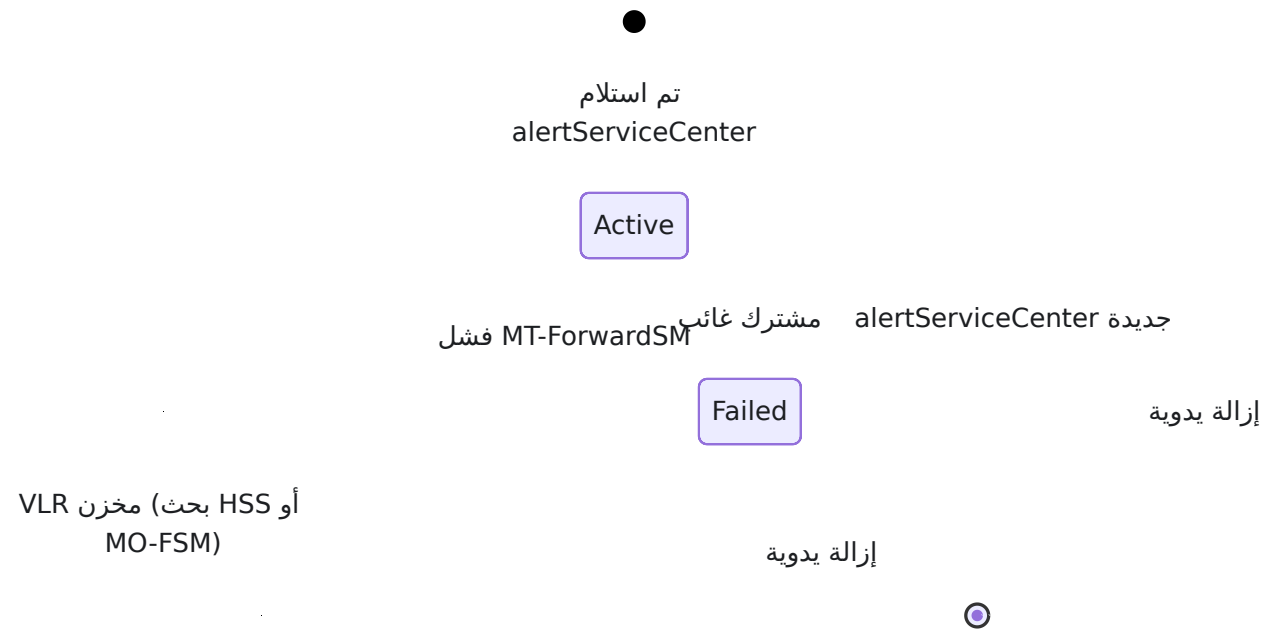
- **مراقبة القابلية للوصول**: أي المشتركين قابلون للوصول حاليًا
- **تتبع HLR**: أي HLR أرسل alertServiceCenter لكل مشترك
- **عدادات الرسائل**: عدد الرسائل المرسلة/المستلمة لكل مشترك
- **تتبع الفشل**: وضع علامة على المشتركين على أنهم فشلوا عندما تفشل محاولات التسليم
- **رؤية واجهة الويب**: لوحة معلومات في الوقت الحقيقي تعرض جميع المشتركين المتعقبين

المعلومات المتعقبة

لكل مشترك، يخزن المتتبع:

الحقل	الوصف	المثال
`msisdn`	رقم هاتف المشترك (المفتاح)	"15551234567"
`imsi`	IMSI الخاص بالمشترك (الحقيقي إذا كان متاحًا من HSS)	"001010123456789"
`hlr_gt`	HLR GT الذي أرسل alertServiceCenter	"15551111111"
`messages_sent`	عدد رسائل MT-FSM المرسلة	5
`messages_received`	عدد رسائل MO-FSM المستلمة	2
`status`	`:failed` أو `:active`	`:active`
`updated_at`	الطابع الزمني لآخر تحديث	1730246400
`vlr_address`	عنوان VLR GT المخزن	"14155550100"
`vlr_cached_at`	الطابع الزمني عند آخر تخزين لـ VLR	1730246400

انتقالات الحالة



السلوك

عند استلام alertServiceCenter:

- إنشاء أو تحديث إدخال المشترك
- تعيين `status = :active`
- تسجيل HLR GT
- حل IMSI الحقيقي من واجهة برمجة التطبيقات HSS (إذا تم تكوينها)، وإلا استخدام IMSI الاصطناعي
- الحفاظ على ذاكرة التخزين المؤقت VLR الحالية وعدادات الرسائل

عند عودة واجهة برمجة التطبيقات HSS بمشترك مع VLR:

- تخزين عنوان VLR وIMSI الحقيقي في المتتبع
- تستخدم التسليمات اللاحقة VLR إعادة استعلام HSS المخزن دون

عند استلام MO-ForwardSM:

- تخزين GT المتصل SCCP كعنوان VLR للمشترك (GT المتصل في MO-ForwardSM هو VLR الذي نقل الرسالة)
- إنشاء إدخال متتبع إذا لم يكن موجودًا

عند فشل تسليم MT-ForwardSM:

- مسح عنوان VLR المخزن في HSS حتى يتم إعادة استعلام واجهة برمجة التطبيقات محاولة التسليم التالية

عند نجاح SRI-for-SM:

- زيادة عداد `messages_sent`
- تحديث الطابع الزمني `updated_at`

عند فشل SRI-for-SM:

- تعيين `status = :failed`
- الاحتفاظ به في المتتبع للمراقبة

عند إزالة المشترك:

- الحذف من جدول ETS
- لم يعد يظهر في واجهة الويب

SMSc واجهة الويب - صفحة مشتركي

التحديث التلقائي: كل 2 ثانية **المسار**: `/smsc_subscribers`

ملاحظة: هذه الصفحة متاحة فقط عند التشغيل في وضع SMSc. بعد تمكين تكوين SMSc في `config/config.exs`، يجب عليك إعادة تشغيل التطبيق لجعل المسار متاحًا.

توفر صفحة **مشتركي SMSc** مراقبة في الوقت الحقيقي لجميع المشتركين المتعقبين:

الميزات

1. **جدول المشتركين**
    - MSISDN، IMSI، HLR GT، VLR GT (عنوان VLR المخزن)
    - عدادات الرسائل المرسلة/المستلمة
    - شارة الحالة (نشط/فشل) مع ترميز الألوان
    - الطابع الزمني لآخر تحديث والمدة
    - زر الإزالة للمشتركين الفرديين

2. **إحصائيات ملخصة**

   - إجمالي المشتركين المتعقبين
   - عدد المشتركين النشطين
   - عدد المشتركين الفاشلين
   - عدد HLRs الفريدة

3. **الإجراءات**

   - مسح الكل: إزالة جميع المشتركين المتعقبين
   - إزالة: إزالة مشترك فردي

عرض المثال

```
┌──────────────────────────────────────────────────────────────────────────────┐
│ المتعقبون                                  الإجمالي: 3 SMSc مشتركو          │
├──────────────┬──────────────────┬──────────────┬──────────────┬──────────────┤
│ MSISDN       │ IMSI             │ HLR GT       │ VLR GT       │ Msg          │
│              │                  │              │              │ S/           │
├──────────────┼──────────────────┼──────────────┼──────────────┼──────────────┤
│ 15551234567  │ 001010123456789  │ 15551111111  │ 14155550100  │ 5/2          │
│ 15559876543  │ 001010987654321  │ 15551111111  │ -            │ 0/0          │
│ 15551112222  │ 001010111222233  │ 15552222222  │ 14155550200  │ 3/1          │
└──────────────┴──────────────────┴──────────────┴──────────────┴──────────────┘

الملخص: الإجمالي: 3 | النشط: 2 | الفاشل: 1 | HLRs 2 :الفريدة
```

وظائف واجهة برمجة التطبيقات

يعرض المتتبع هذه الوظائف للوصول البرمجي:

```
# يتم استدعاؤه عند استلام alertServiceCenter
SMSc.SubscriberTracker.alert_received(msisdn, imsi, hlr_gt)

# زيادة عدادات الرسائل
SMSc.SubscriberTracker.message_sent(msisdn)
SMSc.SubscriberTracker.message_received(msisdn)

# وضع علامة على الفشل (فشل SRI-for-SM)
SMSc.SubscriberTracker.mark_failed(msisdn)

# الإزالة من التتبع
SMSc.SubscriberTracker.remove_subscriber(msisdn)

# وظائف الاستعلام
SMSc.SubscriberTracker.get_active_subscribers()
SMSc.SubscriberTracker.get_subscriber(msisdn)
SMSc.SubscriberTracker.count_subscribers()
SMSc.SubscriberTracker.clear_all()
```

التكامل

يتم دمج المتتبع تلقائيًا مع:

- **معالج alertServiceCenter**: يستدعي `alert_received/3` عند تحديث الموقع بنجاح
- **معالج SRI-for-SM**: يزيد `messages_sent` عند التوجيه الناجح
- **معالج المشترك الغائب**: يستدعي `mark_failed/1` عندما يكون المشترك غائبًا
- **أخطاء المشترك غير المعروف**: يستدعي `mark_failed/1` عندما تفشل SRI-for-SM

تخزين عنوان VLR

يقوم متتبع المشتركين بتخزين HSS، مع تكوين واجهة برمجة التطبيقات IP-SM-GW عند العمل كـ عناوين VLR لتجنب استعلام HSS في كل محاولة تسليم MT.

كيف يعمل

يتم ملء ذاكرة التخزين المؤقت VLR من مصدرين:

1. **بحث واجهة برمجة التطبيقات HSS**: عندما يستعلم خط أنابيب التسليم عن واجهة برمجة التطبيقات HSS وعنوان IMSI خادم، يتم تخزين VLR ويحصل على مشترك مع VLR في متتبع المشتركين.
2. **MO-ForwardSM**: عندما تصل MO-ForwardSM من مشترك، يتم تخزين GT للمشترك VLR كعنوان (الذي نقل الرسالة VLR/MSC وهو) SCCP المتصل.

في محاولات التسليم MT اللاحقة، يتم التحقق من الذاكرة المؤقتة أولاً. إذا كان هناك إدخال صالح (TTL موجودين، ضمن VLRو IMSI كلا من)، يتم تخطي واجهة برمجة التطبيقات HSS تمامًا.

إبطال الذاكرة المؤقتة

يتم مسح ذاكرة التخزين المؤقت VLR عندما:

- **يفشل تسليم MT-ForwardSM** (انتهاء المهلة، خطأ SCCP، رفض VLR): قد تكون المخزنة قديمة VLR
- **تنتهي صلاحية TTL**: يمكن تكوين TTL عبر `vlr_cache_ttl_seconds`. تعتبر الذاكرة المؤقتة الإدخالات التي تزيد عن TTL كإخفاقات في التخزين المؤقت.

الحالي لأن VLR يتم الحفاظ على إدخال alertServiceCenter. لا يتم مسح الذاكرة المؤقتة عند المشترك قد لا يزال قابلًا للوصول في نفس VLR.

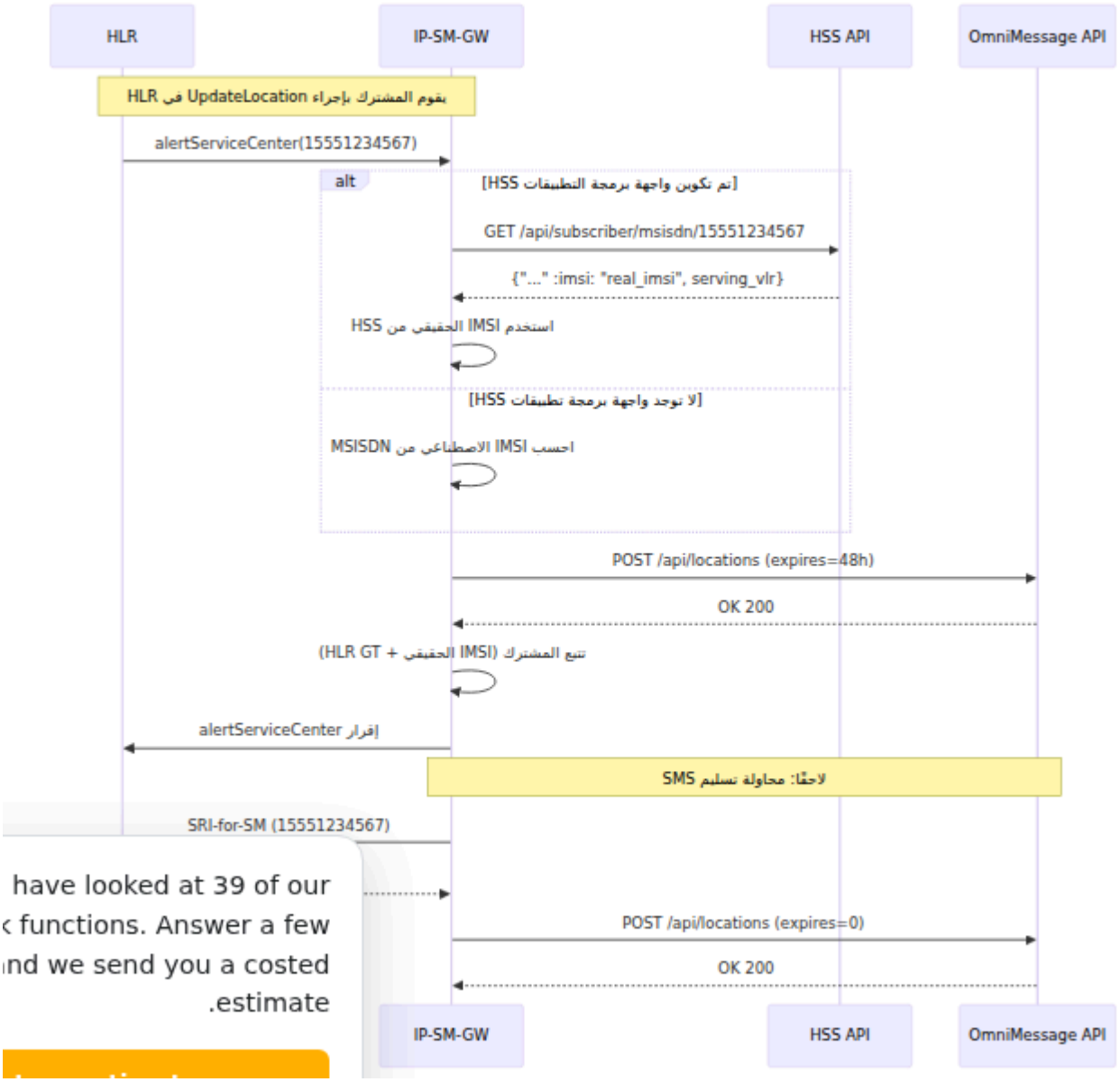
التكوين

```
config :omniss7,
  # نقطة نهاية واجهة برمجة التطبيقات HSS/HLR (مطلوب لتخزين VLR
والتسليم على الشبكة)
  hlr_api_base_url: "https://omnihss.example.com:8443",
  # مفتاح واجهة برمجة التطبيقات OmniHSS (مطلوب عند تعيين
hlr_api_base_url)
  hlr_api_key: "<مفتاح واجهة برمجة التطبيقات OmniHSS مع دور
hlr_interworking>",

  # TTL لذاكرة التخزين المؤقت VLR 1 = 3600 :الافتراضي) بالثواني
(ساعة
  vlr_cache_ttl_seconds: 3600
```

المعامل	النوع	مطلوب	الافتراضي	
`hlr_api_base_url`	سلسلة	لا	`nil`	U عنوان ة برمجة نطبيقات ن البحث كين على وتخزين مباشر لـ كون .MT دم جميع ت تدفق for-SM.
`hlr_api_key`	سلسلة	عند تعيين `hlr_api_base_url`	`nil`	ة برمجة نطبيقات O يرسله `x-` رأس لب `key` ة برمجة نطبيقات ناح الذي On يعينه `hlr_in` تم تعيين `hlr_ap` فقودًا أو كل طلب ة برمجة نطبيقات طأ يذكر `hlr_ap`
`vlr_cache_ttl_seconds`	عدد صحيح	لا	`3600`	ى للعمر ي لإدخال ر صالحًا. رة، تعيد

المعامل	النوع	مطلوب	الافتراضي	
				م التالية ة برمجة نطبيقات ن حركة ة برمجة نطبيقات ن الدقة محمولين

مخطط التسلسل



have looked at 39 of our
k functions. Answer a few
nd we send you a costed
.estimate

خدمة Auto-Flush لرسائل SMS

تقوم خدمة **Auto-Flush** بمعالجة رسائل SMS المعلقة تلقائيًا.

لإشارة مرجعية على معلمات التكوين، راجع تكوين Auto-Flush في مرجع التكوين.

التكوين

```
config :omniss7,
  auto_flush_enabled: true,           # تمكين/تعطيل التفريغ التلقائي
  auto_flush_interval: 10_000,        # فترة الاستعلام بالمللي ثانية
  auto_flush_dest_smsc: nil,          # تصفية: nil = الكل
  auto_flush_tps: 10                  # الحد الأقصى للمعاملات في الثانية
```

كيف يعمل

1. **الاستعلام**: كل `auto_flush_interval` مللي ثانية، استعلام واجهة برمجة التطبيقات عن الرسائل المعلقة
2. **التصفية**: تصفية ختيارية بواسطة `auto_flush_dest_smsc`
3. **تحديد المعدل**: معالجة ما يصل إلى `auto_flush_tps` رسائل لكل دورة
4. **التسليم**: لكل رسالة:
   - إرسال **SRI-for-SM** (إرسال معلومات التوجيه للرسالة القصيرة) إلى HLR للحصول على معلومات التوجيه
     - تعيد HLR IMSI اصطناعي محسوب من MSISDN
     - تعيد HLR عنوان SMSC GT حيث يجب إرسال MT-ForwardSM
     - للحصول على الوثائق راجع تفاصيل SRI-for-SM في دليل HLR الكاملة
   - عند النجاح، إرسال **MT-forwardSM** إلى MSC/VLR
   - تحديث حالة الرسالة عبر واجهة برمجة التطبيقات (تم التسليم/فشل)
   - إضافة تتبع الحدث عبر واجهة برمجة التطبيقات

 للحصول على شرح كامل حول كيفية عمل SRI-for-SM، بما **الغوص التقني العميق**: وتوليد IMSI، وتكوين عنوان مركز الخدمة GT، إلى IMSI في ذلك تحويل MSISDN الاصطناعي مع الحفاظ على الخصوصية، راجع قسم SRI-for-SM في دليل تكوين HLR.

مقاييس SMSc

تقوم OmniSS7 بعرض مقاييس Prometheus. تتعقب المقاييس المشتركة نشاط MAP SMS. لا المقاييس ذات الصلة بعمليات SMS هي: توجد عدادات طوابير/تسليم محددة لـ SMSc.

المقياس	النوع	التسميات	الصلة بـ SI
`map_requests_total`	عداد	`operation`	M يعد عمليات مل العمليات ذات الصلة بـ SMS `sri_for_sr` `forward_sr` `MTforward_` و `prn`.
`map_request_duration_milliseconds`	هيستوجرام	`operation`	دورات طلب/ استجابة MAP في ذلك SRI-SM و MT-ForwardSM

تتكون المجموعة الكاملة من المقاييس المعلنة (SMS ليس كلها محددة بـ) من: `map_requests_total`، `cap_requests_total`، `ussd_requests_total`، `ussd_active_sessions`، `map_pending_requests`، `omniss7_license_status`، و `map_request_duration_milliseconds`.

استعلامات مثال:

```
# معدل عمليات MT-ForwardSM خلال آخر 5 دقائق
rate(map_requests_total{operation="MTforward_sm"}[5m])

# معدل عمليات SRI-for-SM خلال آخر 5 دقائق
rate(map_requests_total{operation="sri_for_sm"}[5m])

# متوسط مدة MT-ForwardSM (آخر 5 دقائق)
rate(map_request_duration_milliseconds_sum{operation="MTforward_sm"}
[5m]) /
rate(map_request_duration_milliseconds_count{operation="MTforward_sm"
[5m])
```

استكشاف الأخطاء وإصلاحها في SMSc

المشكلة: عدم تسليم الرسائل

التحققات:

1. تحقق من تمكين التفريغ التلقائي
2. تحقق من اتصال قاعدة البيانات
3. راقب السجلات بحثًا عن الأخطاء
4. تحقق من أن اتصال M3UA نشط
5. تحقق من حدود TPS

المشكلة: عمق الطابور مرتفع

الأسباب المحتملة:

- حد TPS منخفض جدًا
- مشاكل في مهلة HLR
- مشاكل في الاتصال بالشبكة
- أرقام وجهة غير صالحة

الحلول:

- زيادة `auto_flush_tps`
- تحقق من توفر HLR
- مراجعة سجلات الرسائل الفاشلة

واجهة برمجة تطبيقات MT-forwardSM

إرسال SMS عبر API

نقطة النهاية API: `POST /api/MT-forwardSM`

جميع الحقول الأربعة مطلوبة. أسماء الحقول هي snake_case (لاحظ أن `destination_service_centre` تستخدم التهجئة البريطانية). `smsPDU` هو TPDU SMS مشفر بالهيكس.

الطلب:

```
{
  "imsi": "234509876543210",
  "destination_service_centre": "447999555111",
  "originating_service_center": "447999123456",
  "smsPDU":
"040B917477218345F60000157030185714 0C0BD4F29C0E9281C4E1F11A"
}
```

الاستجابة:

عند النجاح (HTTP 200)، تحتوي الاستجابة على مفتاح `result` واحد يحمل نتيجة TCAP المنسقة من حوار MT-ForwardSM (لا يوجد `message_id`، والقيمة ليست السلسلة النصية الحرفية `"success"`):

```
{
  "result": "<formatted TCAP result>"
}
```

إذا انتهت مهلة حوار MAP، يد نقطة النهاية ت HTTP 504 مع `{"error": "timeout"}`.
الطلب غير الصحيح/المفقود الحقول يعيد HTTP 400 مع `{"error": "invalid request"}`.

الوثائق ذات الصلة

وثائق OmniSS7:

- ← العودة إلى الوثائق الرئيسية
- دليل تكوين HLR - إعداد وضع HLR والعمليات
  - **تفاصيل تقنية SRI-for-SM** - وثائق كاملة حول تحويل MSISDN إلى IMSI وتكوين مركز الخدمة
- واجهة الويب - دليل الميزات الشائعة، API، المراقبة
- دليل عميل MAP - عمليات MAP
- المرجع الفني - مواصفات البروتوكول

وثائق OmniMessage: لإعداد توجيه الرسائل، إدارة الطوابير، تتبع التسليم، تحديد المعدلات، والتحليلات، يرجى الرجوع إلى **وثائق منتج OmniMessage**. يحتوي OmniMessage على جميع منطق توجيه الرسائل، خوارزميات إعادة المحاولة للطوابير، معالجة تقارير التسليم، ومحرك قواعد الأعمال.

OmniSS7 بواسطة خدمات شبكة Omnitouch

دليل تكوين M3UA و M2PA STP

← العودة إلى الوثائق الرئيسية

يوفر هذا الدليل تكوينًا مفصلًا لاستخدام OmniSS7 كنقطة **نقل الإشارات (STP)**.

جدول المحتويات

1. ما هي نقطة نقل الإشارات (STP)؟
2. أدوار شبكة STP
3. التواصل مع الشبكات TDM
4. تمكين وضع STP
5. تكوين الأقران
6. دعم بروتوكول M2PA
    - M3UA مقابل M2PA
    - تكوين أقران M2PA
    - تكوين SLTM
    - إدارة M2PA عبر واجهة الويب
    - مقاييس M2PA
7. توجيه رمز النقطة
8. توجيه العنوان العالمي
9. معالجة عبور SCCP
    - التوجيه على GTT بعد SSN
    - فرض عداد القفزات
    - إرجاع الرسالة عند الفشل
10. حالة الوجهة (إدارة مسار الإشارات)
11. تصفية البوابة (جدار حماية SCCP)
12. ميزات إدارة المسار
    - تعطيل المسارات
    - DROP Routes - منع حلقات التوجيه

13. التوجيه المتقدم
14. اختبار التكوين
15. المقاييس والمراقبة
16. مراقبة أقران M3UA
    - معالجة سياق توجيه M3UA
    - تمييز متعدد الأقران بنفس IP

ما هي نقطة نقل الإشارات (STP)؟

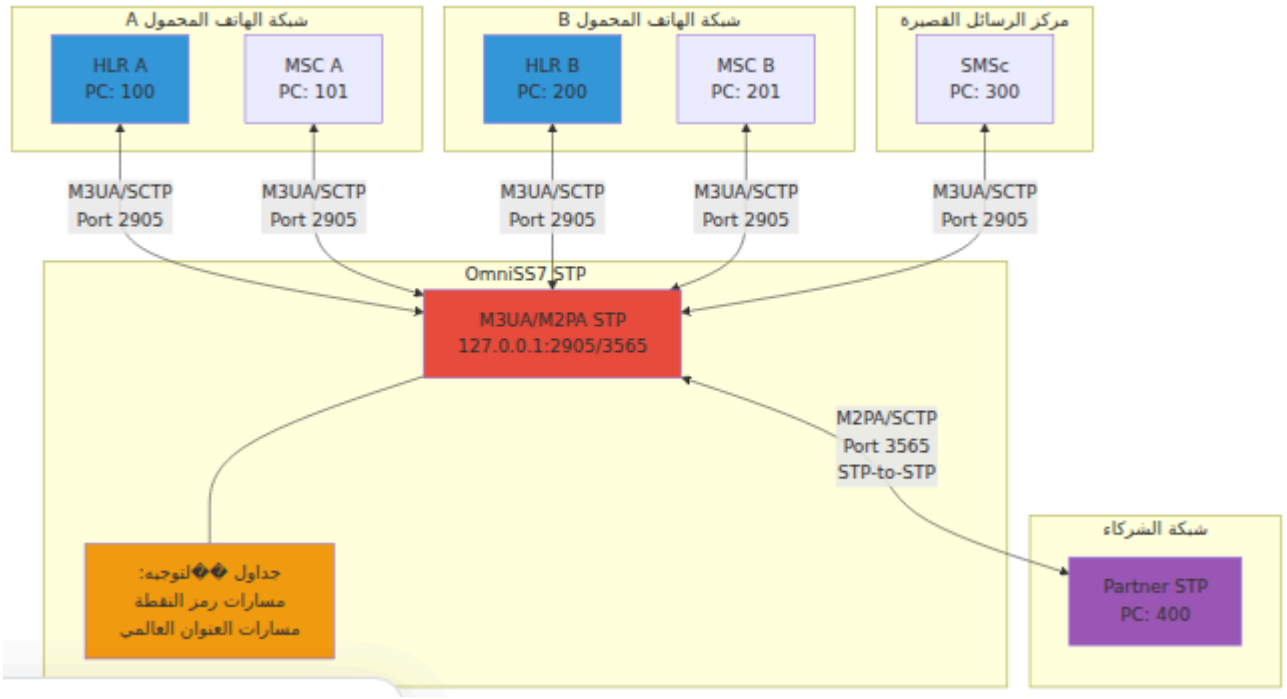
وشبكات الإشارات SS7 هي عنصر شبكة حيوي في شبكات **نقل الإشارات** نقطة **(STP)** تقوم بتوجيه رسائل الإشارات بين عقد الشبكة IP المعتمدة على.

وظائف STP

- **توجيه الرسائل**: توجيه حركة مرور إشارات SS7 بناءً على رمز النقطة الوجهة (PC) أو العنوان العالمي (GT)
- **ترجمة البروتوكولات**: جسر الشبكات التقليدية SS7 مع الشبكات المعتمدة على IP M3UA/SCTP

- **توزيع الحمل**: توزيع الحركة عبر وجهات متعددة باستخدام التوجيه القائم على الأولوية
- **بوابة الشبكة**: ربط شبكات الإشارات المختلفة ومقدمي الخدمة
- **إخفاء الطوبولوجيا**: يمكنه إعادة كتابة العناوين لإخفاء الطوبولوجيا الداخلية للشبكة

مخطط شبكة STP



أدوار شبكة STP موضحة

ASP (عملية خادم التطبيق)

- **الدور**: عميل يتصل بـ SGP/STP عن بُعد
- **الاتجاه**: اتصال خارجي
- **حالة الاستخدام**: تتصل شبكة الشريك STP الخاصة بك بـ STP

SGP (عملية بوابة الإشارات)

- **الدور**: خادم يقبل الاتصالات من ASPs
- **الاتجاه**: ات��ال داخلي
- **حالة الاستخدام**: تتصل الشبكات الشريكة بـ STP الخاص بك

AS (خادم التطبيق)

- **التعريف**: تجميع منطقي لواحد أو أكثر من ASPs
- **الغرض**: يوفر التكرار ومشاركة الحمل
- **حالة الاستخدام**: عدة ASPs تخدم نفس الوجهة

التواصل مع الشبكات TDM عبر بوابات الإشارات

تستخدم SIGTRAN (M3UA/M2PA عبر SCTP). هي منصة إشارات معتمدة على IP OmniSS7 القديمة، تحتاج إلى **بوابة TDM المعتمدة على SS7** لتبادل الإشارات مع **شبكات الإشارات (SGW)** لجسر العالمين.

ما هي بوابة الإشارات؟

بوابة الإشارات (SGW) هي عنصر شبكة يقوم بتحويل إشارات SS7 بين:

- **جانب TDM**: SS7 التقليدي باستخدام MTP1/MTP2/MTP3 عبر روابط E1/T1
- **جانب IP**: SIGTRAN باستخدام M3UA أو M2PA عبر SCTP/IP

تعمل SGW كمترجم بروتوكولات. إنها تسمح للتطبيقات المعتمدة على IP مثل OmniSS7 بالتواصل مع المعدات القديمة المعتمدة على TDM (المفاتيح، STPs، HLRs) عم التي لا ت إشارات IP.

بنية TDM إلى IP



مقارنة مكدس البروتوكولات

الطبقة	TDM SS7	بوابة الإشارات	IP (SIGTRAN)
جزء المستخدم	SCCP/TCAP/MAP	← شفاف →	SCCP/TCAP/MAP
الشبكة	MTP3	تحويل	M3UA/M2PA
الرابط	MTP2	إنهاء	SCTP
الفيزيائي	MTP1 (E1/T1)	إنهاء	IP/Ethernet

تقوم SGW بإنهاء MTP1/MTP2 على جانب TDM وتقدم بيانات مستخدم MTP3 عبر M3UA أو M2PA على جانب IP. تمر الطبقات العليا (SCCP، TCAP، MAP، CAP) عبرها بشكل شفاف.

توصيل OmniSS7 ببوابة الإشارات

يتصل OmniSS7 ببوابة الإشارات كـ **ASP (عملية خادم التطبيق)**، بينما تعمل SGW كـ **SGP (عملية بوابة الإشارات)**.

مثال على التكوين

```
config :omniss7,
  # الاتصال ببوابة الإشارات كـ ASP
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :sgw_connection,
    # نقطة النهاية المحلية (OmniSS7)
    local_ip: {10, 0, 0, 1},
    local_port: 0,                    # منفذ محلي ديناميكي
    # نقطة النهاية لبوابة الإشارات
    remote_ip: {10, 0, 0, 100},       # عنوان IP لـ SGW
    remote_port: 2905,                # منفذ M3UA لـ SGW
    routing_context: 1                # كما تم تعيينه بواسطة SGW
  },

  # أو تكوين كقرين لتوجيه STP
  peers: [
    %{
      peer_id: 1,
      name: "TDM_Gateway",
      role: :client,                  # يبدأ OmniSS7 الاتصال
      local_ip: {10, 0, 0, 1},
      local_port: 0,
      remote_ip: {10, 0, 0, 100},     # عنوان IP لـ SGW
      remote_port: 2905,
      routing_context: 1,
      point_code: 100,                # نطاق رمز النقطة خلف SGW
      network_indicator: :international
    }
  ],

  # توجيه رموز النقطة لشبكة TDM عبر البوابة
  m3ua_routes: [
    %{
      dest_pc: 100,                   # رمز نقطة MSC TDM
      peer_id: 1,                     # التوجيه عبر قرين SGW
      priority: 1,
      network_indicator: :international
    },
    %{
      dest_pc: 200,                   # رمز نقطة HLR TDM
      peer_id: 1,
      priority: 1,
```

```
      network_indicator: :international
    }
  ]
```

اعتبارات تكوين SGW

عند تكوين بوابة الإشارات الخاصة بك للعمل مع OmniSS7:

المعلمة	الوصف	القيمة النموذجية
عنوان IP البعيد	عنوان IP الخاص بـ OmniSS7	عنوان IP الخاص بالخادم الخاص بك
المنفذ البعيد	منفذ SCTP الخاص بـ OmniSS7	2905
سياق التوجيه	معرف AS في SGW	معين بواسطة مسؤول SGW
رموز النقطة	رموز النقطة لشبكة TDM القابلة للوصول عبر SGW	محددة حسب الشبكة
وضع الحركة	وضع معالجة حركة ASP	تجاوز أو مشاركة الحمل

سيناريوهات النشر

السيناريو 1: وصول تطبيق IP إلى شبكة TDM

يرسل OmniSS7 استفسارات MAP (SRI-SM، PRN) إلى HLRs المعتمدة على TDM عبر SGW:

```
OmniSS7 (ASP) → SGW (SGP) → شبكة TDM → HLR
     M3UA          MTP2          MTP3
```

السيناريو 2: OmniSS7 كـ STP بين IP و TDM

يقوم OmniSS7 بتوجيه الحركة بين عناصر الشبكة المعتمدة على IP وشبكات TDM:

```
IP SMSC → OmniSS7 STP → SGW → TDM HLR
                ↓
          IP المعتمد على HLR
```

السيناريو 3: SGW مزدوج للتكرار

الاتصال ببوابتين للإشارات لضمان توافر عالي:

```
peers: [
  %{
    peer_id: 1,
    name: "SGW_Primary",
    role: :client,
    remote_ip: {10, 0, 0, 100},
    remote_port: 2905,
    point_code: 100,
    # ... تكوين آخر
  },
  %{
    peer_id: 2,
    name: "SGW_Backup",
    role: :client,
    remote_ip: {10, 0, 0, 101},
    remote_port: 2905,
    point_code: 100,
    # ... تكوين آخر
  }
],

m3ua_routes: [
  # SGW_Primary المسار الرئيسي عبر
  %{dest_pc: 100, peer_id: 1, priority: 1, network_indicator:
:international},
  # SGW_Backup مسار النسخ الاحتياطي عبر
  %{dest_pc: 100, peer_id: 2, priority: 2, network_indicator:
:international}
]
```

تمكين وضع M3UA STP

يجب عليك تقديم تكوين STP، في أوضاع مختلفة. لاستخدامه كنقطة OmniSS7 يمكن أن يعمل توجيه STP تحت مفتاح تطبيق `:omniss7`.

أين توجد التكوينات

الغرض	الملف
واجهة الويب، تكوين افتراضي، يتم تحميله دائمًا (على المنفذ 8445، تسجيل API مستمع)	`config/config.exs`
كامل وجاهز للتعديل STP مثال (`sctp_handler`، مسارات الأقران، PC، مسارات GT)	`config/stp_example_runtime.exs`
تجاوزات وقت التشغيل لكل نشر. في هذا المستودع، هو نموذج بسيط؛ تقوم خوادم الإنتاج بنشر الخاصة بها	`config/runtime.exs`

> **ملاحظة**: كانت الإصدارات السابقة تحتوي على ثلاث كتل وضع معطلة داخل `config/runtime.exs`. أصبح مثال STP الآن في ملفه الخاص، `config/stp_example_runtime.exs`. لم يعد هذا التخطيط موجودً��.

التبديل إلى وضع STP

لإعداد STP، انسخ تكوين المثال إلى التكوين الذي يقوم نشرك بتحميله في وقت التشغيل:

1. **افتح** `config/stp_example_runtime.exs` واستعرض مثال STP (معالج SCTP، مسارات GT، PC، الأقران، مسارات).
2. **انسخ** كتلة `config :omniss7, ...` إلى `config/runtime.exs` الخاص بنشرك (أو دمج المفاتيح ذات الصلة).
3. **خصص** عناوين IP، المنافذ، رموز النقطة، الأقران، والمسارات لشبكتك.
4. **أعد تشغيل** التطبيق: `iex -S mix`

تكوين وضع STP

يقوم مثال STP في `config/stp_example_runtime.exs` بتكوين مستمع STP داخلي (`sctp_handler`)، وتمكين توجيه GT، وتعريف الأقران بالإضافة إلى مسارات GT و PC:

```
config :omniss7,
  # مستمع SCTP (خادم STP داخلي) - انظر وضع STP المستقل أدناه
  sctp_handler: %{
    enabled: true,
    local_ip: {127, 0, 0, 1},
    local_port: 2905,
    point_code: 101            # رمز النقطة الخاص بـ STP
  },

  # تمكين توجيه العنوان العالمي (بالإضافة إلى توجيه PC)
  enable_gt_routing: true,

  # أقران M3UA، مسارات PC، ومسارات GT تتبع
  m3ua_peers: [ %{peer_id: 1, name: "STP_West", role: :client,
...} ],
  m3ua_routes: [ %{dest_pc: 100, peer_id: 1, priority: 1,
network_indicator: :international} ],
  m3ua_gt_routes: [ %{gt_prefix: "44", peer_id: 2, priority: 1,
description: "أرقام المملكة المتحدة"} ]
```

توجيه رمز النقطة، و توجيه العنوان العالمي أدناه لمجموعة المعلمات M3UA، انظر تكوين أقران الكاملة.

معلمات التكوين لتخصيص

للحصول على مرجع كامل لجميع معلمات التكوين، انظر مرجع التكوين.

هذه هي المفاتيح الرئيسية لـ STP تحت `config :omniss7`:

المعلمة	النوع	الافتراضي	الوصف	المثال
`sctp_handler`	خريطة	*اختياري*	تكوين مستمع STP داخلي (انظر وضع STP المستقل)	`%{enabled: true, local_port: 2905, point_code: 101}`
`enable_gt_routing`	بولياني	`false`	تمكين توجيه العنوان العالمي (بالإضافة إلى توجيه PC)	`true`
`m3ua_peers`	قائمة	`[]`	تعريفات أقران M3UA / M2PA (انظر تكوين أقران M3UA)	لا شيء
`m3ua_routes`	قائمة	`[]`	مسارات رمز النقطة (انظر توجيه رمز النقطة)	لا شيء
`m3ua_gt_routes`	قائمة	`[]`	مسارات العنوان العالمي (انظر توجيه العنوان العالمي)	لا شيء

> **نصيحة:** استخدم تعدد الموائل SCTP من خلال توفير قائمة من عناوين IP لـ `local_ip` و/أو `remote_ip` لتمكين الفشل التلقائي. انظر دليل تعدد موائل SCTP.

ماذا يحدث عند تمكين وضع STP

بمجرد تكوين الأقران والمسارات، تظهر واجهة الويب:

- **SS7 أحداث** - تسجيل الأحداث
- **M3UA** - حالة الاتصال
- **التوجيه** *STP* - إدارة جدول المسار ← محدد لـ
- **اختبار التوجيه** *STP* - اختبار المسار ← محدد لـ
- **الموارد** - مراقبة النظام
- **التكوين** - عارض التكوين

ملاحظات مهمة

- يجب السماح لبروتوكول SCTP (بروتوكول IP 132) عبر جدران الحماية
- المنفذ الافتراضي لـ M3UA هو 2905 (معيار الصناعة)
- تأكد من توفر موارد النظام الكافية لمعالجة حركة المرور التوجيهية
- **استمرارية التوجيه**: جميع المسارات المكونة عبر واجهة الويب أو API مخزنة في **قاعدة بيانات Mnesia** و **تستمر بعد إعادة التشغيل**
- **دمج التكوين**: يتم تحميل المسارات من `runtime.exs` عند بدء التشغيل ودمجها مع مسارات Mnesia
- بعد تغيير التكوين، يجب عليك إعادة تشغيل التطبيق لتفعيل التغييرات
- **واجهة الويب**: انظر دليل واجهة الويب لإدارة المسارات عبر واجهة الويب
- **الوصول إلى API**: انظر دليل API لوثائق REST API والوصول إلى واجهة Swagger

---

وضع STP المستقل

مستقل يستمع للاتصالات الواردة. هذه هي الوضع **M3UA STP** يمكن تشغيل **خادم** المستخدم في المثال في `config/stp_example_runtime.exs`.

تمكين STP المستقل

في تكوين وقت التشغيل الخاص بك، أضف هذا التكوين تحت `config :omniss7`:

```
config :omniss7,
  sctp_handler: %{
    enabled: true,
    local_ip: {127, 0, 0, 1},    # عنوان IP للاستماع عليه
    local_port: 2905,             # المنفذ للاستماع عليه
    point_code: 100               # رمز النقطة الخاص بـ STP
  }
```

معلمات تكوين STP

المعلمة	النوع	الافتراضي	الوصف	المثال
`enabled`	بولياني	`false`	تمكين خادم STP المستقل	`true`
`local_ip`	مجموعة	`{127, 0, 0, 1}`	عنوان IP للاستماع للاتصالات	`{0, 0, 0, 0}`
`local_port`	عدد صحيح	`2905`	المنفذ للاستماع عليه	`2905`
`point_code`	عدد صحيح	*مطلوب*	رمز النقطة الخاص بـ STP	`100`

متى تستخدم STP المستقل

- MAP دون وظيفة عميل M3UA **توجيه نقي**: عندما تحتاج فقط إلى توجيه
- **STP مركزي**: لإنشاء جهاز توجيه إشارات مركزي لعناصر الشبكة المتعددة
- مركزي STP من خلال SMSCs و MSCs و HLRs **بنية المحور**: توصيل عدة

ملاحظة: يمكنك تمكين كل من `sctp_handler` و `map_client_m3ua` في نفس الوقت إذا كنت بحاجة إلى كل من الاتصالات الصادرة ووظيفة STP الواردة.

استمرارية جدول التوجيه (Mnesia)

يتم تخزين جميع جداول التوجيه (الأقران، مسارات رمز النقطة، ومسارات العنوان العالمي) في **قاعدة بيانات Mnesia** لضمان الاستمرارية.

كيف يعمل التوجيه

1. **مسارات runtime.exs**: يتم تحميل المسارات المعرفة في `config/runtime.exs` تحت `peers` (أو `m3ua_peers` القديمة)، `m3ua_routes`، و `m3ua_gt_routes` عند بدء التطبيق
2. **مسارات واجهة الويب**: يتم تخزين المسارات المضافة عبر صفحة إدارة التوجيه في واجهة الويب في Mnesia
3. **دمج المسارات**: عند إعادة التشغيل، يتم دمج مسارات runtime.exs مع المسارات الموجودة في Mnesia (بدون تكرارات)
4. **الاستمرارية**: جميع المسارات المكونة عبر واجهة الويب **تستمر بعد إعادة تشغيل التطبيق**

نوع تخزين Mnesia

تحكم في كيفية تخزين جداول التوجيه. لمزيد من التفاصيل حول تكوين قاعدة البيانات، انظر معلمات قاعدة البيانات في مرجع التكوين.

```
config :omniss7,
  mnesia_storage_type: :disc_copies  # أو :ram_copies للاختبار
```

نوع التخزين	الوصف	الاستمرارية	حالة الاستخدام
`:disc_copies`	تخزين مدعوم بالقرص (افتراضي)	**تستمر بعد إعادة التشغيل**	بيئات الإنتاج
`:ram_copies`	فقط في الذاكرة	تفقد عند إعادة التشغيل	الاختبار، التطوير

الافتراضي: `:disc_copies`

موقع قاعدة بيانات Mnesia

تقوم Mnesia بتخزين جداول التوجيه في دليل Mnesia الخاص بالتطبيق:

- **الموقع**: `Mnesia.{node_name}/` (على سبيل المثال، `Mnesia.nonode@nohost/`)
- **الجداول** (جميعها تم إنشاؤها في `M3UARouting.init_database`):

الجدول	الغرض
`m3ua_peer`	نقاط نهاية اتصال M3UA / M2PA
`m3ua_route`	مسارات رمز النقطة (DPC → قرين، مع دعم القناع)
`m3ua_gt_route`	مسارات العنوان العالمي (البادئة → قرين، مع مطابقة وإعادة كتابة SSN/TT/NPI/NAI)
`m3ua_dest_state`	حالة التوافر لكل وجهة مدفوعة بإدارة مسار الإشارات MTP3 (TFP/TFA/TFR، Q.704 §13)
`ss7_screen_rule`	قواعد السماح/الرفض لتصفية البوابة (جدار حماية SCCP)

إدارة المسارات

لديك ثلاث خيارات لإدارة المسارات:

1. **runtime.exs** - تكوين ثابت يتم تحميله عند بدء التشغيل
2. **واجهة الويب** - إدارة تفاعلية للمسار (انظر دليل واجهة الويب)
3. **REST API** - إدارة المسار برمجيًا (انظر دليل API)

أفضل ممارسة: استخدم `runtime.exs` للتكوين الأساسي وواجهة الويب لتغييرات المسار الديناميكية أثناء التشغيل.

تكوين أق��ان M3UA

يمثل الأقران نقاط نهاية اتصال M3UA (STPs الأخرى، HLRs، MSCs، SMSCs). أضف الأقران إلى `config/runtime.exs`.

مثال على تكوين القرين

> **ملاحظة**: مفتاح التكوين `peers` هو المعيار الحالي. لا يزال المفتاح القديم `m3ua_peers` مدعومًا للتوافق مع الإصدارات السابقة.

```
config :omniss7,
  peers: [
    # اتصال خارجي بـ Partner STP (الدور: :client)
    %{
      peer_id: 1,                          # معرف فريد
      name: "Partner_STP_West",            # اسم وصفي
      role: :client,                       # :client للاتصال
الخارجي، :server للاتصال الداخلي
      local_ip: {10, 0, 0, 1},            # IP المحلي للربط
      local_port: 0,                       # 0 = تعيين منفذ
ديناميكي
      remote_ip: {10, 0, 0, 10},          # عنوان IP للقرين
البعيد
      remote_port: 2905,                   # منفذ القرين البعيد
      routing_context: 1,                  # سياق توجيه M3UA
      point_code: 100,                     # رمز النقطة لهذا
القرين
      network_indicator: :international    # :international أو
:national
    },

    # اتصال بـ HLR المحلي (الدور: :client)
    %{
      peer_id: 2,
      name: "Local_HLR",
      role: :client,
      local_ip: {10, 0, 0, 1},
      local_port: 0,
      remote_ip: {10, 0, 0, 20},
      remote_port: 2905,
      routing_context: 2,
      point_code: 200,
      network_indicator: :international
    },

    # اتصال داخلي من MSC البعيد (الدور: :server)
    # الاتصال الوارد STP تنتظر، بالنسبة للدور :server
    %{
      peer_id: 3,
      name: "Remote_MSC",
      role: :server,                       # قبول الاتصال الوارد
      remote_ip: {10, 0, 0, 30},          # عنوان IP المتوقع
      remote_port: 2905,                   # المنفذ المتوقع (0 =
```

```
(قبول من أي منفذ
      routing_context: 3,
      point_code: 300,
      network_indicator: :international
    },

    # (بدون تصفية المنفذ) اتصال داخلي مع منفذ مصدر ديناميكي
    %{
      peer_id: 4,
      name: "Dynamic_Client",
      role: :server,
      remote_ip: {10, 0, 0, 40},           # عنوان IP المتوقع
      remote_port: 0,                       # 0 = قبول الاتصالات من
أي منفذ مصد��
      routing_context: 4,
      point_code: 400,
      network_indicator: :international
    }
  ]
```

معلمات تكوين الند

المعلمة	النوع	مطلوب	الوصف
`peer_id`	عدد صحيح	نعم	معرف رقمي فريد للند
`name`	سلسلة	نعم	اسم قابل للقراءة البشرية للسجلات والمراقبة
`role`	ذرة	نعم	`:server` أو (خارجية) `:client` (داخلية)
`local_ip`	مجموعة أو قائمة	نعم (عميل)	عنوان IP المحلية للربط. (عناوين) مفرد: `{10, 0, 0, 1}` أو متعدد لـ SCTP multihoming: `[{10, 0, 0, 1}, {10, 0, 0, 2}]`
`local_port`	عدد صحيح	نعم (عميل)	المنفذ المحلي (0 للديناميكي)
`remote_ip`	مجموعة أو قائمة	نعم	**دور العميل**: مجموعة أو قائمة مفردة للند البعيد متعدد المنازل. **دور الخادم**: مجموعة مفردة فقط - عنوان IP الذي يتصل منه الند البعيد (انظر الملاحظة أدناه)
`remote_port`	عدد صحيح	نعم	منفذ الند البعيد (0 للداخل = قبول أي منفذ مصدر)
`routing_context`	عدد صحيح	نعم	معرف سياق التوجيه M3UA
`point_code`	عدد صحيح	نعم	رمز نقطة SS7 لهذا الند
`network_indicator`	ذرة	لا	`:national` أو `:international`

لدور الخادم (الاتصالات الواردة): عند قبول الاتصالات الواردة **SCTP Multihoming** من ند بعيد متعدد المنازل، تحتاج فقط إلى تحديد **عنوان IP المفرد** الذي يستخدمه الند البعيد لبدء اتصال SCTP (SCTP INIT). يجب أن تكون هذه **مجموعة مفردة**، وليس قائمة. سيكتشف SCTP تلقائيًا عناوين IP الأخرى متعددة المنازل للند أثناء مصافحة الارتباط. فقط لدور **العميل** عند الاتصال بند بعيد متعدد `remote_ip` استخدام تنسيق القائمة لـ المنازل.

لدور العميل (الاتصالات الصادرة): من أجل تكرار الشبكة عند **SCTP Multihoming** متعددة لكل من `remote_ip` و `local_ip` الاتصال بالندود البعيدة، يمكنك تكوين عناوين IP باستخدام القوائم. وهذا يمكّن من الفشل التلقائي إذا فشل أحد مسارات الشبكة. راجع دليل SCTP Multihoming للحصول على أمثلة تكوين مفصلة وأفضل الممارسات.

تصفية منفذ المصدر للاتصالات الواردة

في تصفية منفذ `remote_port` تتحكم معلمة `:server`)، الدور) بالنسبة لـ **الاتصالات الواردة** المصدر:

- قبول الاتصالات فقط من (`remote_port: 2905`، على سبيل المثال) **منفذ محدد** ذلك المنفذ المصدر المحدد
  - يوفر أمانًا إضافيًا من خلال التحقق من منفذ المصدر
  - استخدمه عندما يستخدم الند البعيد منفذ مصدر ثابت
- قبول الاتصالات من أي منفذ مصدر (`remote_port: 0`) **أي منفذ**
  - مفيد عندما يستخدم الند البعيد منافذ مصدر ديناميكية/عابرة
  - يتحقق فقط من عنوان IP المصدر
  - أكثر مرونة ولكن أقل أمانًا قليلاً

مثال:

```
# قبول فقط من 192.0.2.10:2905 (منفذ محدد)
%{
  peer_id: 1,
  name: "Strict_Peer",
  role: :server,
  remote_ip: {10, 5, 198, 200},
  remote_port: 2905,
  # ... تكوين آخر
}

# قبول من 192.0.2.10 مع أي منفذ مصدر
%{
  peer_id: 2,
  name: "Flexible_Peer",
  role: :server,
  remote_ip: {10, 5, 198, 200},
  remote_port: 0,  # قبول من أي منفذ مصدر
  # ... تكوين آخر
}
```

دعم بروتوكول M2PA

يدعم OmniSS7 كل من بروتوكولات **M2PA** و **M3UA** لنقل إشارات SS7.

ما هو M2PA؟

M2PA (طبقة التكيف بين مستخدم MTP2) هو بروتوكول موحد من قبل IETF (RFC 4165) لنقل رسائل SS7 MTP3 عبر الشبكات IP باستخدام SCTP.

الاختلافات الرئيسية: M2PA مقابل M3UA

الميزة	M3UA	M2PA
العمارة	عميل/خادم (ASP/SGW)	نظير إلى نظير
حالة الاستخدام	بوابة بين IP و SS7	روابط مباشرة من نقطة إلى نقطة
إدارة حالة الرابط	على مستوى التطبيق (ASPUP/ASPAC)	على نمط MTP2 (محاذاة، إثبات، جاهز)
أرقام التسلسل	لا يوجد تسلسل داخلي	24-بت BSN/FSN للتسليم المرتب
نشر نموذجي	بوابة SS7 إلى IP، STP	روابط إشارات مباشرة بين العقد
RFC	RFC 4666	RFC 4165

إرشادات اختيار البروتوكول

التوصية: استخدم M3UA بشكل افتراضي. استخدم M2PA فقط عند الحاجة المحددة.

متى يجب استخدام M3UA (موصى به)

M3UA هو البروتوكول الموصى به لمعظم النشر:

- **نشر STP**: تنفيذ نقاط نقل الإشارات القياسية
- **وظائف البوابة**: ربط شبكات SS7 بإشارات قائمة على IP
- **اتصالات عناصر الشبكة**: ربط MSCs و HLRs و SMSCs وعناصر الشبكة الأخرى بـ STP الخاص بك
- **بوابة الإشارات (SGW)**: بوابة مركزية تقبل الاتصالات من عدة خوادم تطبيقات
- **التوبولوجيات المرنة**: هياكل عميل/خادم مع تحكم مركزي
- **شبكات متعددة البائعين** (RFC 4666): معيار صناعي مدعوم على نطاق واسع

استخدم M3UA لربط عناصر الشبكة (HLR و MSC و SMSC و VLR وما إلى ذلك) بـ STP الخاص بك.

متى يجب استخدام M2PA (حالات خاصة فقط)

يجب استخدام M2PA فقط في سيناريوهات محددة:

- **STP إلى STP روا��ط**: اتصالات مباشرة من نقطة إلى نقطة بين نقاط نقل الإشارات في شبكة متعددة STP
- **استبدال TDM القديم**: استبدال روابط SS7 TDM التقليدية عندما يتطلب النظام البعيد M2PA بشكل محدد
- **توافق MTP2 مطلوب**: عند الاتصال بالأنظمة القديمة التي تفرض إدارة حالة الرابط على نمط MTP2
- **متطلبات الشريك**: عندما يتطلب شريك أو اتصال بشكل محدد بروتوكول M2PA

مهم: لا تستخدم M2PA لربط عناصر الشبكة (HLR و MSC و SMSC) بـ STP - الخاص بك. تم تصميم M3UA بدلاً من ذلك. حيث تعمل كلا الجانبين كعقد STP للاتصالات بين M2PA استخدم M3UA توجيه.

تكوين أقران M2PA

يتم تكوين أقران M2PA بنفس طريقة تكوين أقران M3UA، مع إضافة معلمة `protocol`.

تكوين نظير M2PA

أضف أقران M2PA إلى تكوين `peers` في `config/runtime.exs` (تشارك أقران M3UA و M2PA نفس قسم التكوين، والتي يتم تمييزها بواسطة معلمة `protocol`):

المعلمات الرئيسية لـ M2PA:

المعلمة	القيمة	ا��وصف
`protocol`	`:m2pa`	يحدد بروتوكول M2PA (يفترض `:m3ua` إذا تم حذفه)
`role`	`:client` أو `:server`	اتجاه الاتصال
`local_port`	عدد صحيح	منفذ SCTP المحلي (المنفذ القياسي لـ M2PA هو **3565**)
`remote_port`	عدد صحيح	منفذ SCTP البعيد (المنفذ القياسي لـ M2PA هو **3565**)
`point_code`	عدد صحيح	رمز النقطة الخاص بك
`adjacent_point_code`	عدد صحيح	رمز نقطة الند البعيد (خاص بـ M2PA)
`send_sltm`	Boolean	يتحكم في سلوك SLTM (انظر تكوين SLTM أدناه)
`network_indicator`	ذرة	`:national` أو `:international` - يجب أن يتطابق مع الند البعيد

ملاحظة: يستخدم M2PA كمعيار صناعي **المنفذ 3565** (لـ 2905 مختلف عن المنفذ M3UA).

تكوين SLTM

هي **SLTA (إقرار اختبار رابط الإشارات)** و **SLTM (رسالة اختبار رابط الإشارات)** تستخدم M2PA للتحقق من الاتصال من النهاية إلى النهاية بعد أن يصل رابط MTP3 رسائل صيانة SLTA ويستجيب الآخر بـ SLTM يرسل أ��د الجانبين، ITU-T Q.707 وفقًا لـ READY. إلى حالة لتأكيد أن الرابط يعمل.

سلوك SLTM

تتحكم معلمة `send_sltm` في أي جانب يبدأ اختبار SLTM:

قيمة `send_sltm`	السلوك
`true`	عندما يصبح الرابط جاهزًا، ننتظر SLTM نرسل SLTA
`false`	ننتظر أن يرسل الند SLTM، ونستجيب بـ SLTA
غير محدد (افتراضي)	يتبع Q.707: يرسل المستجيب SCTP SLTM، وينتظر المبادر SCTP

السلوك الافتراضي (متوافق مع Q.707):

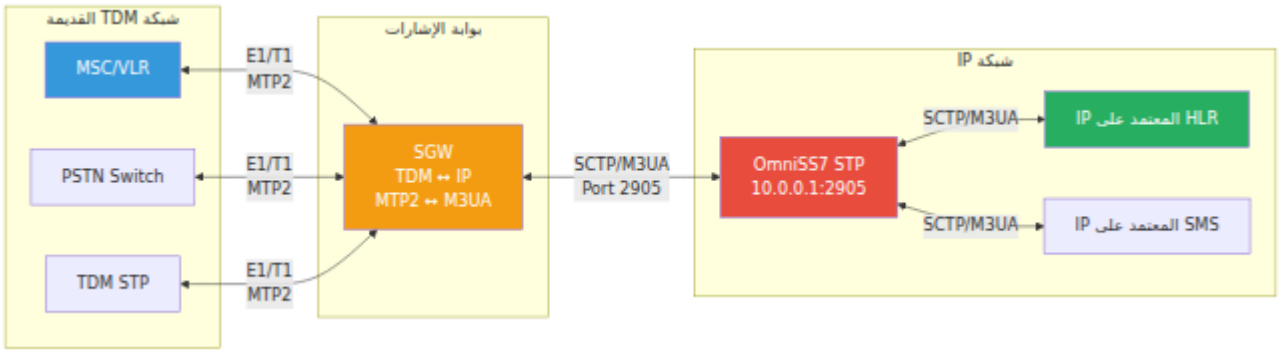
- إذا كان `initiate_connection: true` (مبادر SCTP/عميل) → ننتظر أن يرسل الند SLTM
- إذا كان `initiate_connection: false` (مستجيب SCTP/خادم) → نرسل SLTM

متى يجب تجاوز الافتراضات SLTM

تجاوز السلوك الافتراضي عند الاتصال بمعدات لا تتبع تقاليد Q.707:

```
# مثال: فرض جانبنا لإرسال SLTM بغض النظر عن دور SCTP
%{
  peer_id: 100,
  name: "Partner_STP",
  protocol: :m2pa,
  role: :client,
  local_ip: {10, 0, 0, 1},
  local_port: 3565,
  remote_ip: {10, 0, 0, 2},
  remote_port: 3565,
  point_code: 7415,
  adjacent_point_code: 15528,
  network_indicator: :international,
  send_sltm: true  # تجاوز: نرسل SLTM المبادر على الرغم من أننا
SCTP
}
```

تدفق رسالة SLTM



استكشاف مشاكل SLTM

العرض: يصل الرابط إلى READY ولكن لا تتدفق حركة المرور

تحقق من السجلات عن SLTM. قد يكون كلا الجانبين في انتظار الآخر لإرسال:

```
"waiting for peer to send SLTM"
```

الحل: قم بتكوين `send_sltm: true` على أحد الجانبين لكسر الجمود.

العرض: تم إرسال SLTM ولكن لم يتم استلام SLTA

الأسباب المحتملة:

1. `adjacent_point_code` غير صحيح (تم إرسال SLTM مع DPC خاطئ)
2. عدم تطابق مؤشر الشبكة (`:national` مقابل `:international`)
3. الند ا��بعيد غير مكون للاستجابة إلى رمز النقطة الخاص بنا

الحل: تحقق من أن `adjacent_point_code` لـ STP (المكون) يتطابق مع رمز النقطة الخاص بنا في `sctp_handler.point_code`.

حالات رابط M2PA

تتقدم روابط M2PA عبر عدة حالات أثناء التهيئة:

1. **Down** - لم يتم إنشاء اتصال
2. **Alignment** - مرحلة المزامنة الأولية (~1 ثانية)
3. **Proving** - التحقق من جودة الرابط (~2 ثانية)
4. **Ready** - الرابط نشط وجاهز لحركة المرور

تضمن تقدم حالة الرابط إشارات موثوقة قبل تبادل حركة المرور.

إدارة أقران M2PA عبر واجهة الويب

توفر صفحة **التوجيه** في واجهة الويب دعمًا كاملاً لإدارة أقران M2PA:

1. **انتقل** إلى صفحة التوجيه
2. **حدد** علامة التبويب "الأقران"
3. **انقر** على "إضافة نظير جديد"
4. **اختر** "M2PA (RFC 4165)" من قائمة البروتوكولات المنسدلة
5. **املأ** تكوين الند:
   - اسم الند (معرف وصفي)
   - البروتوكول: M2PA
   - الدور: عميل أو خادم
   - رمز النقطة (رمز PC الخاص بك)
   - عناوين IP المحلية/البعيدة
   - المنافذ المحلية/البعيدة (عادة M2PA 3565 لـ)
   - مؤشر الشبكة (دولي أو وطني)
6. **انقر** على "حفظ الند"

تعرض جدول الأقران نوع البروتوكول مع ترميز الألوان:

- أقران M3UA - **أزرق**
- أقران M2PA - **أخضر**

سلوك توجيه M2PA

تتكامل أقران M2PA في نظام التوجيه OmniSS7 بسلاسة مع:

- **طرق رمز النقطة**: تعمل M3UA و M2PA بشكل متطابق لـ
- **طرق العنوان العالمي**: مدعومة بالكامل على روابط M2PA
- **أولوية الطريق**: يمكن خلط أقران M3UA و M2PA في نفس جداول التوجيه
- **إعادة توجيه الرسائل**: يمكن أن تصل الرسائل على M3UA، وتُوجه إلى M2PA والعكس صحيح

مقاييس M2PA

يوفر M2PA مقاييس شاملة من Prometheus لمراقبة صحة الرابط وحركة المرور:

مقاييس الحركة:

- `m2pa_messages_sent_total` - إجمالي رسائل MTP3 المرسلة لكل رابط
- `m2pa_messages_received_total` - إجمالي رسائل MTP3 المستلمة لكل رابط
- `m2pa_bytes_sent_total` - إجمالي البايتات المرسلة عبر M2PA
- `m2pa_bytes_received_total` - إجمالي البايتات المستلمة عبر M2PA

تُوسم جميع مقاييس الحركة بـ: `link_name`, `point_code`, `adjacent_pc`

مقاييس حالة الرابط:

- `m2pa_link_state_changes_total` - انتقالات حالة الرابط (DOWN → ALIGNMENT → PROVING → READY)
  - التسميات: `link_name`, `from_state`, `to_state`

مقاييس الأخطاء:

- `m2pa_errors_total` - إجمالي الأخطاء حسب النوع
  - `decode_error` - فشل فك تشفير رسالة M2PA
  - `encode_error` - فشل تشفير رسالة M2PA
  - `sctp_send_error` - فشل نقل SCTP
  - التسميات: `link_name`, `error_type`

مقاييس الوصول:

- نقطة نهاية Prometheus: `http://your-server:9090/metrics`
- يتم تسجيل المقاييس تلقائيًا عند بدء التطبيق

أفضل الممارسات لـ M2PA

1. **اختيار المنفذ**: استخدم المنفذ 3565 لـ M2PA (معيار صناعي)
2. **مراقبة الرابط**: راقب تغييرات حالة الرابط عبر المقاييس
3. **قواعد جدار الحماية**: تأكد من السماح لـ SCTP (بروتوكول IP 132)
4. **رموز ال��قطة**: تأكد من تكوين رموز النقطة المجاورة بشكل صحيح على كلا الجانبين
5. **مؤشر الشبكة**: يجب أن يتطابق بين الأقران (دولي أو وطني)
6. **الاختبار**: استخدم صفحة اختبار التوجيه للتحقق من الاتصال بعد التكوين

متطلبات مقبس M2PA

M2PA يستخدم مقابس مشتركة SCTP.SocketHandler. تستخدم جميع أقران M2PA لإدارة المقابس، مما يسمح لمتعدد الأقران بمشاركة نفس منفذ SCTP.SocketHandler تلقائيًا بكفاءة SCTP.

المتطلبات:

- يجب تمكين SCTP.SocketHandler وتشغيله
- يجب تكوين المقبس المشترك قبل بدء أقران M2PA

مثال:

```
# تمكين SCTP.SocketHandler
sctp_handler: %{
  enabled: true,
  local_ip: {10, 179, 4, 10},
  local_port: 3565,
  point_code: 100
}

# تستخدم أقران M2PA المشترك المقبس تلقائيًا
peers: [
  %{
    peer_id: 1,
    name: "M2PA_Link_1",
    protocol: :m2pa,
    role: :client,
    local_ip: {10, 179, 4, 10},
    local_port: 3565,
    remote_ip: {10, 179, 4, 20},
    remote_port: 3565,
    point_code: 100,
    adjacent_point_code: 200
  },
  %{
    peer_id: 2,
    name: "M2PA_Link_2",
    protocol: :m2pa,
    role: :client,
    local_ip: {10, 179, 4, 10},
    local_port: 3565,           # نفس المنفذ المشترك عبر الأقران
    remote_ip: {10, 179, 4, 30},
    remote_port: 3565,
    point_code: 100,
    adjacent_point_code: 300
  }
]
```

متطلبات SCTP:

- التسليم المرتب (بدون علامة `unordered`)
- PPID 5 (معرف M2PA وفقًا لـ RFC 4165)

تكوين توجيه رمز النقطة

رمز النقطة الوجهة يوجه توجيه رمز النقطة الرسائل بناءً على **(DPC)** في رأس MTP3.

فهم رموز النقطة في كومة بروتوكول SS7

توجد رموز النقطة في طبقات مختلفة من كومة بروتوكول SS7. فهم هذا التمييز مهم:

طبقات كومة البروتوكول:

```
┌─────────────────────────────────────────────┐
│  طبقة التطبيق (SCCP/TCAP/MAP)               │
├─────────────────────────────────────────────┤
│  طبقة MTP3                                  │ ← توجيه بيانات المستخدم
│  - تسمية التوجيه: DPC، OPC، SLS             │ ← تستخدم لتوجيه STP
│  - ثمانية معلومات الخدمة (SIO)              │
├─────────────────────────────────────────────┤
│  M3UA أو M2PA (طبقة التكيف)                 │ ← بروتوكول النقل
│  - بيانات البروتوكول (تحتوي على MTP3)       │
│  - إدارة الشبكة (DUNA/DAVA)                 │ ← حالة الشبكة
├─────────────────────────────────────────────┤
│  SCTP (نقل)                                 │
└─────────────────────────────────────────────┘
```

نوعان من رموز النقطة:

1. **رموز النقطة في طبقة MTP3** (تستخدم للتوجيه):
   - تقع في تسمية توجيه MTP3 (DPC، OPC)
   - موجودة في معلمة بيانات بروتوكول M3UA (التاج 528)
   - موجودة في رسائل بيانات مستخدم M2PA
   - **يستخدم STP هذه القيم DPC لاتخاذ قرارات التوجيه**
   - تحدد المكان الذي يتم تسليم الرسالة إليه في النهاية
2. **رموز النقطة في طبقة M3UA** (تستخدم لإدارة الشبكة):
   - موجودة في رسائل إدارة M3UA (DUNA، DAVA، SCON، DUPU)
   - تشير إلى رموز النقطة المتأثرة بحالة الشبكة

- تخبر الأقران بالوجهات المتاحة/غير المتاحة
- لا تستخدم لتوجيه بيانات المستخدم

كيف يعمل توجيه STP:

- **M3UA DATA بالنسبة لرسائل**: يستخرج STP رسالة MTP3 من معلمة بيانات البروتوكول (M3UA) والذي يحتوي على تسمية، أو حقل بيانات المستخدم (M2PA). يتم استخدام DPC من طبقة MTP3 للبحث عن توجيه MTP3 (DPC، OPC، SLS) الطرق.
- **M2PA بالنسبة لرسائل بيانات مستخدم**: يستخرج STP رسالة MTP3 من حقل بيانات المستخدم M2PA، ثم يقرأ DPC من تسمية توجيه MTP3.
- **M3UA رسائل إدارة**: تحتوي رسائل إدارة الشبكة (DUNA، DAVA، SCON) على رموز النقطة المتأثرة في طبقة M3UA لإشارة حالة الشبكة بين الأقران.

طرق رمز النقطة الأساسية

أضف طرقًا إلى `config/runtime.exs`:

```
config :omniss7,
  m3ua_routes: [
    # توجيه كل حركة المرور لرمز PC 100 1 إلى الند (STP الشريك)
    %{
      dest_pc: 100,                  # رمز النقطة الوجهة
      peer_id: 1,                    # الند الذي سيتم التوجيه من
خلاله
      priority: 1,                   # الأولوية (الأقل = أعلى
أولوية)
      network_indicator: :international
      # mask: 14                     # 14 اختياري: الافتراضي هو
(مطابقة دقيقة)
    },

    # توجيه كل حركة المرور لرمز PC 200 2 إلى الند (HLR المحلي)
    %{
      dest_pc: 200,
      peer_id: 2,
      priority: 1,
      network_indicator: :international
    },

    # مع طرق أساسية واحتياطية PC 300 :مثال على تحميل التوازن
    %{
      dest_pc: 300,
      peer_id: 3,                    # الطريق الأساسي
      priority: 1,
      network_indicator: :international
    },
    %{
      dest_pc: 300,
      peer_id: 4,                    # الطريق الاحتياطي (رقم
أولوية أعلى)
      priority: 2,
      network_indicator: :international
    }
  ]
```

ملاحظة: حقل `mask` اختياري ويفترض أنه 14 (مطابقة دقيقة). حدد `mask` فقط عندما تحتاج إلى توجيه يعتمد على النطاق (انظر قسم أقنعة رمز النقطة أدناه).

منطق التوجيه

1. يستقبل STP رسالة M3UA DATA مستخدم بيانات أو رسالة M2PA
2. يستخرج STP **رسالة MTP3** من بيانات بروتوكول (M3UA) المستخدم (M2PA)
3. يقرأ STP **رمز النقطة الوجهة (DPC)** من تسمية توجيه MTP3
4. يبحث في جدول التوجيه عن DPC المطابق (مع مراعاة الأقنعة)
5. إذا كانت هناك طرق متعددة، يختار الطريق مع **أكثر قناع محدد** (أعلى قيمة قناع)، ثم **أقل رقم أولوية**
6. يلف رسالة MTP3 في بيانات مستخدم M3UA DATA أو بيانات M2PA للند الوجهة
7. يوجه الرسالة إلى الند المقابل
8. إذا كان الند المختار معطلاً، يحاول الطريق التالي ذي الأولوية الأعلى

أقنعة رمز النقطة

رموز النقطة هي قيم 14 بت (النطاق 0-16383). بشكل افتراضي، تتطابق الطرق مع رمز نقطة واحد بدقة (قناع `/14`). ومع ذلك، يمكنك استخدام **أقنعة رمز النقطة** لإنشاء طرق تتطابق مع **نطاقات** من رموز النقطة.

فهم الأقنعة

يحدد القناع عدد **أعلى بتات ذات دلالة** التي يجب أن تتطابق بين رمز النقطة الوجهة للطريق و لرسالة الواردة. يمكن أن تكون البتات المتبقية أي قيمة، مما ينشئ نطاقًا من رموز النقطة DPC المتطابقة.

جدول مرجع القناع:

القناع	رموز النقطة المتطابقة	حالة الاستخدام
`/14`	1 PC (مطابقة دقيقة)	وجهة واحدة (افتراضي)
`/13`	2 PCs	نطاق صغير
`/12`	4 PCs	نطاق صغير
`/11`	8 PCs	نطاق صغير
`/10`	16 PCs	نطاق متوسط
`/9`	32 PCs	نطاق متوسط
`/8`	64 PCs	نطاق متوسط
`/7`	128 PCs	نطاق متوسط-كبير
`/6`	256 PCs	نطاق كبير
`/5`	512 PCs	نطاق كبير
`/4`	1,024 PCs	نطاق كبير جدًا
`/3`	2,048 PCs	نطاق كبير جدًا
`/2`	4,096 PCs	نطاق كبير جدًا
`/1`	8,192 PCs	نصف جميع PCs
`/0`	16,384 PCs	جميع PCs (طريق افتراضي/احتياطي)

أمثلة على أقنعة رمز النقطة

ملاحظة: حقل `mask` (مطابقة 14) هو **اختياري** في جميع الأمثلة. إذا تم حذفه، فإنه يفترض أنه (دقيقة).

مثال 1: رمز نقطة واحد (سلوك افتراضي)

```
# موصى به لرمز (واحد PC) بدون حقل القناع
%{
  dest_pc: 1000,
  peer_id: 1,
  priority: 1,
  network_indicator: :international
}
# فقط: يتطابق - 14 القناع الافتراضي هو PC 1000

# قناع صريح (نفس النتيجة)
%{
  dest_pc: 1000,
  peer_id: 1,
  priority: 1,
  mask: 14,                                  # مطابقة دقيقة صريحة
  network_indicator: :international
}
# فقط: يتطابق PC 1000
```

مثال 2: نطاق صغير

```
%{
  dest_pc: 1000,
  peer_id: 2,
  priority: 1,
  mask: 12,                                  # يتطابق مع 4 PCs
  network_indicator: :international
}
# يتطابق: PC 1000، 1001، 1002، 1003
```

مثال 3: نطاق متوسط

```
%{
  dest_pc: 1000,
  peer_id: 3,
  priority: 1,
  mask: 8,                                  # 64 يتطابق مع PCs
  network_indicator: :international
}
# يتطابق: PC 1000-1063 (رمز نقطة متتالية 64)
```

مثال 4: طريق افتراضي/احتياطي

```
%{
  dest_pc: 0,
  peer_id: 4,
  priority: 10,                             # أولوية منخفضة (رقم
(عالي
  mask: 0,                                  # يتطابق مع جميع PCs
  network_indicator: :international
}
# يتطابق: جميع رموز النقطة (0-16383)
# استخدم كطريق شامل/افتراضي مع أولوية منخفضة
```

دمج الطرق المحددة والمقنعة

يمكنك دمج الطرق المحددة مع الطرق المقنعة لتوجيه مرن:

```
config :omniss7,
  m3ua_routes: [
    # مسار محدد لـ PC 1000 (يأخذ الأولوية)
    %{
      dest_pc: 1000,
      peer_id: 1,
      priority: 1,
      network_indicator: :international
      # القناع الافتراضي هو 14 (مطابقة دقيقة)
    },

    # مسار نطاق لـ PCs 1000-1063
    %{
      dest_pc: 1000,
      peer_id: 2,
      priority: 1,
      mask: 8,                          # 64 يتطابق مع PC
      network_indicator: :international
    },

    # مسار افتراضي/احتياطي لجميع PCs الأخرى
    %{
      dest_pc: 0,
      peer_id: 3,
      priority: 10,                     # أولوية منخفضة
      mask: 0,                          # يتطابق مع جميع PCs
      network_indicator: :international
    }
  ]
```

قرار التوجيه لـ DPC 1000:

1. `14/` يتطابق مع قناع (PC 1000 بالضبط) - (الأكثر تحديدًا) **محدد**
2. `8/` يتطابق أيضًا مع قناع (PC 1000-1063 نطاق) - (أقل تحديدًا) تم تجاهله
3. `0/` يتطابق أيضًا مع قناع (جميع PCs) - (الأقل تحديدًا) تم تجاهله

قرار التوجيه لـ DPC 1015:

1. `14/` لا يتطابق مع قناع (فقط PC 1000)
2. `8/` يتطابق مع قناع (PC 1000-1063 نطاق) - (أكثر تطابقًا تحديدًا) **محدد**
3. `0/` يتطابق أيضًا مع قناع (جميع PCs) - (أقل تحديدًا) تم تجاهله

قرار التوجيه لـ DPC 5000:

1. `14/` لا يتطابق مع قناع
2. `8/` لا يتطابق مع قناع
3. `0/` يتطابق مع قناع (جميع PCs) - (التطابق الوحيد، مسار احتياطي) **محدد**

أفضل الممارسات

1. **تجنب** `mask` **للوجهات الفردية**: لمطابقات رموز النقاط الدقيقة، تجنب حقل `mask` (`14/` الافتراضي هو تمامًا)
2. **استخدم** `14/` **بشكل صريح فقط عند الحاجة**: حدد `mask: 14` فقط عندما تحتاج إلى توضيح ذلك في الوثائق أو عند الخلط مع مسارات النطاق
3. **استخدم أقنعة النطاق لكتل الشبكة**: قم بتوجيه مقاطع الشبكة بالكامل إلى أقران محددين مع أقنعة من `0/` إلى `13/`
4. **استخدم** `0/` **كاحتياطي**: أنشئ مسارًا افتراضيًا بأولوية منخفضة لالتقاط حركة المرور غير المطابقة
5. **الأكثر تحديدًا يفوز**: يقوم محرك التوجيه دائمًا باختيار المسار الأكثر تحديدًا (أعلى قيمة قناع) أولاً
6. **الأولوية ككسر لل��عادل**: إذا كانت هناك مسارات متعددة تحمل نفس القناع، فإن أقل رقم أولوية يفوز

تكوين توجيه العنوان العالمي (GT)

يمكن توجيه العنوان العالمي IMSI **توجيهًا قائمًا على المحتوى** باستخدام أرقام الهواتف أو قيم بدلاً من رموز النقاط. لمزيد من المعلومات حول ترجمة عنوان العنوان العالمي المتقدمة بناءً على الطرف المتصل/المتصل به، راجع دليل للعنوان العالمي NAT.

المتطلبات الأساسية

- تمكين توجيه GT: `enable_gt_routing: true` في `config/runtime.exs`

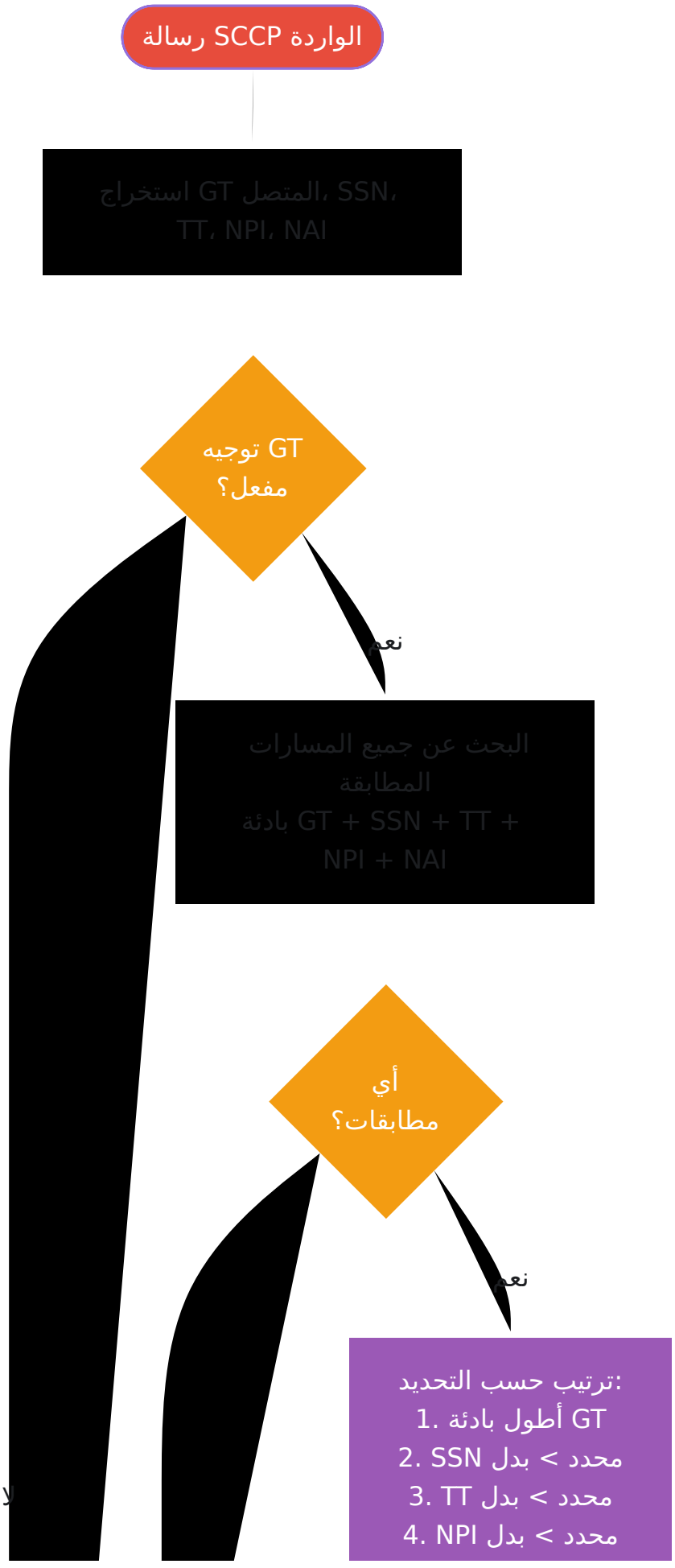
تكوين مسار GT

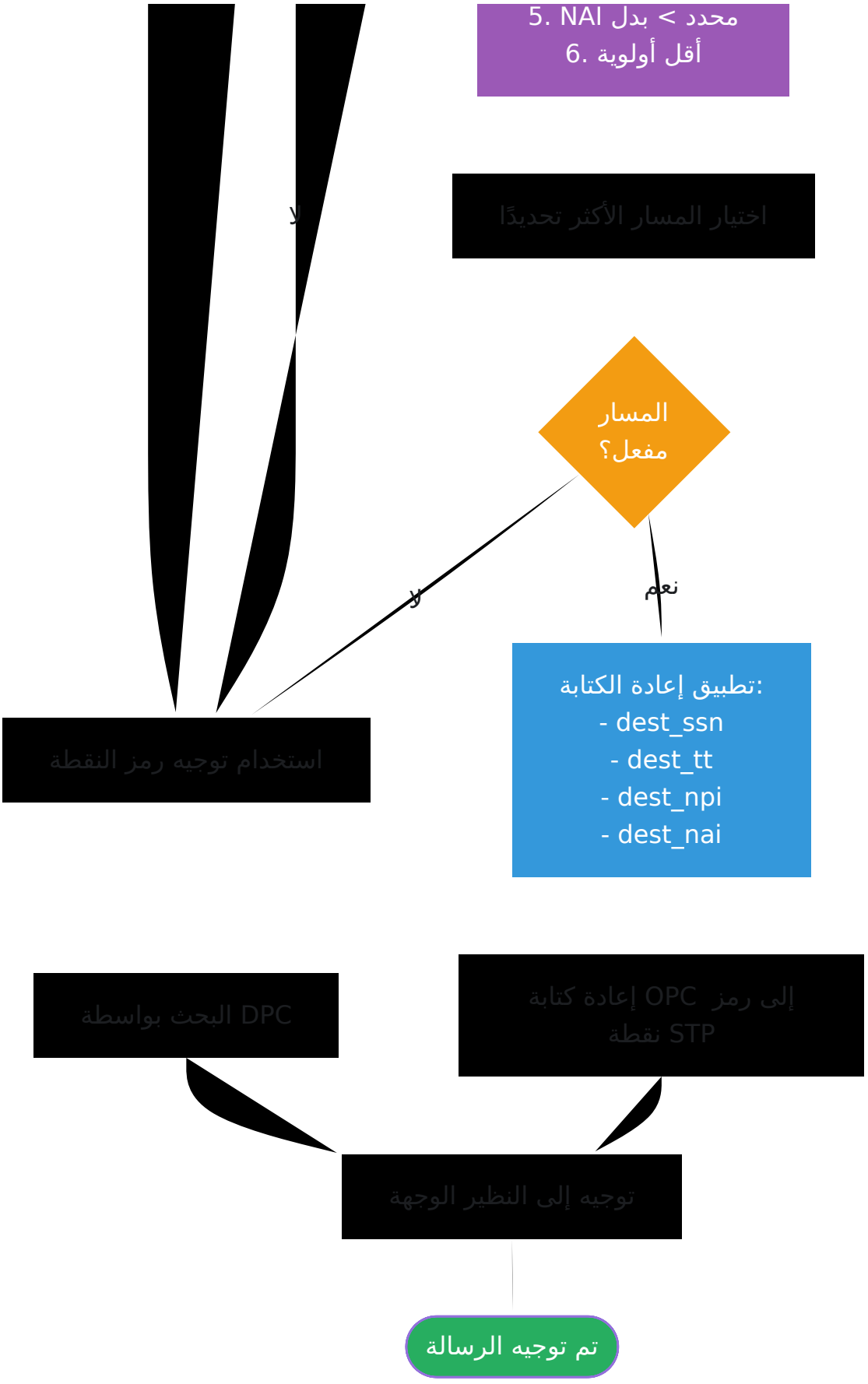
```
config :omniss7,
  # تمكين توجيه GT
  enable_gt_routing: true,

  m3ua_gt_routes: [
    # توجيه جميع الأرقام البريطانية (البادئة 44) إلى النظير 1
    %{
      gt_prefix: "44",                # بادئة العنوان العالمي للمطابقة
      peer_id: 1,                     # النظير الوجهة
      priority: 1,                    # الأولوية (أقل = أعلى)
      description: "أرقام المملكة المتحدة"  # وصف للتسجيل
    },

    # توجيه الأرقام الأمريكية (البادئة 1) إلى النظير 2
    %{
      gt_prefix: "1",
      peer_id: 2,
      priority: 1,
      description: "أرقام الولايات المتحدة"
    },

    # مسار أكثر تحديدًا: أرقام الهواتف المحمولة البريطانية التي تبدأ بـ 447
    %{
      gt_prefix: "447",               # أطول بادئة مطابقة تفوز
      peer_id: 3,
      priority: 1,
      description: "أرقام الهواتف المحمولة البريطانية"
    },

    # توجيه محدد SSN (اختياري)
    %{
      gt_prefix: "555",
      source_ssn: 8,                  # يتطابق فقط إذا كان SSN المصدر = 8 (SMSC)
      peer_id: 4,
      dest_ssn: 6,                    # إعادة كتابة SSN الوجهة إلى 6 (HLR)
      priority: 1,
      description: "حركة SMS 61 للبادئة"
```

```
    }
  ]
```

GT منطق توجيه

هذه العملية GT يتبع خوارزمية توجيه:


رسالة SCCP الواردة
استخراج GT المتصل، SSN، TT، NPI، NAI
توجيه GT مفعل؟
نعم
البحث عن جميع المسارات المطابقة
بادئة GT + SSN + TT + NPI + NAI
أي مطابقات؟
نعم
ترتيب حسب التحديد:
1. أطول بادئة GT
2. SSN بدل > محدد
3. TT بدل > محدد
4. NPI بدل > محدد



خطوات التوجيه:

1. حيث تتطابق البادئة مع بداية GT جميع مسارات STP **أطول بادئة مطابقة**: يجد العنوان العالمي

    - مثال: GT "447712345678" يتطابق مع كل من "44" و "447"، ولكن "447" يفوز (أطول مطابقة)

2. **مطابقة SSN** (اختياري):
    - إذا تم تحديد `source_ssn`، فإن المسار يتطابق فقط عندما يساوي SSN الطرف المتصل قيمة ذلك
    - إذا كان `source_ssn` هو `nil`، فإن المسار يتطابق مع أي SSN (بدل)
3. **مطابقة TT/NPI/NAI** (اختياري):
    - إذا تم تحديد `source_tt` أو `source_npi` أو `source_nai`، يجب أن تتطابق المسارات مع ��لك المؤشرات
    - تعمل القيم `nil` كبدائل (تتطابق مع أي قيمة)
4. **اختيار بناءً على التحديد**:
    - تفوز المسارات التي تحتوي على معايير مطابقة أكثر تحديدًا على البدائل
    - ترتيب الأولوية: طول بادئة GT → SSN → TT → NPI → NAI → رقم الأولوية
5. **إعادة كتابة المؤشرات** (اختياري):
    - إذا تم تحديد `dest_ssn` أو `dest_tt` أو `dest_npi` أو `dest_nai`، يقوم STP بإعادة كتابة تلك المؤشرات
    - مفيد لتطبيع البروتوكول وترابط الشبكة
6. **العودة إلى رمز النقطة**:
    - إذا لم يتطابق أي مسار GT، يعود STP باستخدام رمز النقطة إلى توجيه DPC

توجيه GT المتقدم: نوع الترجمة، NPI، و NAI

يدعم STP عنوان GT التوجيه والتحويل بناءً على مؤشرات SSN، و GT بالإضافة إلى مطابقة بادئة SCCP:

- **نوع الترجمة (TT)**: يحدد خطة الترقيم ونوع العنوان
- **مؤشر خطة الترقيم (NPI)**: يحدد خطة الترقيم (مثل ISDN، تيليكس، بيانات)
- **مؤشر طبيعة العنوان (NAI)**: يحدد تنسيق العنوان (مثل دولي، وطني، مشترك)

المطابقة (مؤشرات المصدر)

يمكن أن تتطابق المسارات على مؤشرات الرسالة الواردة:

- `source_tt`: مطابقة الرسائل مع نوع الترجمة المحدد
- `source_npi`: مطابقة الرسائل مع مؤشر خطة الترقيم المحدد
- `source_nai`: مطابقة الرسائل مع مؤشر طبيعة العنوان المحدد
- القيمة `nil` = (تتطابق مع أي قيمة) بدل

التحويل (مؤشرات الوجهة)

يمكن أن تعيد المسارات كتابة المؤشرات عند التوجيه:

- `dest_tt`: تحويل نوع الترجمة إلى قيمة جديدة
- `dest_npi`: تحويل مؤشر خطة الترقيم إلى قيمة جديدة
- `dest_nai`: تحويل مؤشر طبيعة العنوان إلى قيمة جديدة
- القيمة `nil` = (لا تحويل) الحفاظ على القيمة الأصلية

اختيار بناءً على التحديد

عندما تتطابق مسارات متعددة، يتم اختيار المسار الأكثر تحديدًا باستخدام ترتيب الأولوية هذا:

1. أطول مطابقة بادئة GT
2. المصدر المحدد على SSN البديل SSN
3. المصدر المحدد على TT البديل TT
4. المصدر المحدد على NPI البديل NPI

5. NAI المصدر المحدد على NAI البديل
6. أقل رقم أولوية

أمثلة التكوين

```
config :omniss7,
  enable_gt_routing: true,

  m3ua_gt_routes: [
    # المثال 1: مطابقة وتحويل نوع الترجمة
    %{
      gt_prefix: "44",
      peer_id: 1,
      source_tt: 0,      # مطابقة TT=0 (غير معروف)
      dest_tt: 3,        # تحويل إلى TT=3 (وطني)
      priority: 1,
      description: "تحويل: أرقام المملكة المتحدة TT 0→3"
    },

    # المثال 2: مطابقة NPI محدد وتحويل NAI
    %{
      gt_prefix: "1",
      peer_id: 2,
      source_npi: 1,     # مطابقة NPI=1 (ISDN/الهاتف)
      source_nai: 4,     # مطابقة NAI=4 (دولي)
      dest_nai: 3,       # تحويل إلى NAI=3 (وطني)
      priority: 1,
      description: "أرقام الولايات المتحدة: NAI دولي→وطني"
    },

    # المثال 3: توجيه SSN ومؤشر مجمع
    %{
      gt_prefix: "33",
      source_ssn: 8,     # مطابقة حركة SMSC
      source_tt: 0,      # مطابق TT=0
      dest_ssn: 6,       # إعادة كتابة SSN إلى HLR
      dest_tt: 2,        # تحويل إلى TT=2
      dest_npi: 1,       # تعيين NPI=1 (ISDN)
      dest_nai: 4,       # تعيين NAI=4 (دولي)
      peer_id: 3,
      priority: 1,
      description: "فرنسي: تطبيع كامل SMS"
    },

    # المثال 4: TT بديل، NPI محدد
    %{
      gt_prefix: "49",
      source_tt: nil,    # أي مطابقة TT (بديل)
```

```
      source_npi: 6,     # مطابقة NPI=6 (بيانات)
      dest_npi: 1,       # تحويل إلى NPI=1 (ISDN)
      peer_id: 4,
      priority: 1,
      description: "تطبيع شبكة البيانات الألمانية"
    }
  ]
```

القيم الشائعة لـ TT/NPI/NAI

نوع الترجمة (TT):

- `0` = غير معروف
- `1` = دولي
- `2` = وطني
- `3` = محدد الشبكة

مؤشر خطة الترقيم (NPI):

- `0` = غير معروف
- `1` = الهاتف/ISDN (E.164)
- `3` = بيانات (X.121)
- `4` = تيليكس (F.69)
- `6` = الهاتف المحمول (E.212)

مؤشر طبيعة العنوان (NAI):

- `0` = غير معروف
- `1` = رقم مشترك
- `2` = محجوز للاستخدام الوطني
- `3` = رقم مهم وطني
- `4` = رقم دولي

مثال على قرار التوجيه

لرسالة واردة مع:

- GT: "447712345678"

- SSN: 8
- TT: 0
- NPI: 1
- NAI: 4

مع هذه المسارات المكونة:

```
# المسار A: TT بديل
%{gt_prefix: "447", peer_id: 1, priority: 1}

# المسار B: TT محدد
%{gt_prefix: "447", source_tt: 0, peer_id: 2, priority: 1}

# المسار C: TT محدد + NPI
%{gt_prefix: "447", source_tt: 0, source_npi: 1, peer_id: 3,
priority: 1}
```

النتيجة: يتم اختيار المسار C (يتطابق مع GT + TT + NPI الأكثر تحديدًا)

يتم توجيه الرسالة مع تحويل المؤشرات وفقًا لقيم `dest_nai` في `dest_npi` و `dest_tt` المسار C.

أمثلة توجيه GT

GT المتصل	SSN المصدر	TT	NPI	NAI	المسار المتطابق	السبب
447712345678	6	-	-	-	"447" → النظير 3	أطول بادئة مطابقة
441234567890	6	-	-	-	"44" → النظير 1	مطابقة بادئة، لا يوجد مسار أكثر تحديدًا
12125551234	6	-	-	-	"1" → النظير 2	مطابقة بادئة لأرقام الولايات المتحدة
555881234567	8	-	-	-	"555" (SSN 8) → النظير 4	مطابقة GT + SSN، يعيد كتابة SSN إلى 6
555881234567	6	-	-	-	"555" (SSN بديل) → النظير X	مطابقة GT، لا إعادة كتابة SSN
441234567890	6	0	1	4	"44" (TT=0) → النظير 1	مطابقة GT + TT، يحول TT إلى 3
12125551234	8	0	1	4	"1" (TT=0، NPI=1، NAI=4)	الأكثر تحديدًا: مطابقة GT+TT+NPI+NAI

حالات الاستخدام العملية لتوجيه TT/NPI/NAI

1. **تطبيع الترابط الشبكي**
    - قد تستخدم الشبكات المختلفة تقنيات مؤشرات مختلفة
    - تحويل المؤشرات عند نقطة الترابط لضمان التوافق
    - مثال: تستخدم الشبكة الشريكة TT=0 وتستخدم شبكتك TT=1 للدولي
2. **تحويل البروتوكول**
    - تحويل بين خطط الترقيم عند التوجيه بين أنواع الشبكات المختلفة
    - مثال: توجيه من شبكة الهاتف المحمول (NPI=6) إلى PSTN (NPI=1)
3. **توحيد تنسيق العنوان**
    - تطبيع جميع حركة المرور الواردة لاستخدام قيم NAI متسقة
    - مثال: تحويل جميع التنسيقات الدولية (NAI=4) إلى التنسيق الوطني للتوجيه المحلي (NAI=3)
4. **توجيه محدد حسب الناقل**
    - توجيه بناءً على نوع الترجمة إلى مزودي الخدمة المختلفين
    - مثال: TT=0 يوجه إلى الناقل A، TT=2 يوجه إلى الناقل B
5. **دمج الأنظمة القديمة**
    - قد تستخدم الأنظمة الحديثة قيم مؤشرات مختلفة عن الأنظمة القديمة
    - تحويل في STP للحفاظ على التوافق مع الأنظمة القديمة

معالجة عبور SCCP

بجانب اختيار نظير الوجهة، يقوم STP بتنفيذ وظائف عبور طبقة SCCP المطلوبة من نقطة التحكم في الإشارة الحقيقية. تنطبق هذه على فئات الرسائل غير المتصلة التي يقوم STP بإعادة توجيهها: **UDT** (0x09)، **XUDT** (0x11)، و **LUDT** (0x13).

توجيه على SSN بعد ترجمة العنوان العالمي

عند **GT على مؤشر توجيه الطرف المتصل كمسار على STP** بشكل افتراضي، **يحافظ** هذا مقصود: اختيار الخطوة التالية بواسطة بادئة GT. إعادة توجيه رسالة تم مطابقتها بواسطة مسار

يجب على GT *توجيه* *GT* (وسيط)، *ترجمة* وليس *GT* *النهائية*. وفقًا لـ ITU-T Q.714 §2.4، حتى تقوم العقدة السفلية بأداء الترجمة. GT العقدة الوسيطة ترك مؤشر التوجيه كمسار على مع GT هناك يخطئ نتيجة وسيطة كنهائية. تحمل الرسالة بعد ذلك SSN التحويل إلى مسار على التي كانت MSS مثل) ولا تحتوي على رمز نقطة، وهو ما ترفضه العقدة السفلية SSN مسار على مع (الخاص بها GTT تتوقع تشغيل `providerAbort(abnormalDialogue)`.

هو المترجم النهائي STP **فقط** عندما يكون *هذا* SSN قم بتعيين مؤشر التوجيه إلى مسار على إضافي. هذا هو خيار صريح لكل GTT فقط لا يمكنها إجراء SSN ويوصل المسار إلى نقطة نهاية مسار عبر علامة `route_on_ssn: true` على مسار GT:

```
m3ua_gt_routes: [
  # توجيه العبور - يحافظ على المسار على GT (افتراضي)؛ تقوم العقدة
السفلية بأداء GTT
  %{gt_prefix: "44", peer_id: 2, priority: 1, description: "أرقام
المملكة المتحدة"},

  # فقط - التحويل إلى مسار على SSN التسليم النهائي إلى نقطة نهاية
SSN
  %{
    gt_prefix: "1234",
    peer_id: 1,
    priority: 1,
    dest_ssn: 8,
    route_on_ssn: true,
    description: "فقط SSN التسليم النهائي إلى MSC"
  }
]
```

على `route_on_ssn`) يمكن أيضًا تعيين العلامة لكل مسار عبر واجهة برمجة التطبيقات للتوجيه `POST /api/routing/gt-routes`) والافتراضي هو `false`. عند التمكين، ينطبق التحويل غير صفري SSN يحمل GT فقط على طرف متصل موجه حاليًا على (ITU-T Q.714 §2.4).

فرض عداد القفز (XUDT / LUDT)

الذي يحمي من حلقات التوجيه بين عقد نقل **SCCP عداد قفز** LUDT و XUDT تحمل رسائل SCCP. في كل قفزة عبور، يقوم STP:

1. **يقلل** عداد القفز في الرسالة المعاد توجيهها (Q.714 §2.4).

2. إذا كان تقليل العداد سيؤدي إلى **صفر**، يتم **التخلص** من الرسالة بدلاً من توجيهها (Q.714 §2.8.6).
3. يؤدي التخلص إلى زيادة `m3ua_stp_routing_failures_total{reason="hcounter_violation"}`.
4. إذا كانت الرسالة الأصلية قد طلبت العودة عند الخطأ، يتم إنشاء رسالة `XUDTS`/`LUDTS` للعود�إلى المنشئ (انظر أدناه).

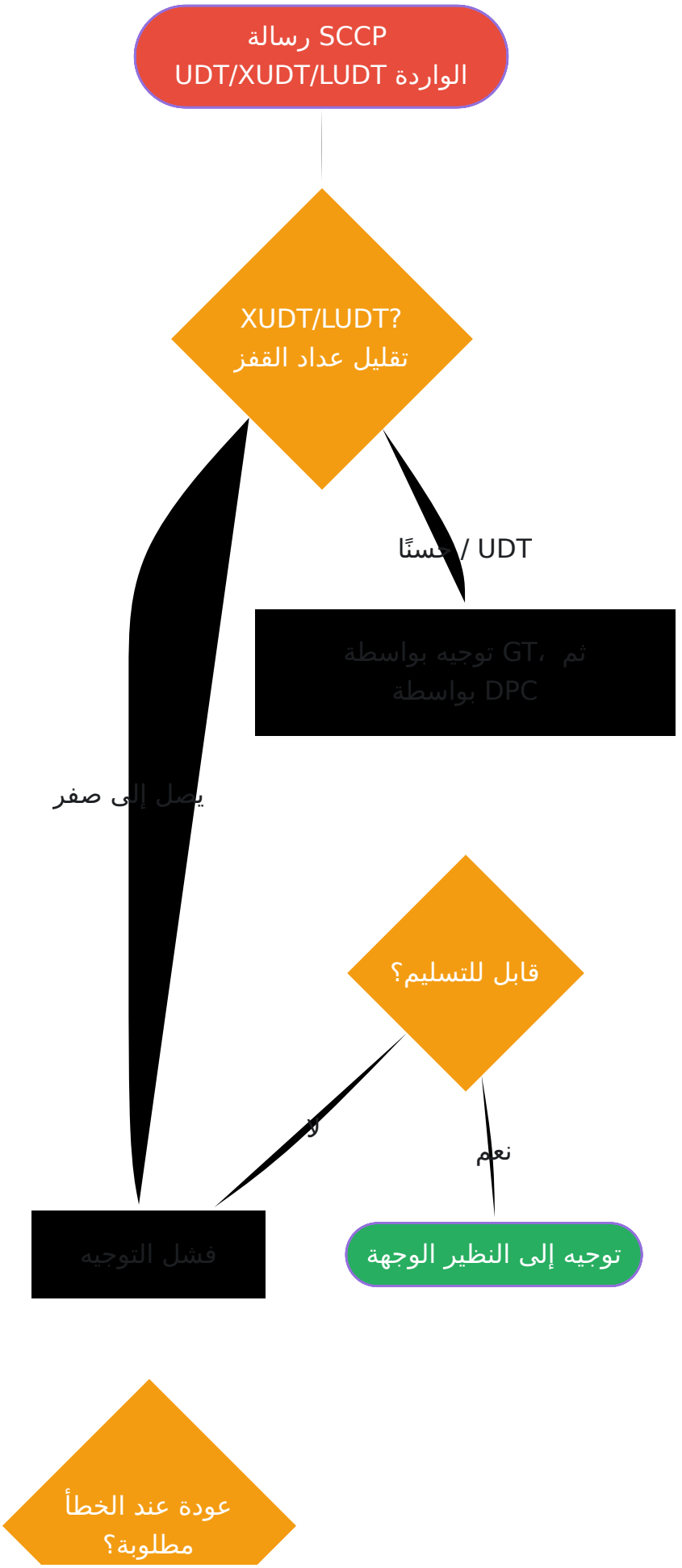
لا يحتوي UDT على عداد قفز ويتم توجيهه دون تغيير في هذه الخطوة.

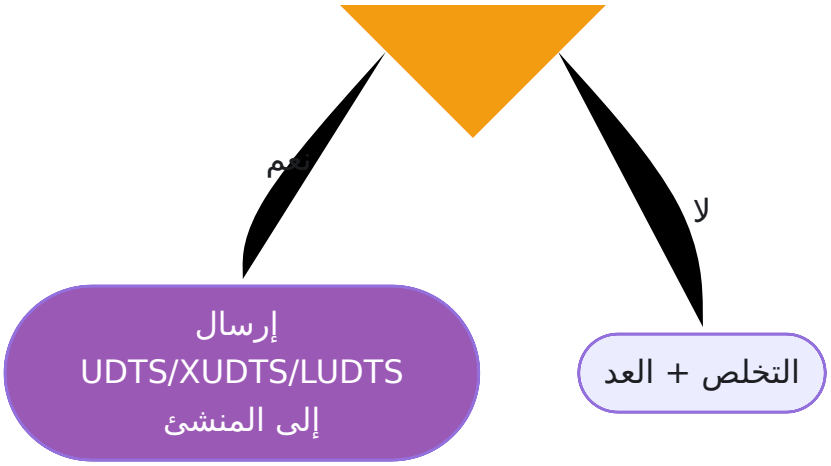
إعادة الرسالة عند فشل التوجيه (UDTS / XUDTS / LUDTS)

عندما لا يمكن تسليم رسالة غير متصلة (لا يوجد مسار، وجهة محظورة، استنفاد عداد القفز، إلخ.) **وإذا كان المنشئ قد حدد خيار العودة عند الخطأ**، يقوم SCCP للعودة ببناء رسالة STP ويرسلها مرة أخرى نحو المنشئ (ITU-T Q.714 §4.2):

الرسالة الأصلية	تم إرجاعها كـ
UDT (0x09)	UDTS (0x0A)
XUDT (0x11)	XUDTS (0x12)
LUDT (0x13)	LUDTS (0x14)

على سبيل المثال، فشل MTP لوجهة) تعكس سبب العودة المضمن في الرسالة سبب الفشل إذا لم يطلب المنشئ **عودة عند الخطأ**، يتم .(محظورة، أو انتهاك عداد القفز لعداد مستنفد ببساطة التخلص من الرسالة ويتم احتسابها.


رسالة SCCP
الواردة UDT/XUDT/LUDT
XUDT/LUDT؟
تقليل عداد القفز
UDT / حسنًا
ثم ، GT توجيه بواسطة
بواسطة DPC
يصل إلى صفر
قابل للتسليم؟
لا
نعم
فشل التوجيه
توجيه إلى النظير الوجهة
عودة عند الخطأ
مطلوبة؟



حالة الوجهة (إدارة مسار الإشارة)

يتتبع STP توفر كل رمز نقطة وجهة باستخدام رسائل إدارة مسار الإشارة MTP3، وفقًا لـ ITU-T Q.704 §13. يتم الاحتفاظ بالحالة في جدول Mnesia `m3ua_dest_state`؛ غياب صف يعني أن الوجهة **مسموح بها** (الافتراضي).

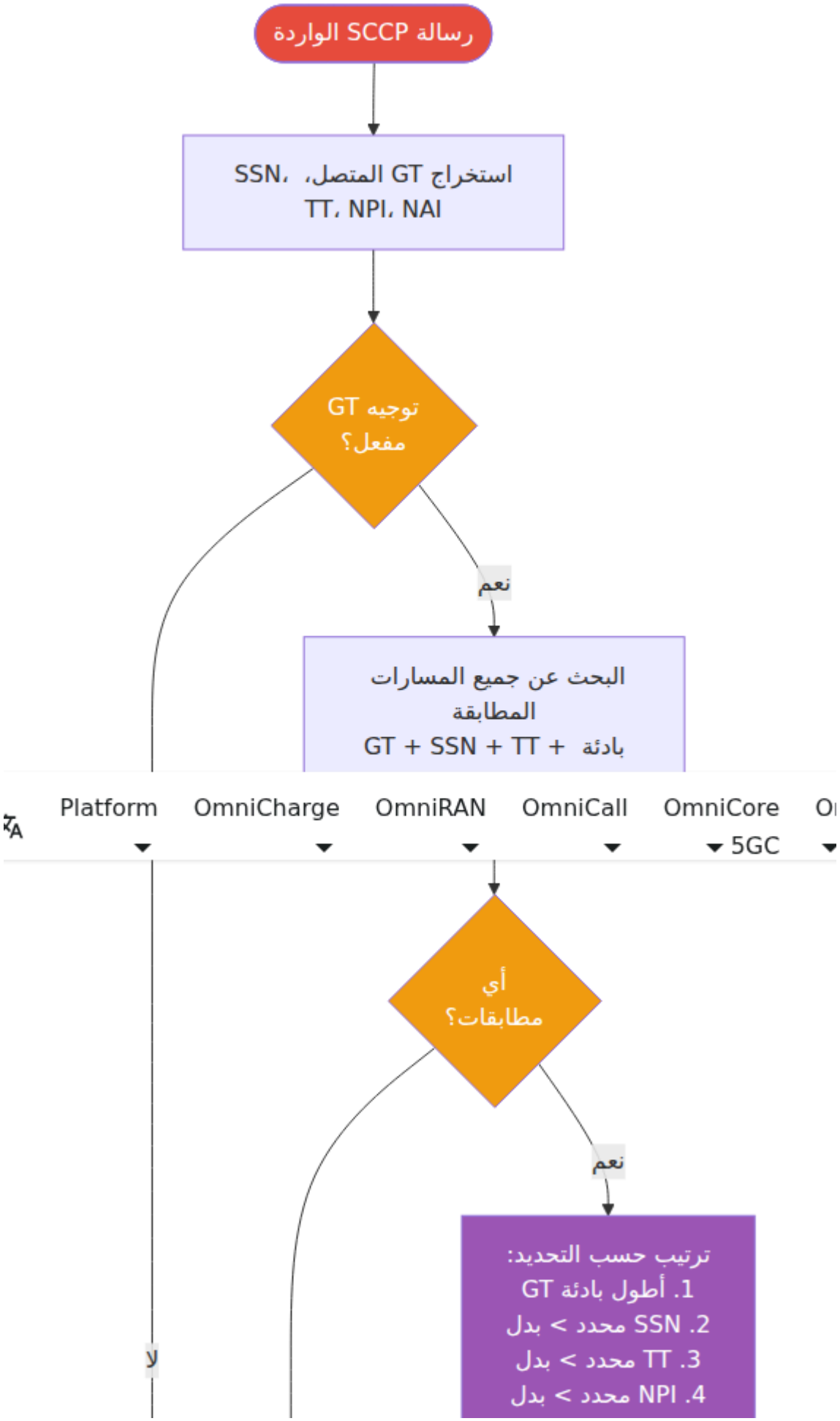
معالجة TFP / TFA / TFR

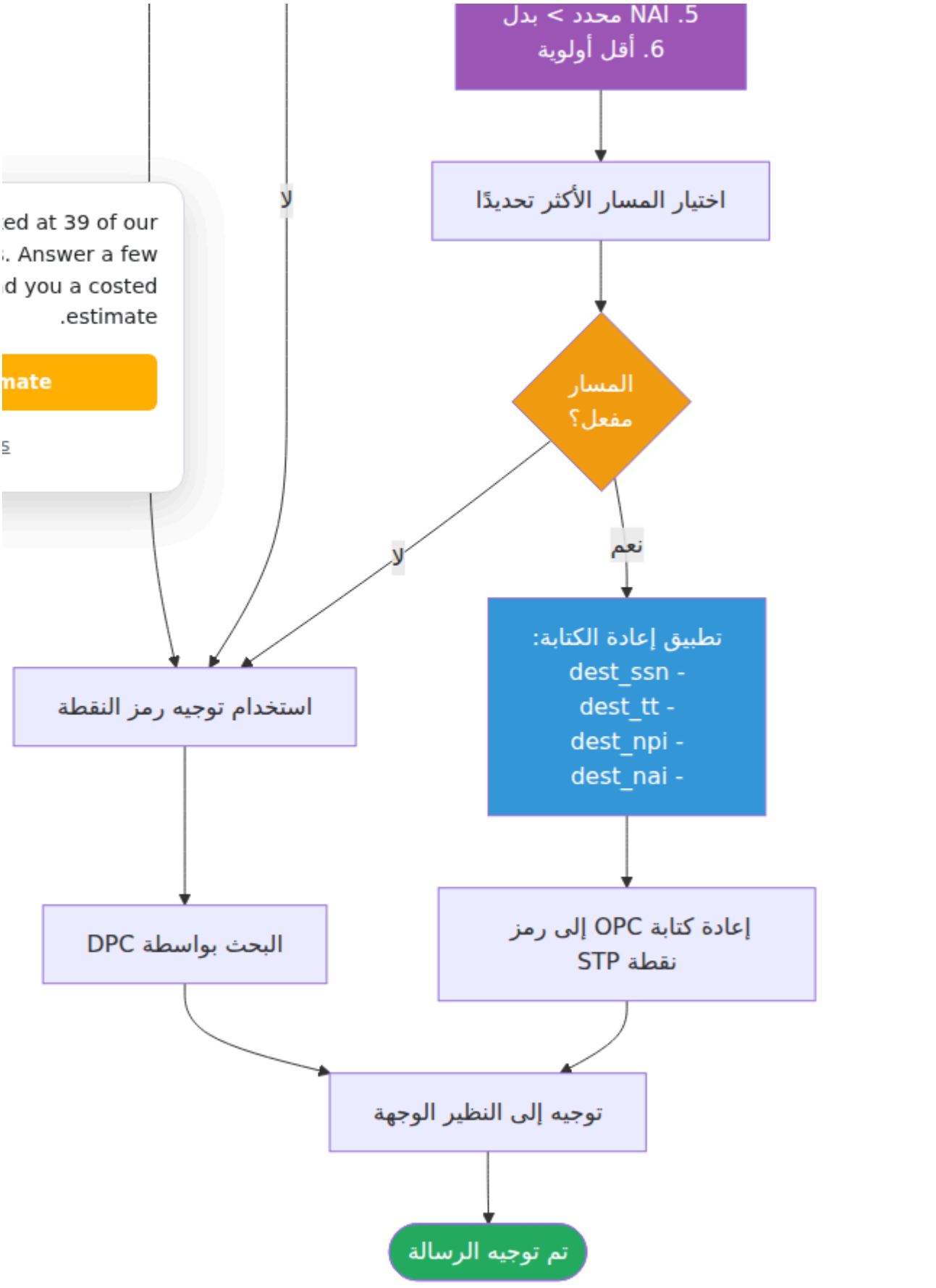
الرسالة	المعنى	التأثير على حالة STP
TFP (نقل محظور)	لم يعد بالإمكان الوصول إلى وجهة عبر هذا المسار	تم وضع علامة على الوجهة `:prohibited`
TFA (نقل مسموح)	يمكن الوصول إلى وجهة مرة أخرى	تم مسح حالة الوجهة (العودة إلى المسموح)
TFR (نقل مقيد)	يجب استخدام وجهة فقط عندما لا توجد بدائل	تم وضع علامة على الوجهة `:restricted`

التأثير على التوجيه

- أولاً من حالة الوجهة. يتم تخطي الوجهة STP أثناء البحث عن رمز النقطة، يتحقق ولا يتم توجيه `{:error, :prohibited}` المعلنة **نقل محظور**. يعود البحث الرسالة إلى هناك (Q.704 §13.2.3). هذا يزيد من `m3ua_stp_routing_failures_total{reason="prohibited"}`.

- من هذه الحالة الحقيقية بدلاً من "متاحة" (تدقيق الوجهة) **DAUD** يتم الرد على طلبات والوجهات غير القابلة، **DAVA** بشكل عام: يتم الإبلاغ عن الوجهات القابلة للوصول مع للوصول أو المحظورة مع **DUNA** (RFC 4666 §3.4.3).


رسالة SCCP الواردة
استخراج GT المتصل، SSN،
TT، NPI، NAI
توجيه GT
مفعل؟
نعم
البحث عن جميع المسارات
المطابقة
بادئة + GT + SSN + TT
Platform
OmniCharge
OmniRAN
OmniCall
OmniCore
5GC
أي
مطابقات؟
نعم
ترتيب حسب التحديد:
1. أطول بادئة GT
2. SSN محدد > بدل
3. TT محدد > بدل
4. NPI محدد > بدل


5. NAI محدد > بدل
6. أقل أولوية
اختيار المسار الأكثر تحديدًا
المسار
مفعل؟
نعم
لا
تطبيق إعادة الكتابة:
dest_ssn -
dest_tt -
dest_npi -
dest_nai -
استخدام توجيه رمز النقطة
البحث بواسطة DPC
إعادة كتابة OPC إلى رمز
نقطة STP
توجيه إلى النظير الوجهة
تم توجيه الرسالة
لا
ed at 39 of our
. Answer a few
d you a costed
.estimate
mate

جدار حماية (SCCP) فحص البوابة

يتيح فحص البوابة لـ STP **السماح أو رفض حركة المرور العابرة** قبل توجيهها. بدون ذلك، يقوم STP بإعادة توجيه كل رسالة بدون شروط. مع ذلك، يمكن للمشغلين حظر الإشارات غير المرغوب فيها عند البوابة. هذه هي السيطرة الأساسية على أمان SS7.

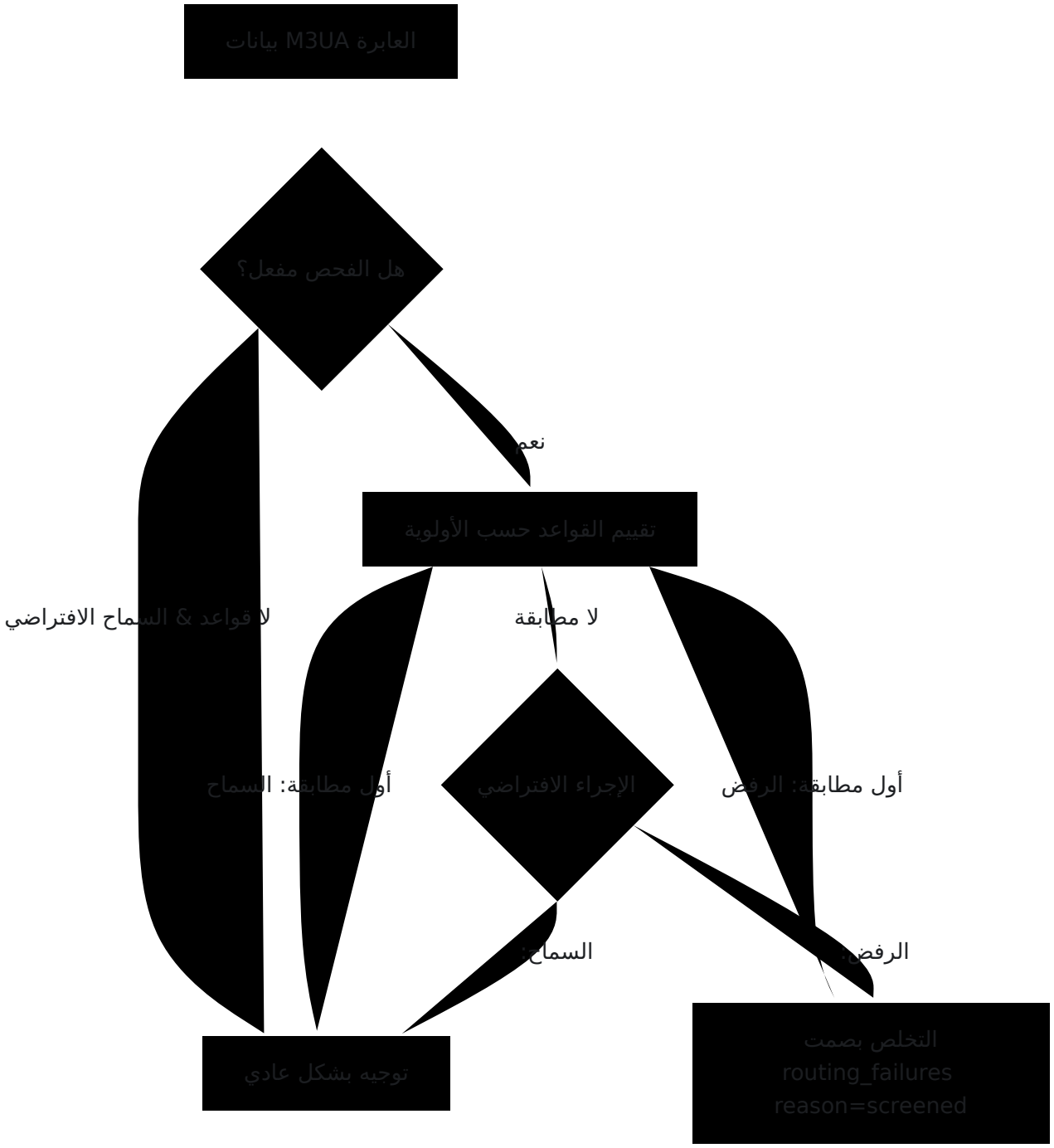
يتم تقييم كل رسالة بيانات M3UA أي مع القاعدة تتطابق القاعدة. ضد مجموعة مرتبة من القواعد العابرة
مجموعة من الصفات أدناه (صفة غير محددة هي بدل):

المطابقات	الصفة
النظير الذي وصلت منه الرسالة	النظير المصدر
رمز النقطة الأصلية	OPC
رمز النقطة الوجهة	DPC
MTP3 SI	مؤشر الخدمة
عنوان الطرف المتصل SCCP (مطابقة بادئة)	للطرف المتصل GTA بادئة
عنوان الطرف المتصل SCCP (مطابقة بادئة)	للطرف المتصل GTA بادئة
رقم النظام الفرعي SCCP	SSN

النتيجة: `السماح` أو `الرفض`. إذا (أولاً `priority` أقل قيمة) تحدد **أول قاعدة مطابقة مفعل** لم تتطابق أي قاعدة، فإن **الإجراء الافتراضي** القابل للتكوين ينطبق (`config :omniss7, :screening_default_action`، الافتراضي `:allow`).

تُسقط الرسالة المرفوضة **بصمت** وتُحتسب كـ `m3ua_stp_routing_failures_total{reason="screened"}`. لا يتم إنشاء عودة، UDTS/لأن جدار الحماية يجب ألا يكشف الحظر مرة أخرى إلى المرسل.

السلوك



تخطي الفحص بالكامل (تكلفة المسار الساخن صفر) عندما لا تكون هناك قواعد مكونة **و** الإجراء `:deny` ؛ يصبح نشطًا بمجرد وجود قاعدة أو تعيين الإجراء الافتراضي إلى`:allow` الافتراضي هو.

الإجراء الافتراضي

المعامل	النوع	الافتراضي	الوصف
`screening_default_action`	`:allow` | `:deny`	`:allow`	الإجراء المطبق عندما لا تتطابق أي قاعدة. تعيينه لوضع `:deny` إلى افتراضي يمنع الوصول (قائمة السماح).

```
config :omniss7,
  # حظر كل شيء باستثناء ما تسمح به القواعد صراحة.
  screening_default_action: :deny
```

تُخزن القواعد في جدول Mnesia `ss7_screen_rule` وتُدار برمجيًا (تحمل كل قاعدة أولوية، ونمط، و`allow`/`deny` وعلم `enabled`). راجع مرجع التكوين لمعلمة الإجراء الافتراضي.

ميزات إدارة المسارات

تعطيل المسارات

يمكن تعطيل المسارات مؤقتًا دون حذفها. هذا مفيد للاختبار، والصيانة، أو إدارة الحركة.

علم التمكين

اختياري `enabled` تدعم كل من مسارات رمز النقطة والعنوان العالمي علم:

```
config :omniss7,
  m3ua_routes: [
    # مسار نشط
    %{
      dest_pc: 100,
      peer_id: 1,
      priority: 1,
      network_indicator: :international,
      enabled: true  # (افتراضي إذا تم حذفه) المسار نشط
    },

    # (لا يتم تقييمه أثناء التوجيه) مسار معطل
    %{
      dest_pc: 200,
      peer_id: 2,
      priority: 1,
      network_indicator: :international,
      enabled: false  # المسار معطل
    }
  ],

  m3ua_gt_routes: [
    # مسار GT معطل
    %{
      gt_prefix: "44",
      peer_id: 1,
      priority: 1,
      description: "أرقام المملكة المتحدة - معطلة مؤقتًا",
      enabled: false
    }
  ]
```

السلوك الافتراضي:

- إذا لم يتم تحديد `enabled`، فإن المسارات افتراضيًا تكون `enabled: true`
- يتم تخطي المسارات المعطلة تمامًا أثناء البحث عن المسار
- استخدم واجهة الويب لتبديل المسارات تشغيل/إيقاف دون تعديل التكوين

حالات الاستخدام:

- اختبار تغييرات تدفق الحركة

- نوافذ صيانة مؤقتة
- اختبار A/B لمسارات توجيه مختلفة
- طرح تدريجي لمسارات جديدة

مسارات DROP: منع حلقات التوجيه

تقوم مسارات DROP (مع `peer_id: 0`) بالتخلص من الحركة بهدوء بدلاً من إعادة توجيهها. هذا يمنع حلقات التوجيه ويمكّن من تصفية الحركة المتقدمة.

تكوين مسارات DROP

```
config :omniss7,
  m3ua_routes: [
    # مسار DROP لرمز نقطة محدد
    %{
      dest_pc: 999,
      peer_id: 0,        # peer_id=0 يعني DROP
      priority: 1,
      network_indicator: :international
    }
  ],

  m3ua_gt_routes: [
    # مسار DROP لبادئة GT
    %{
      gt_prefix: "999",
      peer_id: 0,        # peer_id=0 يعني DROP
      priority: 99,
      description: "حظر نطاق الاختبار"
    }
  ]
```

كيف تعمل مسارات DROP

عندما تتطابق رسالة مع مسار DROP:

1. محرك التوجيه يحد `peer_id: 0`
2. يتم **التخلص من الرسالة بهدوء** (لا يتم إعادة توجيهها)

3. **سجل** يتم إنشاء **INFO**: `"DROP route matched for DPC 999"` أو `"DROP route matched for GT 999"`
4. تعود عملية البحث عن المسار `{:error, :dropped}`

مهم: يتم تسجيل الحركة المفقودة على مستوى INFO للمراقبة واستكشاف الأخطاء وإصلاحها.

حالة الاستخدام الشائعة: قائمة السماح للبريد

هي **قائمة السماح للبريد** - السماح فقط بأرقام DROP واحدة من أقوى استخدامات مسارات محددة ضمن نطاق كبير مع حظر جميع الأرقام الأخرى.

النمط:

1. إنشاء مسار DROP (مثل 99) **رقم أولوية عالية** للنطاق الكامل مع
2. إنشاء مسارات السماح المحددة لأرقام فردية مع **أرقام أولوية منخفضة** (مثل 1)
3. نظرًا لأن الأرقام ذات الأولوية المنخفضة يتم تقييمها أولاً، فإن مسارات السماح تتطابق قبل مسار DROP
4. أي رقم غير مسموح به صراحة يتم التقاطه بواسطة مسار DROP

سيناريو المثال:

لديك بادئة GT `1234` تمثل نطاقًا من 10,000 رقم (1234000000 - 1234999999)، لكنك تريد فقط توجيه 3 أرقام محددة: `1234567890`، `1234555000`، و `1234111222`.

```
config :omniss7,
  m3ua_gt_routes: [
    # مسار DROP (يتم تقييمه آخرًا) مع رقم أولوية عالية
    %{
      gt_prefix: "1234",
      peer_id: 0,              # DROP
      priority: 99,            # يتم = أولوية منخفضة = رقم عالي
تقييمه آخرًا
      description: "حظر جميع 1234* باستثناء الأرقام المسموح بها"
    },

    # مسارات السماح المحددة مع أرقام أولوية منخفضة (يتم تقييمها
أولاً)
    %{
      gt_prefix: "1234567890",
      peer_id: 1,              # توجيه إلى النظير 1
      priority: 1,             # يتم = أولوية عالية = رقم منخفض
تقييمه أولاً
      description: "رقم مسموح به 1"
    },
    %{
      gt_prefix: "1234555000",
      peer_id: 1,
      priority: 1,
      description: "رقم مسموح به 2"
    },
    %{
      gt_prefix: "1234111222",
      peer_id: 1,
      priority: 1,
      description: "رقم مسموح به 3"
    }
  ]
```

سلوك التوجيه:

الوارد GT	المسارات المطابقة	المسار المحدد	الإجراء
1234567890	✅ "1234567890" (أولوية 1) ✅ "1234" DROP (أولوية 99)	"1234567890" (الأكثر تحديدًا، أعلى أولوية)	**تم توجيهه إلى النظير 1**
1234555000	✅ "1234555000" (أولوية 1) ✅ "1234" DROP (أولوية 99)	"1234555000" (الأكثر تحديدًا، أعلى أولوية)	**تم توجيهه إلى النظير 1**
1234111222	✅ "1234111222" (أولوية 1) ✅ "1234" DROP (أولوية 99)	"1234111222" (الأكثر تحديدًا، أعلى أولوية)	**تم توجيهه إلى النظير 1**
1234999999	✅ "1234" DROP (أولوية 99)	"1234" DROP (المطابقة الوحيدة)	**تم حذفه + تم تسجيله**
1234000000	✅ "1234" DROP (أولوية 99)	"1234" DROP (المطابقة الوحيدة)	**تم حذفه + تم تسجيله**

النتيجة:

- ✅ يتم توجيه 3 أرقام محددة فقط إلى النظير 1
- ✅ يتم التخلص من جميع الأرقام الأخرى `1234*` بهدوء
- ✅ يتم تسجيل جميع الحركة المفقودة للمراقبة

السجلات التي تم إنشاؤها:

```
[INFO] DROP route matched for GT 1234999999
[INFO] DROP route matched for GT 1234000000
```

مسارات DROP لرموز النقطة

يعمل نفس نمط قائمة السماح للبريد لتوجيه رمز النقطة:

```
config :omniss7,
  m3ua_routes: [
    # DROP (1063-1000 :رمز نقطة 64) 8/ نطاق كامل
    %{
      dest_pc: 1000,
      peer_id: 0,
      priority: 99,
      mask: 8,
      network_indicator: :international
    },

    # السماح برموز PC محددة
    %{dest_pc: 1010, peer_id: 1, priority: 1, network_indicator:
:international},
    %{dest_pc: 1020, peer_id: 1, priority: 1, network_indicator:
:international},
    %{dest_pc: 1030, peer_id: 1, priority: 1, network_indicator:
:international}
  ]
```

النتيجة: يتم توجيه رموز PC 1010 و 1020 و 1030 فقط. يتم التخلص من جميع رموز PC الأخرى في النطاق 1000-1063.

مراقبة مسارات DROP

تحقق من السجلات:

```
# مراقبة الحركة المفقودة
tail -f logs/app.log | grep "DROP route matched"

# المخرجات المتوقعة:
[INFO] DROP route matched for GT 1234999999
[INFO] DROP route matched for DPC 1050
```

عبر واجهة الويب:

- انتقل إلى علامة تبويب **سجلات النظام**
- تصفية حسب مستوى `INFO`

- البحث عن "DROP route matched"

أفضل الممارسات:

1. ⚠ مراقبة السجلات بانتظام لضمان عدم حظر مسارات DROP للحركة الشرعية
2. استخدام حقول `description` الوصفية لتوثيق سبب حظر المسارات
3. استخدام أرقام أولوية عالية (90-99) لمسارات DROP لضمان كونها مسارات شاملة
4. اختبار سلوك مسار DROP قبل نشره في الإنتاج
5. إعداد تنبيهات لزيادة غير متوقعة في الحركة المفقودة

التوجيه المتقدم: التوجيه القائم على SSN وإعادة الكتابة

أرقام النظام الفرعي (SSN)

تحدد أرقام النظام الفرعي طبقة التطبيق:

- **SSN 6**: HLR (سجل الموقع المنزلي)
- **SSN 7**: VLR (سجل الموقع الزائر)
- **SSN 8**: MSC (مركز التبديل المحمول) / SMSC (مركز الرسائل القصيرة)
- **SSN 9**: GMLC (مركز موقع الهاتف المحمول البوابة)

مثال على التوجيه القائم على SSN

توجيه حركة SMS إلى HLR مختلفة بناءً على بادئة الرقم:

```
m3ua_gt_routes: [
  # توجيه SMS إلى المملكة المتحدة، إعادة HLR لأرقام المملكة المتحدة
كتابة SSN 8 من (SMSC) 6 إلى (HLR)
  %{
    gt_prefix: "44",
    source_ssn: 8,                  # تطابق SSN 8 الوارد (SMSC)
    peer_id: 1,
    dest_ssn: 6,                    # إعادة الكتابة إلى SSN 6
(HLR)
    priority: 1,
    description: "SMS المملكة المتحدة إلى HLR"
  },

  # توجيه حركة الصوت لأرقام المملكة المتحدة (SSN 6) دون إعادة
الكتابة
  %{
    gt_prefix: "44",
    source_ssn: 6,                  # تطابق SSN 6 الوارد (HLR)
    peer_id: 1,
    dest_ssn: nil,                  # لا توجد إعادة كتابة SSN
    priority: 1,
    description: "حركة الصوت في المملكة المتحدة"
  }
]
```

اختبار تكوين توجيه STP

بعد تكوين النظائر والمسارات، تحقق من تكوينك:

1. تحقق من حالة النظير

عبر واجهة الويب:

- انتقل إلى http://localhost
- تحقق من صفحة حالة M3UA
- تحقق من أن النظائر تظهر **الحالة: نشطة**

عبر وحدة IEx:

```
# الحصول على جميع حالات النظير
SCTP.SocketHandler.get_peers_status()

# المخرجات المتوقعة:
# [
#   %{peer_id: 1, name: "Partner_STP_West", status: :active,
point_code: 100, ...},
#   %{peer_id: 2, name: "Local_HLR", status: :active, point_code:
200, ...}
# ]
```

2. اختبار توجيه رمز النقطة

```
# اختبار إلى DPC 100 إرسال رسالة M3UA
test_payload = <<1, 2, 3, 4>>  # حمولة وهمية
SCTP.SocketHandler.route_by_pc(100, test_payload, 0)

# تحقق من السجلات لقرار التوجيه
# السجل المتوقع: OPC=... -> DPC=100 1 عبر النظير "توجيه الرسالة"
```

3. اختبار توجيه العنوان العالمي

```
# البحث عن مسار GT يدويًا
M3UARouting.lookup_peer_by_gt("447712345678")

# تعود {:ok, peer, rewrites}. `peer` هو سجل m3ua_peer 13 المكون من
عنصرًا؛
# `rewrites` هو خريطة لإعادة كتابة المؤشرات (nil = الاحتفاظ بالأصلي):
# {:ok, {:m3ua_peer, 3, "UK_Mobile_Peer", ...},
#       %{dest_ssn: nil, dest_tt: nil, dest_npi: nil, dest_nai:
nil}}

# البحث عن مسار GT مع SSN (مسار يعيد كتابة SSN 8 → 6)
M3UARouting.lookup_peer_by_gt("555881234567", 8)

# تظهر إعادة كتابة SSN في خريطة إعادة الكتابة، وليس كعدد عادي:
# {:ok, {:m3ua_peer, 4, "SMS_HLR_Peer", ...},
#       %{dest_ssn: 6, dest_tt: nil, dest_npi: nil, dest_nai:
nil}}
```

4. مراقبة مقاييس التوجيه

الوصول إلى مقاييس Prometheus في `/metrics`

المقاييس الرئيسية:

```
# الرسائل المستلمة لكل نظير
m3ua_stp_messages_received_total{peer_name="Partner_STP_West",point_c
1523

# الرسائل المرسلة لكل نظير
m3ua_stp_messages_sent_total{peer_name="Local_HLR",point_code="200"}

# حالات الفشل في التوجيه (حسب السبب)
m3ua_stp_routing_failures_total{reason="no_route"} 5
m3ua_stp_routing_failures_total{reason="no_gt_route"} 2
m3ua_stp_routing_failures_total{reason="no_active_peer"} 1
m3ua_stp_routing_failures_total{reason="prohibited"} 3
m3ua_stp_routing_failures_total{reason="hcounter_violation"} 4
m3ua_stp_routing_failures_total{reason="screened"} 7
```

مقاييس STP والمراقبة

المقاييس المتاحة

مقاييس حركة المرور لكل نظير:

- `m3ua_stp_messages_received_total` - إجمالي الرسائل المستلمة من كل نظير
  - التسميات: `peer_name`, `point_code`
- `m3ua_stp_messages_sent_total` - إجمالي الرسائل المعاد توجيهها إلى كل نظير
  - التسميات: `peer_name`, `point_code`

مقاييس فشل التوجيه:

- `m3ua_stp_routing_failures_total` - عدد حالات فشل التوجيه حسب السبب
  - التسميات: `reason`

`reason`	المعنى
`no_route`	لم يتم العثور على مسار رمز النقطة للوجهة
`no_gt_route`	لم يتم العثور على مسار عنوان عالمي، وفشل توجيه رمز النقطة أيضًا
`no_active_peer`	تم مطابقة مسار، لكن نظيره غير معروف/غير نشط
`prohibited`	الوجهة محظورة من النقل (تم استلام TFP، Q.704 §13)
`hcounter_violation`	وصل عداد القفز إلى الصفر على SCCP XUDT/LUDT (Q.714 §2.8.6)

تفسير المقاييس

- **أعداد الرسائل العالية**: تشير إلى تدفق حركة نشط

- **حالات فشل التوجيه**: تشير إلى المسارات المفقودة، وجهة غير قابلة للوصول، أو رسالة حلقة
  - `no_route`: لم يتم العثور على مسار رمز النقطة للوجهة
  - `no_gt_route`: لم يتم العثور على مسار عنوان عالمي، وفشل توجيه رمز النقطة أيضًا
  - `no_active_peer`: يوجد مسار لكن النظير المستهدف متوقف/غير معروف
  - `prohibited`: الوجهة معلمة على أنها محظورة من النقل بواسطة إدارة مسار الإشارة
  - `hcounter_violation`: تم استنفاد عداد القفز؛ تم التخلص من الرسالة (وتم إرجاعها إذا تم تعيين العودة عند الخطأ)

استكشاف الأخطاء باستخدام المقاييس

السيناريو: لا حركة تصل إلى الوجهة

1. تحقق مما إذا كانت الرسائل تُستقبل:

```
m3ua_stp_messages_received_total{peer_name="Source_Peer"}
> 0
```

2. تحقق مما إذا كانت الرسائل تُرسل:

```
m3ua_stp_messages_sent_total{peer_name="Dest_Peer"} > 0
```

3. تحقق من حالات فشل التوجيه:

```
m3ua_stp_routing_failures_total{reason="no_route"} > 0
```

الحل: إذا كانت حالات فشل التوجيه مرتفعة، أضف المسارات المفقودة في التكوين.

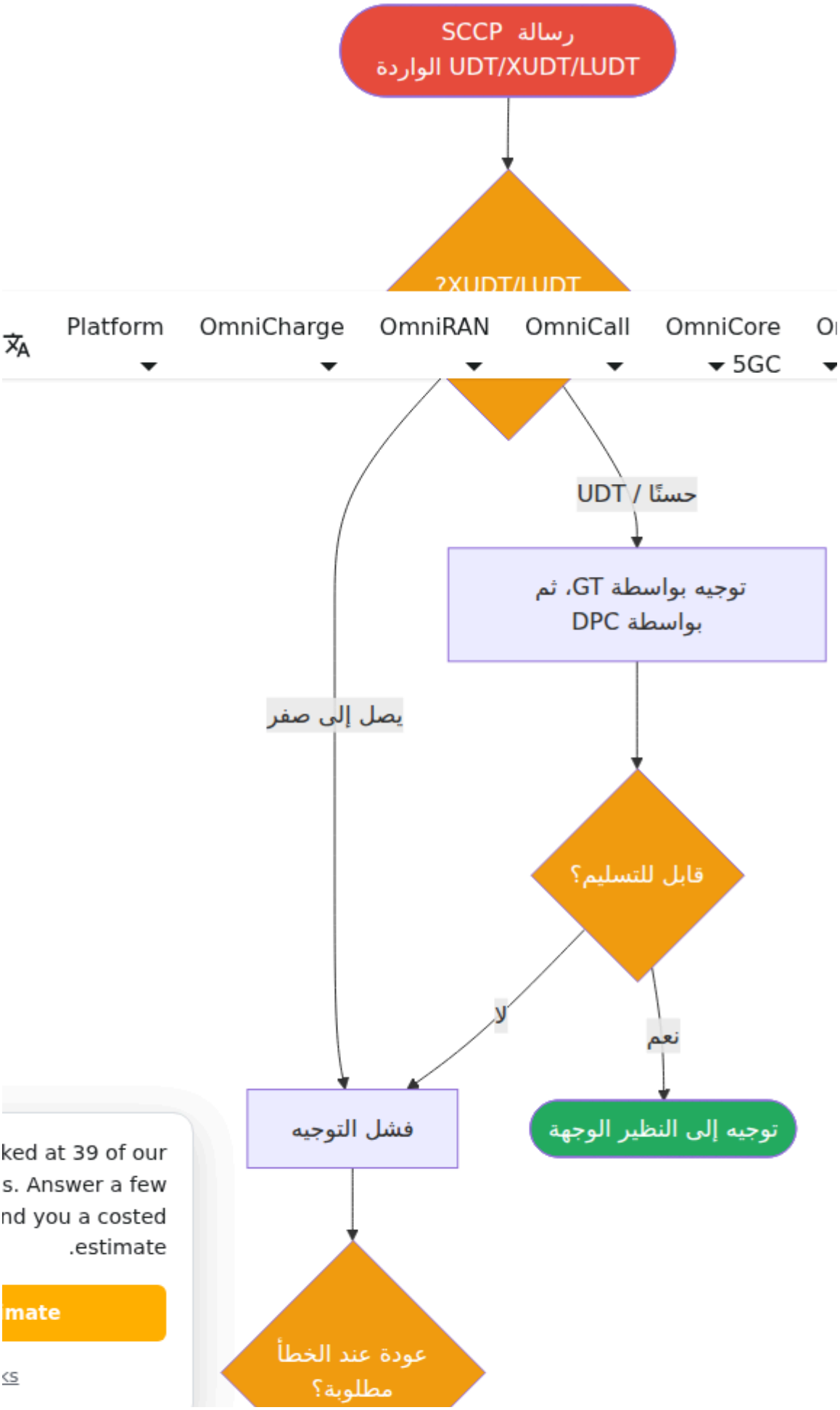
مراقبة حالة نظير M3UA

فهم M3UA

M3UA (طبقة تكييف مستخدم MTP3) هو بروتوكول يسمح بنقل إشارات SS7 عبر الشبكات IP باستخدام SCTP.

حالات اتصال M3UA

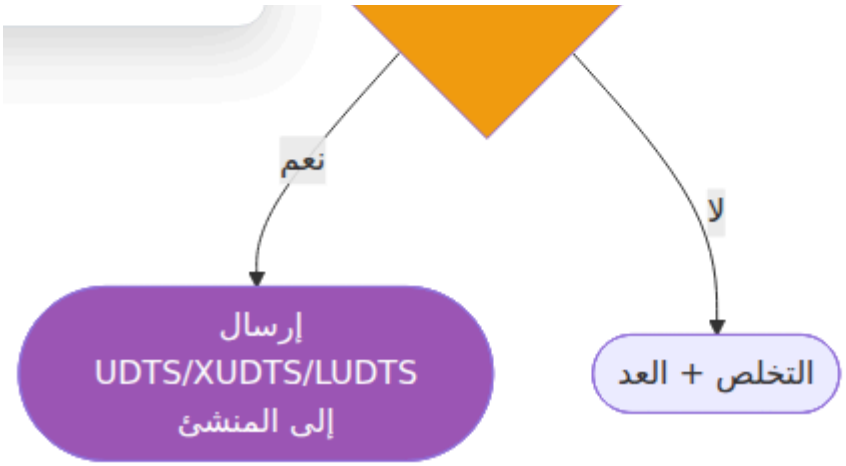
تتقدم اتصالات M3UA عبر عدة حالات:



ked at 39 of our
s. Answer a few
nd you a costed
.estimate

mate

cs



وصف الحالات:

- **DOWN** - لا يوجد اتصال SCTP
- **CONNECTING** - اتصال SCTP قيد التقدم
- **ASPUP_SENT** - في انتظار تأكيد ASPUP
- **INACTIVE** - ASP نشط لكن لا يحمل حركة مرور بنشاط (يمكن الانتقال إلى ACTIVE عبر ASPAC)
- **ASPAC_SENT** - في انتظار تأكيد ASPAC
- **ACTIVE** - جاهز للحركة، يعمل بالكامل
- **ASPDOWN_SENT** - إيقاف تشغيل سلس قيد التقدم

مراقبة نظائر M3UA عبر واجهة الويب

توفر واجهة الويب مراقبة في الوقت الحقيقي لاتصالات نظير M3UA.

الوصول إلى صفحة حالة M3UA:

1. انتقل إلى الصفحة الرئيسية لواجهة الويب
2. في قائمة التنقل انقر على "حالة M3UA"
3. يتم تحديث الصفحة تلقائيًا كل ثانية

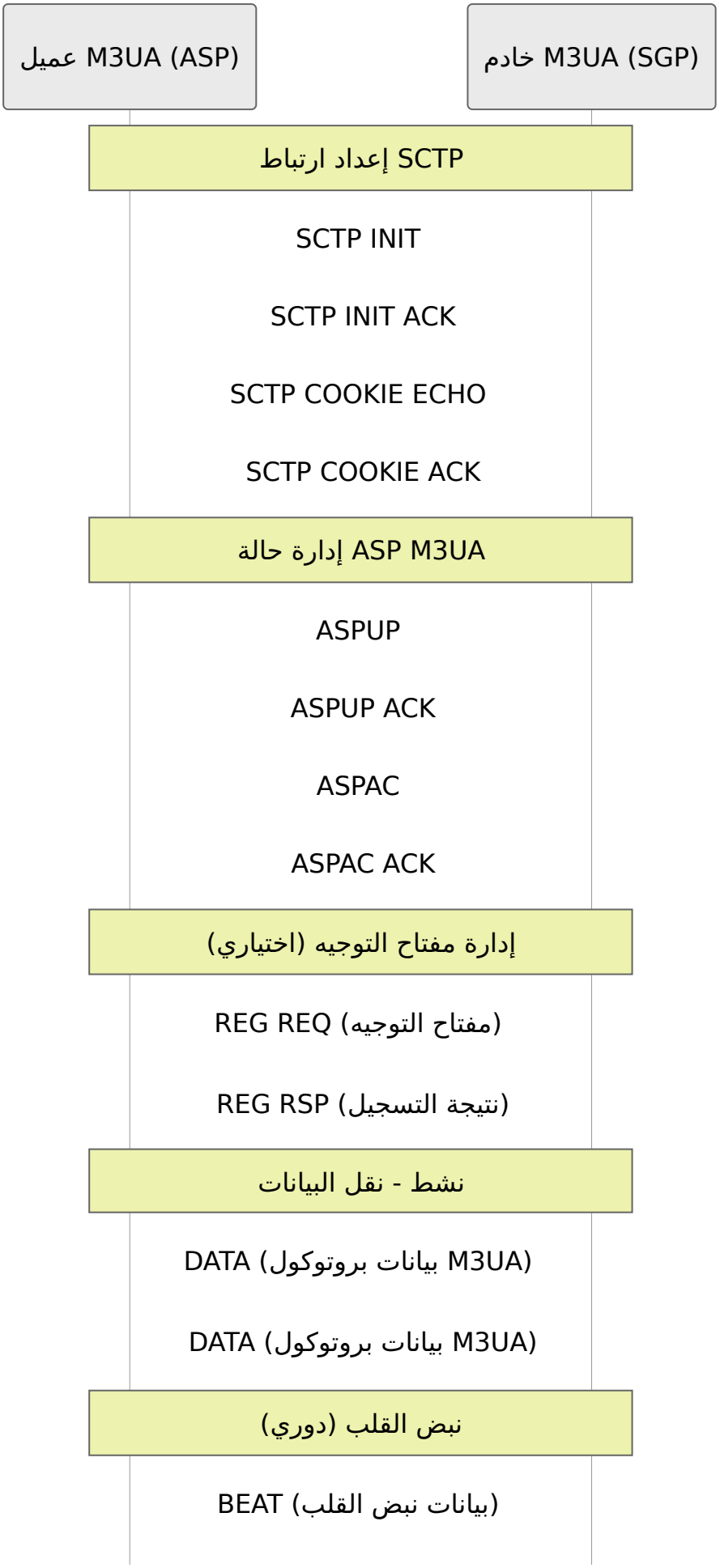
جدول حالة M3UA:

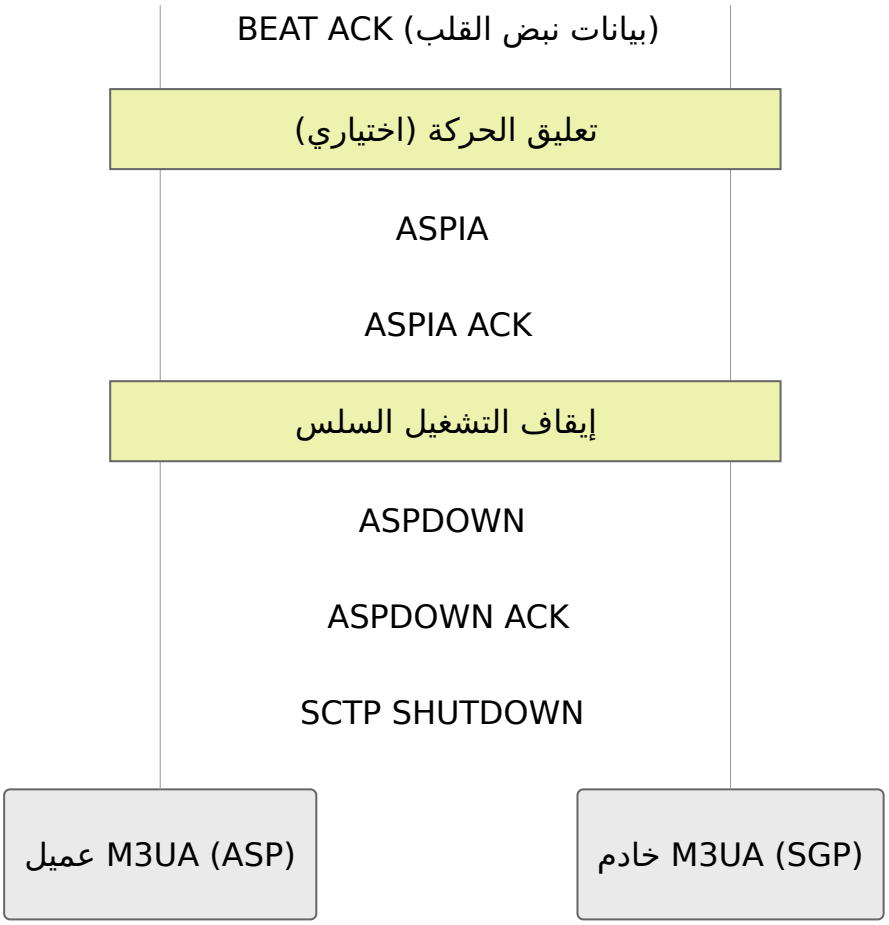
العمود	الوصف
الاسم	اسم الاتصال (مثل `testASP`)
PID	معرف العملية
الحالة	نشطة (أخضر) أو غير نشطة (أحمر)
حالة ASP	الحالة الحالية لـ M3UA (مثل، نشطة، غير نشطة)
Assoc/SCTP	حالة ارتباط SCTP
محلي	IP:Port المحلي
بعيد	IP:Port البعيد
RC	معرف سياق التوجيه

مؤشرات الحالة:

- **أخضر (نشط)** - الاتصال نشط وصحي
- **أحمر (غير نشط)** - الاتصال متوقف أو غير متاح
- **حالة ASP** - تظهر الحالة الحالية لاتصال M3UA
- **Assoc/SCTP** - تظهر حالة ارتباط SCTP

تدفق رسائل M3UA


عميل M3UA (ASP)
خادم M3UA (SGP)
إعداد ارتباط SCTP
SCTP INIT
SCTP INIT ACK
SCTP COOKIE ECHO
SCTP COOKIE ACK
إدارة حالة ASP M3UA
ASPUP
ASPUP ACK
ASPAC
ASPAC ACK
إدارة مفتاح التوجيه (اختياري)
REG REQ (مفتاح التوجيه)
REG RSP (نتيجة التسجيل)
نشط - نقل البيانات
DATA (بيانات بروتوكول M3UA)
DATA (بيانات بروتوكول M3UA)
نبض القلب (دوري)
BEAT (بيانات نبض القلب)



دعم رسائل M3UA وفقًا لـ RFC 4666

تقوم OmniSS7 بتنفيذ مجموعة كاملة من أنواع رسائل M3UA المحددة في RFC 4666:

الفئة	النوع	الرسالة	الاتجاه	السلوك
0 (الإدارة)	0	ERR	كلاهما	يسجل رمز الخطأ ومعلومات التشخيص
0 (الإدارة)	1	NTFY	كلاهما	يسجل الإشعار
1 (النقل)	1	DATA	كلاهما	يوجه الحمولة عبر GT أو PC
2 (SSNM)	1	DUNA	استلام	يسجل الوجهة غير المتاحة
2 (SSNM)	2	DAVA	استلام	يسجل الوجهة المتاحة
2 (SSNM)	3	DAUD	استلام	يستجيب حسب حالة الوجهة: DAVA للـ PCs القابلة للوصول، DUNA للـ PCs غير القابلة للوصول/المحظورة
2 (SSNM)	4	SCON	استلام	يسجل الازدحام
2 (SSNM)	5	DUPU	استلام	يسجل الجزء المستخدم غير المتاح
2 (SSNM)	6	DRST	استلام	يسجل الوجهة المقيدة
3 (ASPSM)	1	ASPUP	كلاهما	يرسل/يستقبل ACK ASPUP
3 (ASPSM)	2	ASPDN	كلاهما	يرسل ACK ASPDN، ويزيل مسارات النظير
3 (ASPSM)	3	BEAT	كلاهما	يكرر بيانات نبض القلب في BEAT ACK

الفئة	النوع	الرسالة	الاتجاه	السلوك
3 (ASPSM)	4	ASPUP ACK	استلام	ينتقل بالنظير إلى نشط
3 (ASPSM)	5	ASPDN ACK	استلام	ينتقل بالنظير إلى غير نشط
3 (ASPSM)	6	BEAT ACK	استلام	يؤكد نبض القلب
4 (ASPTM)	1	ASPAC	كلاهما	ينشط، ASPAC ACK يرسل/يستقبل المسارات
4 (ASPTM)	2	ASPIA	كلاهما	ينتقل بالنظير إلى، ASPIA ACK يرسل غير نشط
4 (ASPTM)	3	ASPAC ACK	استلام	ينتقل بالنظير إلى نشط
4 (ASPTM)	4	ASPIA ACK	استلام	ينتقل بالنظير إلى غير نشط
9 (RKM)	1	REG REQ	استلام SGP	يحلل مفتاح التوجيه، يستجيب بـ REG RSP
9 (RKM)	2	REG RSP	استلام ASP	يسجل نتيجة التسجيل
9 (RKM)	3	DEREG REQ	استلام SGP	يستجيب بـ DEREG RSP
9 (RKM)	4	DEREG RSP	استلام ASP	يسجل نتيجة إلغاء التسجيل

نبض القلب (BEAT/BEAT ACK): عندما يستقبل STP رسالة BEAT، فإ��ه يكرر بيانات نبض القلب مرة أخرى في BEAT ACK. هذا يسمح للنظائر باكتشاف فشل الروابط دون الاعتماد

فقط على مهلات مستوى SCTP.

إدارة مفتاح التوجيه (RKM): عندما يرسل ASP REG REQ، يقوم SGP بتحليل مفتاح التوجيه (معرف RK وضع الحركة، التوجيه، سياق المحلي، DPC) ويستجيب بـ REG RSP تحتوي على حالة التسجيل. هذا يمكّن من توفير المسارات الديناميكية دون الحاجة إلى تكوين ثابت.

ASP غير نشط (ASPDN): عندما يستقبل SGP ASPDN، فإنه يستجيب بـ ASPDN ACK، ويحدد النظير على أنه غير نشط، ويزيل المسارات المرتبطة من جدول التوجيه.

ASP غير نشط (ASPIA): عندما يستقبل SGP ASPIA، فإنه يستجيب بـ ASPIA ACK وينتقل بالنظير إلى غير نشط، مما يوقف الحركة دون إزالة الارتباط.

التعامل مع سياق التوجيه في M3UA

سياق التوجيه (RC) هو معرف 32 بت في M3UA يحدد سياق خادم التطبيق (AS) في كل نقطة في نقاط مختلفة في مسار الإشارة لضمان تكوين RC نهاية. من المهم فهم كيفية التعامل مع الشبكة بشكل صحيح.

ما هو سياق التوجيه؟

وفقًا لـ RFC 4666، القسم 3.3.1، فإن سياق التوجيه:

- يحدد بشكل فريد خادم التطبيق (AS) في نقطة النهاية
- يتم حمله في رسائل DATA الخاصة بـ M3UA (علامة المعامل 6)
- يسمح لبوابة الإشارة بتحديد أي AS يجب أن يستقبل رسالة
- له معنى لكل قفزة، وليس من النهاية إلى النهاية

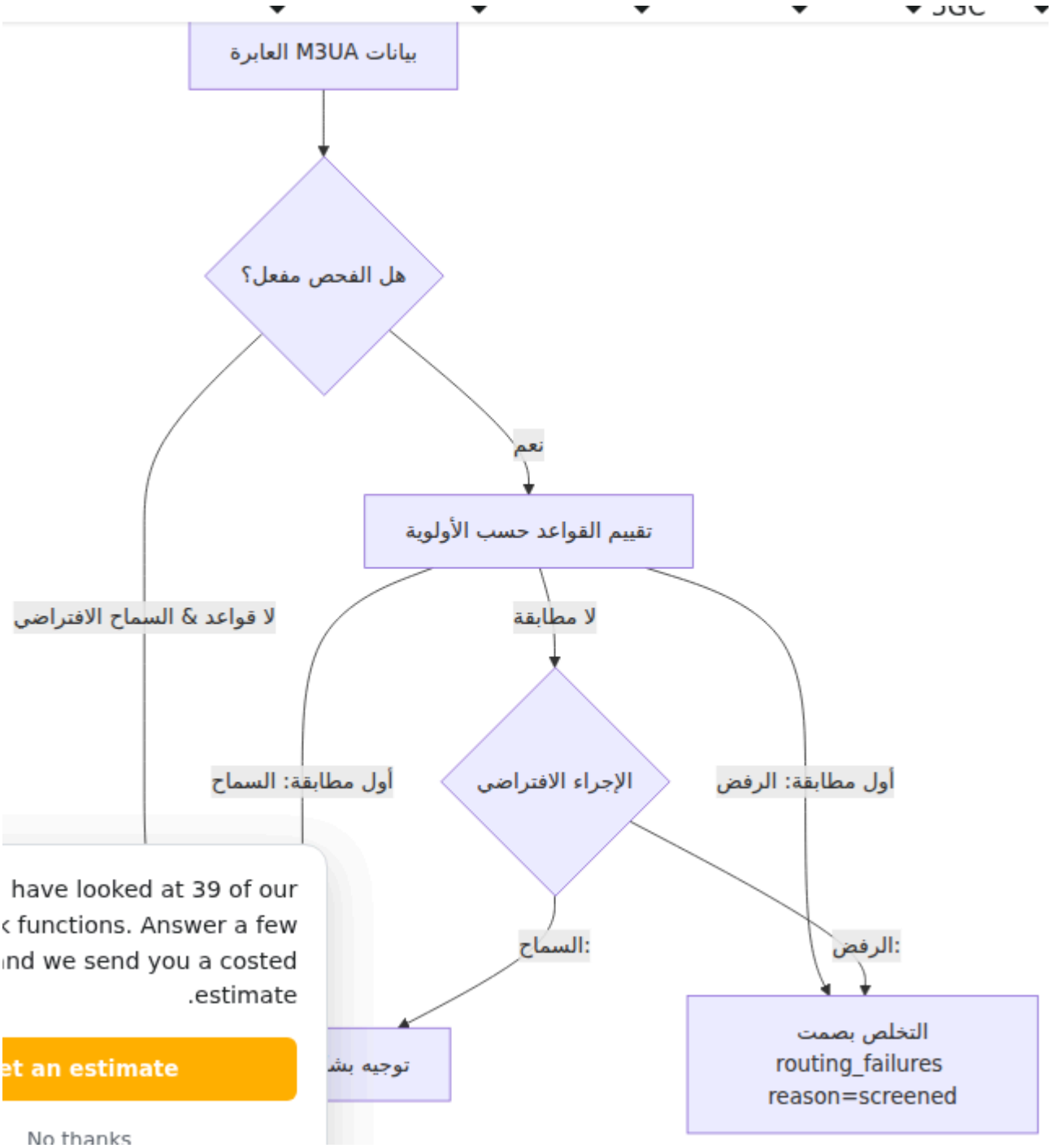
سياق التوجيه في بروتوكول M3UA



جنبًا إلى جنب مع معامل M3UA الخاصة بـ DATA يظهر معامل سياق التوجيه (علامة 6) في رسائل بيانات البروتوكول (علامة 528) التي تحتوي على الحمولة الفعلية لـ MTP3/SCCP.

كيف تتعامل OmniSS7 مع سياق التوجيه

تتعامل OmniSS7 مع سياق التوجيه بشكل مختلف حسب المكون واتجاه الرسالة:



عميل CAP وعميل MAP

يتم التعامل مع سياق التوجيه على النحو التالي، بالنسبة لعملاء طبقة التطبيق (CAPو MAP):

السيناريو	معالجة RC	التكوين
استجابة لطلب وارد	عكس RC الوارد في الاستجابة	تلقائي - لا حاجة لتكوين
طلب مبادر من العميل (SRI، PRN، IDP، إلخ)	استخدام RC المكون من تكوين التطبيق	تعيين `routing_context` في `map_client_m3ua` أو `cap_client_m3ua`
لا يوجد RC في الطلب الوارد	حذف RC من الاستجابة	يحافظ على التوافق العكسي
لا يوجد مكون RC لطلبات العميل	حذف RC من الرسالة الصادرة	يحافظ على التوافق العكسي

مثال على التكوين:

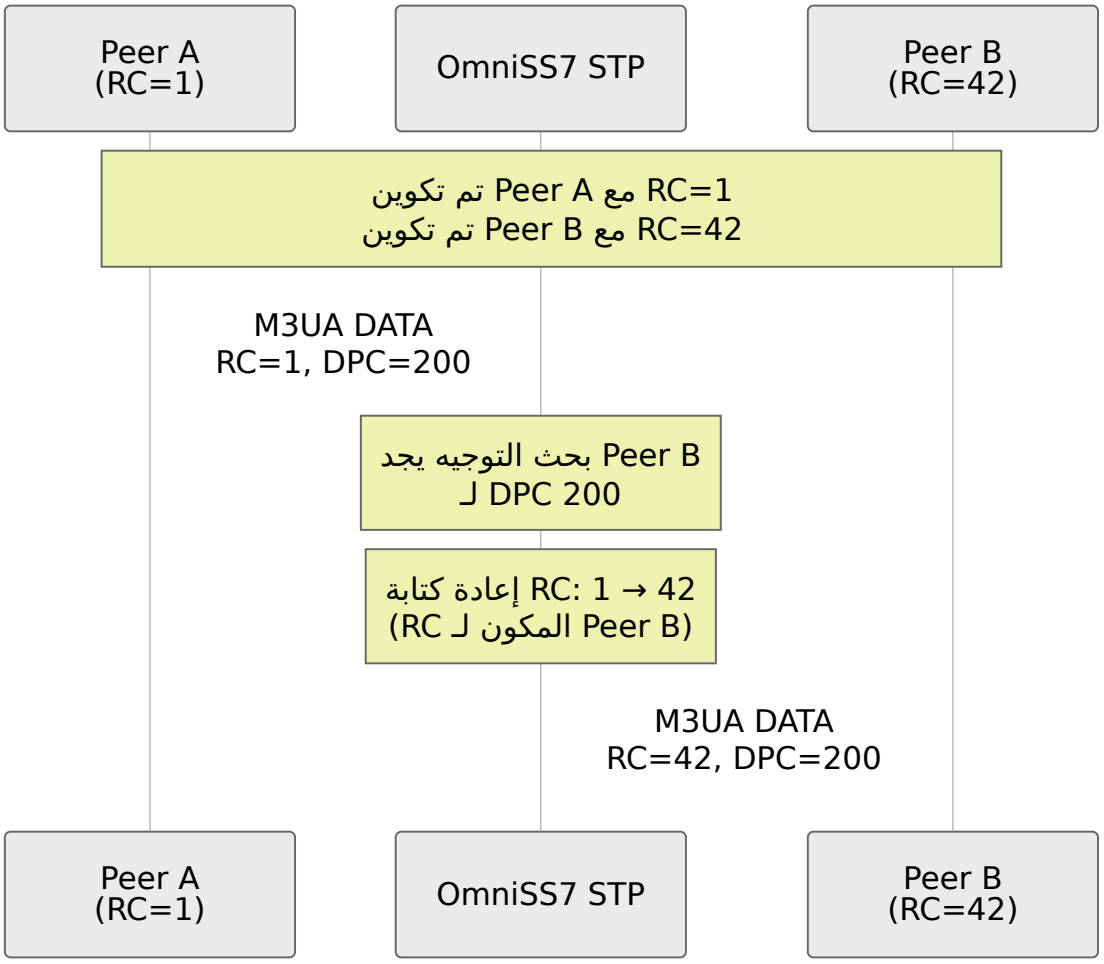
```
config :omniss7,
  # تكوين RC لعميل MAP
  map_client_m3ua: %{
    mode: "ASP",
    local_ip: {10, 0, 0, 1},
    local_port: 2905,
    remote_ip: {10, 0, 0, 10},
    remote_port: 2905,
    routing_context: 1    # مستخدم لطلبات MAP من العميل (SRI، PRN، إلخ)
  },

  # تكوين RC لعميل CAP
  cap_client_m3ua: %{
    mode: "ASP",
    local_ip: {10, 0, 0, 1},
    local_port: 2906,
    remote_ip: {10, 0, 0, 20},
    remote_port: 2906,
    routing_context: 2    # مستخدم لطلبات CAP من العميل (InitialDP، إلخ)
  }
```

إعادة كتابة سياق التوجيه STP

عندما يقوم STP المكون RC بإعادة توجيه الرسائل بين الأقران، فإنه **يعيد كتابة** سياق التوجيه إلى للقرين الوجهة. هذا هو السلوك الصحيح وفقًا لـ RFC 4666 لأن:

1. في كل نقطة نهاية AS يحدد سياق RC
2. قد يرتبط أقران مختلفون بسياقات AS مختلفة
3. عند التوجيه بين RC يقوم STP بترجمة قيم ASs المختلفة

تدفق إعادة كتابة RC في STP:



التكوين:

يتم تكوين سياق التوجيه لكل قرين في تعريف القرين:

```
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "Network_A_STP",
      role: :server,
      remote_ip: {10, 0, 0, 100},
      remote_port: 2905,
      routing_context: 1,        # RC المتوقع من/إلى هذا القرين
      point_code: 100,
      network_indicator: :international
    },
    %{
      peer_id: 2,
      name: "Network_B_HLR",
      role: :client,
      local_ip: {10, 0, 0, 1},
      local_port: 0,
      remote_ip: {10, 0, 0, 200},
      remote_port: 2905,
      routing_context: 42,       # RC المتوقع من هذا القرين
      point_code: 200,
      network_indicator: :international
    }
  ]
```

عندما تصل رسالة من Peer 1 (RC=1) موجهة إلى Peer 2، يقوم STP:

1. بتوجيه الرسالة بناءً على DPC (200 → Peer 2)
2. إعادة كتابة RC 42 إلى 1 (RC المكون لـ Peer 2)
3. إرسال الرسالة إلى Peer 2

ملخص معالجة RC

المكون	الطلب الوارد	الاستجابة الصادرة	المبادر من العميل
عميل MAP	استخراج RC	عكس الوارد RC	استخدام المكون RC
عميل CAP	استخراج RC	عكس الوارد RC	استخدام المكون RC
STP	استخراج RC	N/A	إعادة كتابة إلى القرين RC الوجهة

عندما يتم حذف RC

إذا لم يكن لدى قرين أي (أو تم تعيينه إلى `nil`) مكون `routing_context`:

- **استجابات عملاء MAP/CAP**: يتم حذف معلمة RC من الاستجابة
- **طلبات عملاء MAP/CAP**: يتم حذف معلمة RC من الطلب
- **إعادة توجيه STP**: لا يتم إعادة كتابة RC (يتم الحفاظ على RC الأصلي أو حذفه)

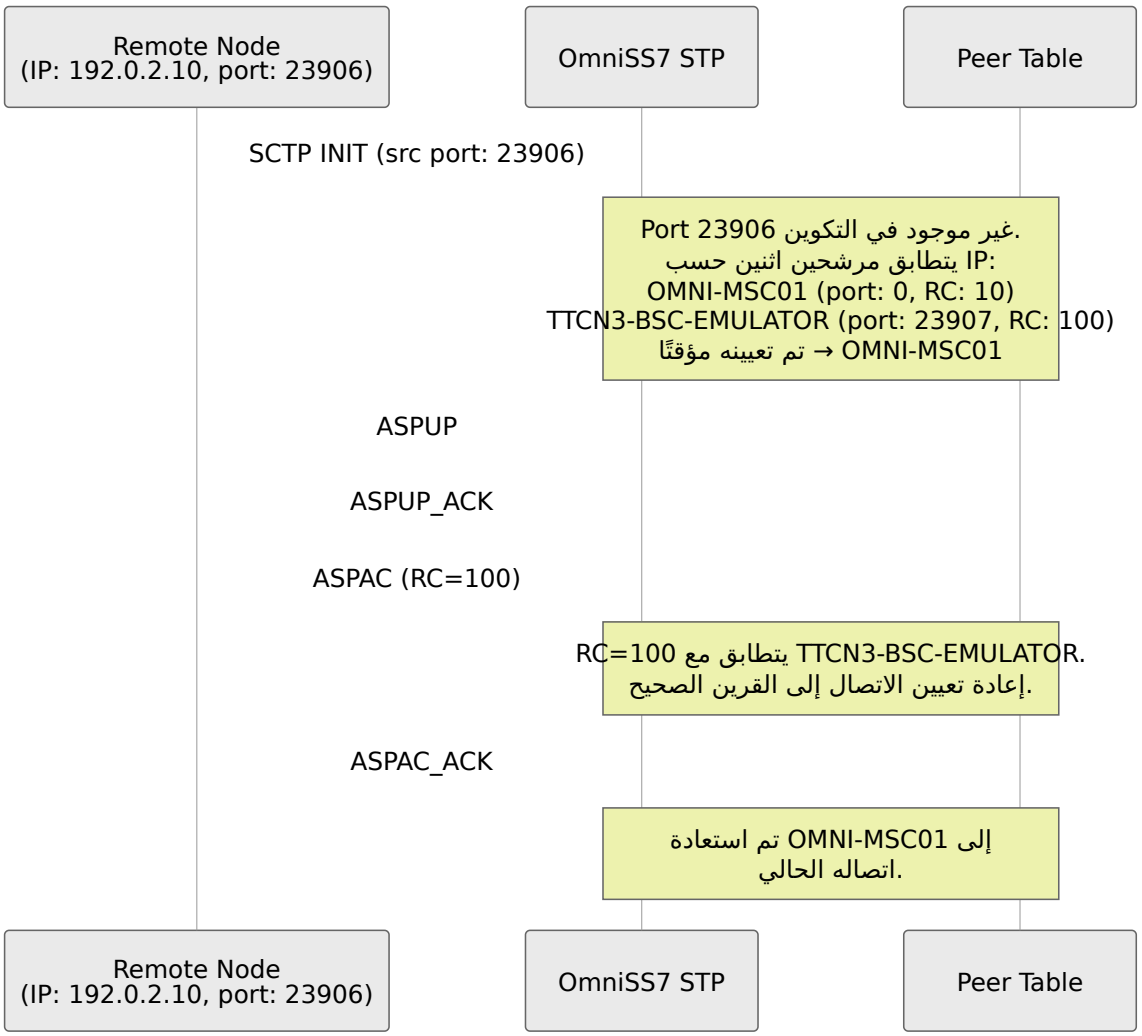
هذا يحافظ على التوافق العكسي مع الأقران التي لا تستخدم أو تتوقع سياق التوجيه.

تفريق الأقران متعدد IP بنفس العنوان

يمكن أن تشترك اثنان من الأقران المكونة في نفس عنوان IP على سبيل المثال: MSC البعيد، الوارد، لا يمكن لـ SCTP الإنتاج ومحاكي اختبار يعمل على نفس المضيف. عندما يصل اتصال تحديد هوية القرين الصحيحة من عنوان المصدر فقط. سيطابق قرين مكون بـ OmniSS7 (wildcard) `remote_port: 0`، مما قد يتسبب في تعيين اتصال جديد من IP أي اتصال من ذلك القرين الآخر بشكل مؤقت إلى الإدخال الخطأ.

تحل OmniSS7 هذه المشكلة تلقائيًا باستخدام **سياق توجيه** **M3UA ASPAC**. يتم قبول الاتصال وتعيينه مؤقتًا إلى القرين الذي يتطابق بشكل أفضل (يفضل المطابقة المحددة على wildcard). تحمل سياق توجيه، يقوم النظام بمقارنة `ASPAC` عندما يرسل الطرف البعيد رسالة مع قرين مختلف، يتم إعادة RC مكون للأقران. إذا تطابق `routing_context` مع كل RC هذا تعيين الاتصال بهدوء إلى الصحيح.

كيف يعمل



مخرجات السجل

بتسجيل جميع المرشحين حتى يتمكن OmniSS7 عندما يتطابق عدة أقران مع اتصال وارد، يقوم المشغلون من التحقق من أن التفريق يعمل بشكل صحيح:

```
Inbound connection from 192.0.2.10:23906 matches peers: ["OMNI-
MSC01", "TTCN3-BSC-EMULATOR"] - tentatively using "OMNI-MSC01"
until ASPAC RC confirms identity
```

بعد استلام سياق توجيه ASPAC:

```
ASPAC RC=100 matches peer 9 (TTCN3-BSC-EMULATOR), but connection
was assigned to peer 8 (OMNI-MSC01) - remapping
```

متطلبات التكوين

بشكل صحيح، **يجب أن يكون لكل قرين** IP لكي يعمل تفريق الأقران بنفس RC (أو نفس IP **فريد مكون**. إذا كان لدى اثنين من الأقران على نفس `routing_context` كلاهما لديه `nil`)، فإن إعادة التعيين لا يمكن أن تحدد أي قرين هو الصحيح.

```
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 8,
      name: "OMNI-MSC01",
      role: :server,
      remote_ip: {10, 5, 198, 200},
      remote_port: 0,              # Wildcard - accepts any source
port
      routing_context: 10,         # Must be unique across peers
sharing this IP
      point_code: 500,
      network_indicator: :national
    },
    %{
      peer_id: 9,
      name: "TTCN3-BSC-EMULATOR",
      role: :server,
      remote_ip: {10, 5, 198, 200},
      remote_port: 23907,          # Expected source port (may
differ for ephemeral ports)
      routing_context: 100,        # Different RC - used to
identify this peer after ASPAC
      point_code: 193,
      network_indicator: :national
    }
  ]
```

> **ملاحظة حول المنافذ المصدر المؤقتة**: المطابقة بين المنافذ دقيقة. إذا تم تكوين قرين بـ `remote_port: 23907` ولكنه يتصل من المنفذ 23906، يتم استبعاد ذلك القرين من المطابقة الأولية ويفوز القرين wildcard هو ASPAC بدلاً من ذلك. هذا متوقع. سياق توجيه `remote_port` المكون على قرين فقط لهيوريستيك إشارة الهوية الموثوقة. يتم استخدام `routing_context` مميز وسيتم تعيين الاتصال الأولي، وليس لهوية نهائية. أعطِ كل قرين حل إعادة التعيين بشكل صحيح بغض النظر عن المنفذ المصدر الذي يستخدمه البعيد.

استكشاف اتصالات M3UA

المشكلة: لا يمكن إنشاء الاتصال

الأعراض:

- الحالة تظهر DOWN
- لا يوجد ارتباط SCTP

التحققات:

1. تحقق من الاتصال الشبكي: `ping remote_ip`
2. تحقق من أن جدار الحماية يسمح بـ SCTP (البروتوكول 132)
3. تحقق من أن STP/SGP البعيد يستمع على المنفذ الصحيح
4. تحقق من `remote_ip` و `remote_port` في التكوين
5. راجع سجلات التطبيق بحثًا عن أخطاء SCTP

المشكلة: الاتصال تم إنشاؤه ولكن ASP غير نشط

الأعراض:

- يوجد ارتباط SCTP
- حالة ASP عالقة في INACTIVE أو ASPUP_SENT

التحققات:

1. تحقق من أن سياق التوجيه يتطابق مع التكوين البعيد
2. تحقق من أن STP البعيد يقبل رمز النقطة الخاص بك
3. راجع السجلات بحثًا عن رفضات ASPUP/ASPAC
4. تحقق من عدم وجود متطلبات مصادقة/أمان

المشكلة: البيانات لا تتدفق

الأعراض:

- حالة ASP تظهر ACTIVE
- لا يتم توجيه أي رسائل

التحققات:

1. تحقق من سياق التوجيه في الرسائل
2. تحقق من عناوين SCCP (تنسيق GT، قيم SSN)
3. تحقق من أن جداول التوجيه تم تكوينها بشكل صحيح
4. راجع صفحة `/events` بحثًا عن أخطاء SCCP
5. تحقق من توجيه رمز النقطة على مستوى STP

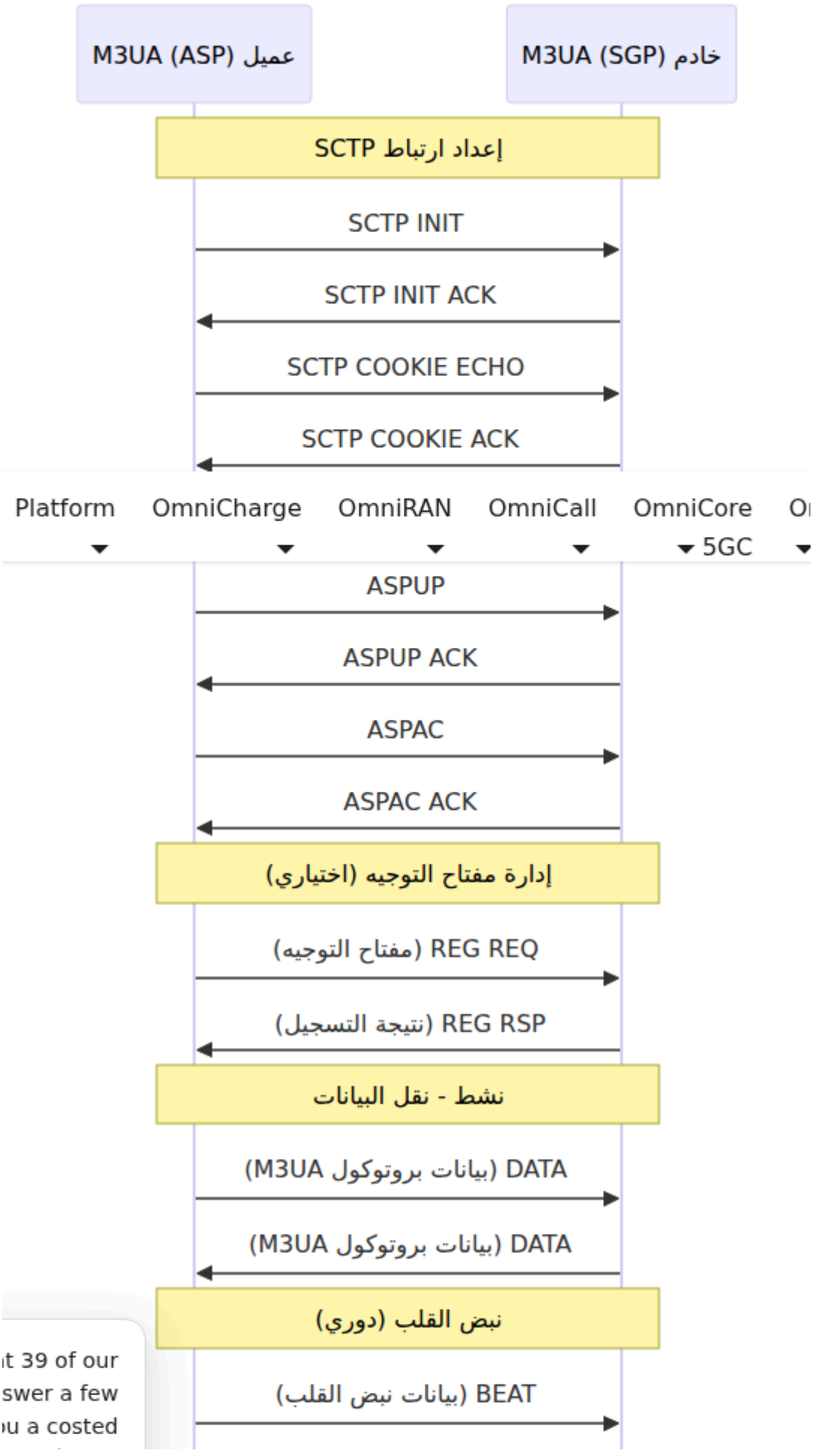
مراقبة حالة الأقران M2PA

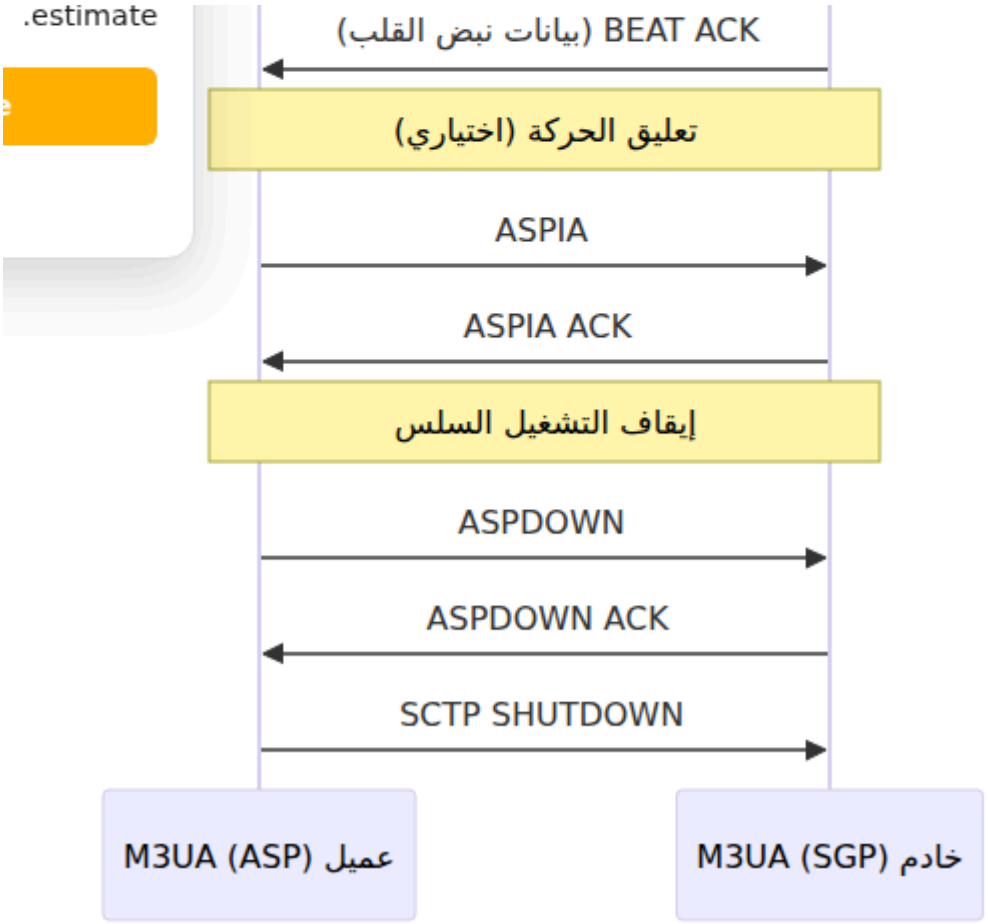
فهم M2PA

M2PA (طبقة التكيف بين الأقران لمستخدم MTP2) هو بروتوكول محدد في RFC 4165 يوفر نقل رسائل MTP3 من نقطة إلى نقطة عبر SCTP. على عكس M3UA الذي يستخدم بنية ASP/SGP، يوفر M2PA روابط بين الأقران مشابهة لروابط TDM SS7 التقليدية.

حالات رابط M2PA

تتقدم روابط M2PA عبر عدة حالات أثناء الإنشاء:


عميل M3UA (ASP)
خادم M3UA (SGP)
إعداد ارتباط SCTP
SCTP INIT
SCTP INIT ACK
SCTP COOKIE ECHO
SCTP COOKIE ACK
Platform
OmniCharge
OmniRAN
OmniCall
OmniCore
5GC
ASPUP
ASPUP ACK
ASPAC
ASPAC ACK
إدارة مفتاح التوجيه (اختياري)
REG REQ (مفتاح التوجيه)
REG RSP (نتيجة التسجيل)
نشط - نقل البيانات
DATA (بيانات بروتوكول M3UA)
DATA (بيانات بروتوكول M3UA)
نبض القلب (دوري)
BEAT (بيانات نبض القلب)



وصف الحالات:

- **DOWN** - الرابط غير نشط، لا يوجد اتصال SCTP
- **CONNECTING** - تقدم ارتباط SCTP
- **ALIGNMENT** - تم تبادل رسائل حالة الرابط (~1 ثانية)
- **PROVING** - فترة إثبات الرابط، اختبار سلامة الرابط (~2 ثواني)
- **READY** - الرابط نشط، جاهز لنقل بيانات مستخدم MTP3
- **ALIGNMENT (إعادة الدخول)** - يتطلب تغيير حالة الرابط إعادة المحاذاة

تقدم حالة الرابط:

1. **اتصال SCTP**: إنشاء ارتباط SCTP (DOWN → CONNECTING)
2. **المحاذاة**: تبادل رسائل حالة الرابط للتزامن (CONNECTING → ALIGNMENT)
3. **الإثبات**: اختبار موثوقية الرابط وتزامن أرقام التسلسل (ALIGNMENT → PROVING)
4. **جاهز**: يصبح الرابط نشطًا لنقل البيانات (PROVING → READY)

تدفق رسائل M2PA


M2PA Peer 1 (PC: 100)
M2PA Peer 2 (PC: 200)
إعداد ارتباط SCTP
SCTP INIT
SCTP INIT ACK
SCTP COOKIE ECHO
SCTP COOKIE ACK
محاذاة رابط M2PA
حالة الرابط (ALIGNMENT)
حالة الرابط (ALIGNMENT)
إثبات رابط M2PA (~2s)
حالة الرابط (PROVING)
حالة الرابط (PROVING)
الرابط جاهز
حالة الرابط (READY)
حالة الرابط (READY)
جاهز - نقل بيانات MTP3
بيانات المستخدم (رسالة MTP3، BSN/FSN)
بيانات المستخدم (رسالة MTP3، BSN/FSN)
بيانات المستخدم (رسالة MTP3، BSN/FSN)
بيانات المستخدم (رسالة MTP3، BSN/FSN)



مراقبة الأقران M2PA عبر واجهة الويب

توفر واجهة الويب مراقبة في الوقت الحقيقي لاتصالات الأقران M2PA.

الوصول إلى صفحة إدارة التوجيه:

1. انتقل إلى الصفحة الرئيسية لواجهة الويب
2. انقر على "إدارة التوجيه" في قائمة التنقل
3. عرض جدول "أقران M3UA/M2PA"

جدول الأقران M2PA:

العمود	الوصف
معرف القرين	معرف فريد للقرين
الاسم	اسم القرين (مثل `M2PA_Link_STP_A`)
البروتوكول	يظهر "M2PA" باللون الأخضر
رمز النقطة	رمز النقطة المحلي
رمز النقطة المجاورة	رمز النقطة للقرين المجاور
المحلي	IP:Port المحلي (عادةً المنفذ 3565)
البعيد	IP:Port البعيد
الحالة	حالة الرابط (مثل READY، ALIGNMENT، DOWN)

مؤشرات الحالة:

- **READY (أخضر)** - الرابط نشط وينقل الحركة
- **ALIGNMENT (أصفر)** - الرابط في مرحلة المحاذاة، غير جاهز بعد
- **PROVING (أصفر)** - الرابط في حالة الإثبات
- **DOWN (أحمر)** - الرابط معطل أو غير متاح

استكشاف اتصالات M2PA

المشكلة: الرابط عالق في ALIGNMENT

الأعراض:

- حالة الرابط تظهر ALIGNMENT لفترة طويلة
- لا تقدم إلى READY أو PROVING

التحققات:

1. تحقق من أن الجانبين مكونان برموز النقطة الصحيحة

2. تحقق من أن جدار الحماية يسمح بـ SCTP 132 البروتوكول
3. تحقق من أن `point_code` و `adjacent_point_code` تم تعيينهما بشكل صحيح
4. راجع سجلات التطبيق بحثًا عن أخطاء رسائل حالة الرابط
5. تأكد من أن القرين البعيد أيضًا في حالة ALIGNMENT

المشكلة: الرابط عالق في PROVING

الأعراض:

- يصل الرابط إلى PROVING ولكنه لا ينتقل إلى READY
- تتجاوز فترة الإثبات 2-3 ثواني

التحققات:

1. تحقق من استقرار الشبكة (لا فقدان للحزم)
2. تحقق من أخطاء ارتباط SCTP
3. راجع السجلات بحثًا عن عدم تطابق أرقام التسلسل
4. تأكد من أن القرين البعيد أيضًا في حالة PROVING
5. تحقق من أن تعدد استخدام SCTP لا يسبب مشاكل في التوجيه

المشكلة: الرابط يتقلب (DOWN ↔ READY)

الأعراض:

- يتكرر الرابط بين DOWN و READY
- إعادة محاذاة متكررة

التحققات:

1. تحقق من استقرار الاتصال الشبكي
2. تحقق من إعدادات نبض SCTP
3. راجع إعدادات انتهاء جلسة جدار الحماية
4. تحقق من مشاكل MTU/التجزئة
5. تحقق من عدم وجود عناوين IP مكررة

المشكلة: البيانات لا تتدفق

الأعراض:

- حالة الرابط تظهر READY
- لا يتم نقل أي رسائل MTP3

التحققات:

1. تحقق من أن جداول التوجيه تتضمن مسارات إلى هذا القرين
2. تحقق من تكوين توجيه رمز النقطة MTP3
3. راجع قيم DPC المتوقعة تطابق المسارات في الرسائل
4. تحقق من صفحة `/events` بحثًا عن أخطاء التوجيه
5. تحقق من أن أرقام التسلسل (BSN/FSN) تتزايد

الوثائق ذات الصلة

- ← العودة إلى الوثائق الرئيسية
- واجهة الويب - دليل الميزات الشائعة، API، المراقبة
- إرسال طلبات MAP - دليل عميل MAP
- تسليم الرسائل القصيرة - دليل مركز الرسائل القصيرة
- مواصفات البروتوكول - المرجع الفني

OmniSS7 بواسطة Omnitouch Network Services

دليل بوابة USSD

← العودة إلى الوثائق الرئيسية

يغطي هذا الدليل **بوابة USSD** من OmniSS7، التي تربط حوارات USSD SS7/MAP مع ردود HTTP/JSON، مما يمكّن المطورين من بناء تطبيقات USSD من خلال نقطة نهاية HTTP بسيطة.

جدول المحتويات

1. نظرة عامة
2. الهيكلية
3. تمكين بوابة USSD
4. التكوين
5. بروتوكول رد HTTP
6. USSD المنشأ من الشبكة (Push API)
7. دورة حياة الجلسة
8. معالجة الأخطاء
9. المقاييس والمراقبة
10. خادم ردود المثال
11. استكشاف الأخطاء

ن��رة عامة

تتعامل بوابة USSD مع اتجاهين من حركة مرور USSD:

- **المنشأ من الهاتف المحمول (Inbound)**: يقوم المشترك بالاتصال برمز قصير (مثل `*100#`). تتلقى البوابة طلب MAP `processUnstructuredSS-Request` (opcode 59). ثم تعيد إرسال HTTP تقوم بإعادة توجيه الطلب إلى رد الخاص بك. ردك عبر SS7.
- **المنشأ من الشبكة (Outbound)**: يقوم تطبيقك بدفع رسالة USSD إلى مشترك عبر REST API. ترسل البوابة طلب MAP `unstructuredSS-Request` (opcode

60) عبر SS7. ثم تقوم بتوجيه رد المشترك إلى ردك.

يدعم كلا الاتجاهين **حوارات متعددة الأدوار**: قوائم تفاعلية حيث يرد المشترك ويتلقى مطالبات متابعة.

الخصائص الرئيسية

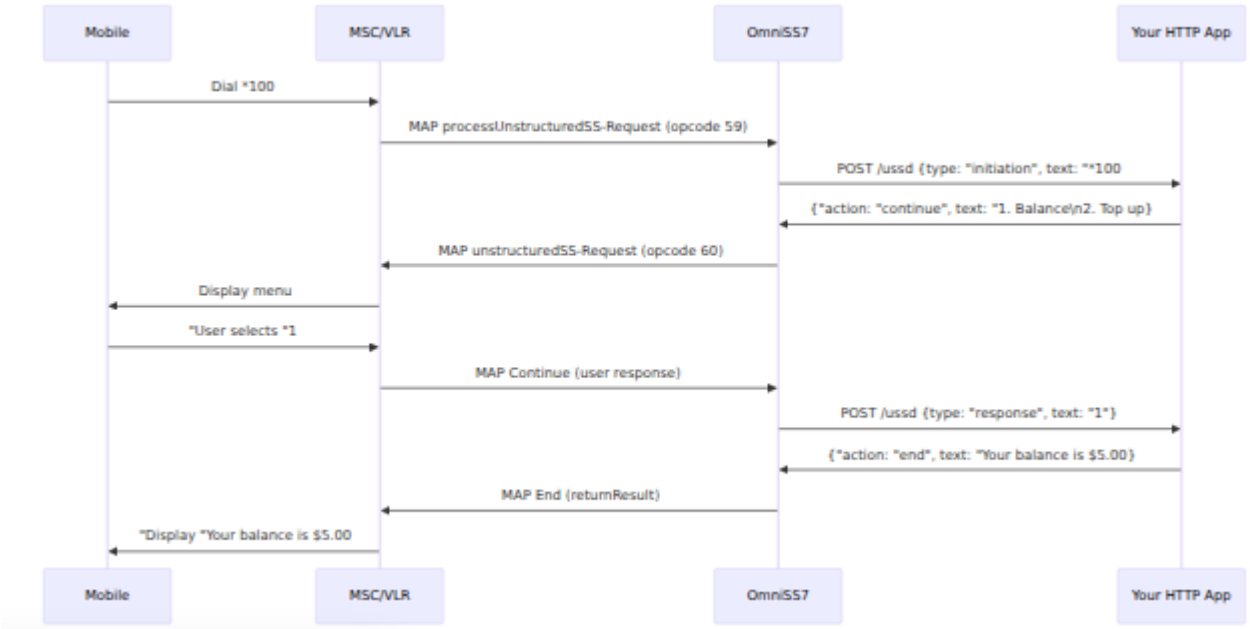
الخاصية	القيمة
النقل	HTTP POST متزامن لكل دور
التشفير	أبجدية GSM 7-bit الافتراضية (DCS 0x0F) 3 لـ GPP TS وفقًا 23.038
أقصى طول نص	182 حرف (قابل للتكوين)
تتبع الجلسة	تم إنشاؤه بواسطة البوابة لكل حوار UUID
المصادقة	ل�� شيء (SS7 يثق بشبكة)
التوجيه	تطابق بادئة الرمز القصير مع عناوين ردود

مراجع 3GPP

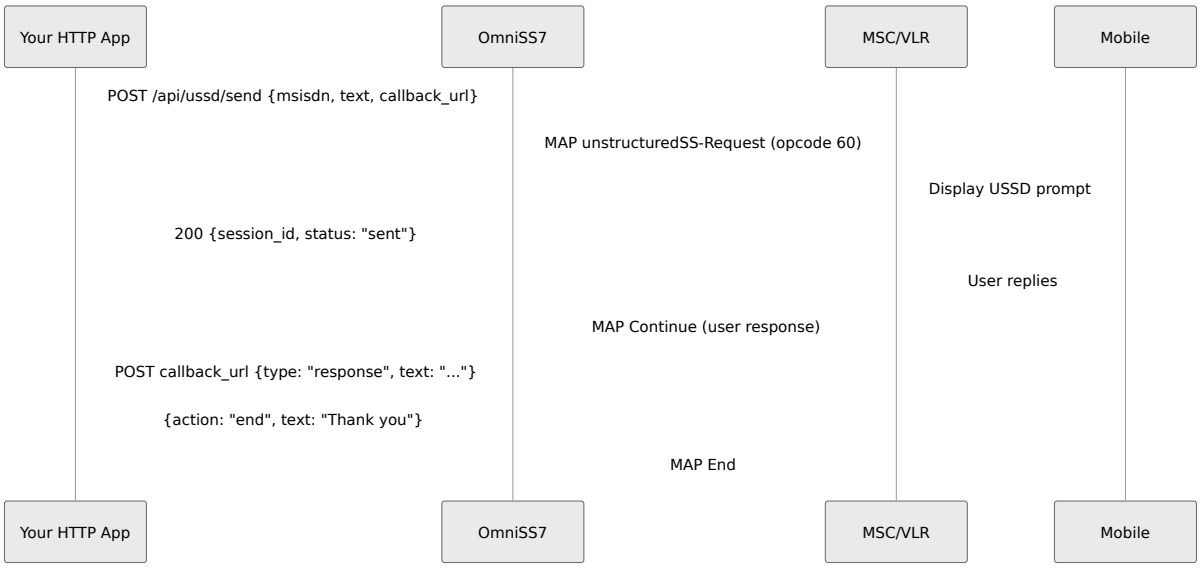
المواصفة	الصلة
3GPP TS 23.090	المرحلة 2: الهيكلية والإجراءات USSD
3GPP TS 24.090	المرحلة 3: تفاصيل البروتوكول USSD
3GPP TS 29.002	بروتوكول MAP: USSD-Arg، USSD-Res، opcodes 59/60/61
3GPP TS 23.038	أبجدية GSM 7-bit الافتراضية ونظام ترميز البيانات

الهيكلية

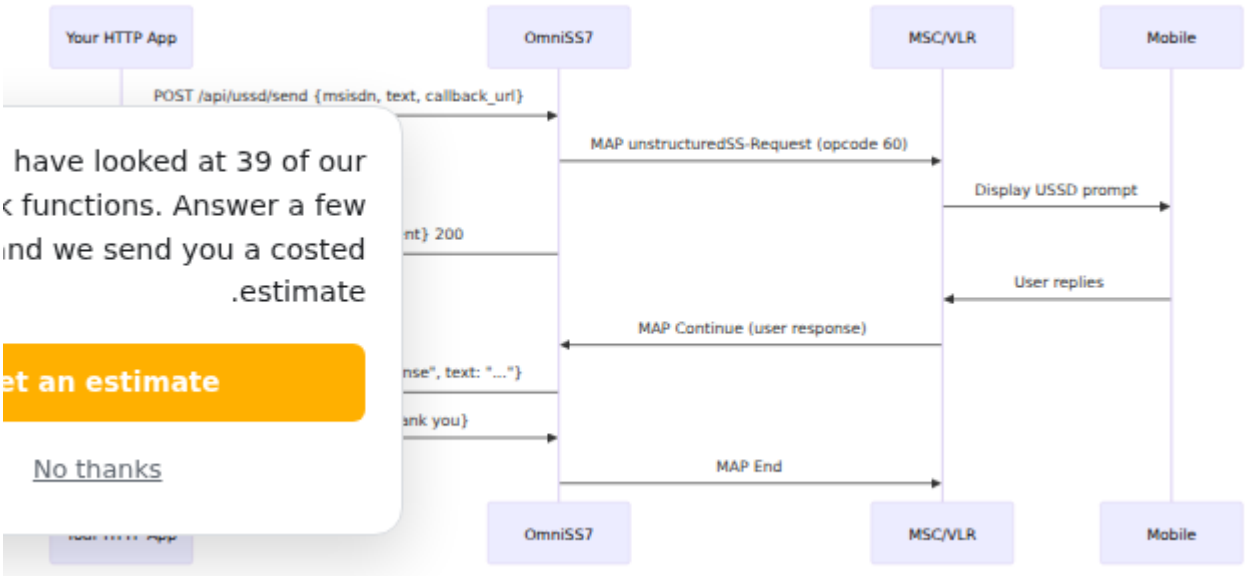
تدفق المنشأ من الهاتف المحمول (Inbound)



تدفق المنشأ من الشبكة (Outbound Push)



نظرة عامة على المكونات



تمكين بوابة USSD

تتطلب بوابة USSD **تمكين وضع عميل MAP**، بالإضافة إلى علم الميزات الخاص بها.

```
config :omniss7,
  map_client_enabled: true,
  ussd_gateway_enabled: true
```

تتطلب البوابة أيضًا اتصال M3UA يعمل (راجع دليل عميل MAP لإعداد M3UA).

التكوين

معلمات بوابة USSD

```
config :omniss7,
  ussd_gateway_enabled: true,
  ussd_gateway: %{
    # توجيه الرمز القصير - أطول تطابق بادئة
    routes: [
      %{pattern: "*100", url: "http://balance-app:9000/ussd"},
      %{pattern: "*200", url: "http://topup-app:9000/ussd"},
      %{pattern: "*",    url: "http://default-app:9000/ussd"}
    ],

    # مهلات الجلسة
    session_timeout_ms: 180_000,    # إجمالي مدة الجلسة (3 دقائق)
    turn_timeout_ms: 30_000,        # الحد الأقصى للانتظار لرد
المشترك لكل دور (30 ثانية)

    # إعدادات رد HTTP
    http_timeout_ms: 5_000,         # مهلة لـ HTTP POST إلى تطبيقك
(5 ثواني)

    # حدود النص
    max_text_length: 182            # الحد الأقصى لـ GSM 7-bit (يتم
اقتطاعه مع تحذير إذا تم تجاوزه)
  }
```

مرجع المعلمات

المعلمة	النوع	مطلوب	الافتراضي	صف
`ussd_gateway_enabled`	Boolean	نعم	`false`	مفتاح ئيسي لميزة بوابة USSD
`ussd_gateway.routes`	قائمة من الخرائط	نعم	`[]`	قواعد توجيه بادئة الرمز قصير. ي كل إدخال على `patt` لسلة) (بادئة `url` نوان) ل .(رد طابق ة يفوز
`ussd_gateway.session_timeout_ms`	عدد صحيح	لا	`180_000`	الحد أقصى لمدة جلسة جمالية لمللي ة. يتم إنهاء جلسة طأ إذا

المعلمة	النوع	مطلوب	الافتراضي	صف
				تم اوزها
`ussd_gateway.turn_timeout_ms`	عدد صحيح	لا	`30_000`	الحد أقصى لوقت لانتظار لرد شترك حوار متعدد لأدوار، لمللي ثانية.
`ussd_gateway.http_timeout_ms`	عدد صحيح	لا	`5_000`	طلب HTTP رد إلى بيقك، لمللي ثانية. ي كل وقت اتصال ووقت ستجابة
`ussd_gateway.max_text_length`	عدد صحيح	لا	`182`	الحد أقصى لعدد أحرف في سلسلة نص USSD يتم

صف	الافتراضي	مطلوب	النوع	المعلمة
قتطاع صوص التي تجاوز ويتم سجيل تحذير				

معلمات التوجيه

هو خريطة `routes` كل إدخال في قائمة:

المعلمة	النوع	مطلوب	الوصف
`pattern`	سلسلة	نعم	بادئة الرمز القصير للتطابق. استخدم `"*"` كخيار احتياطي شامل. تأخذ البادئات الأطول الأولوية.
`url`	سلسلة	نعم	نقطة النهاية لاستقبال ردود URL عنوان POST للرموز القصيرة المتطابقة.

مطابقة التوجيه

`#`تتم مطابقة التوجيهات بواسطة **أطول بادئة أ��لاً**. لسلسلة الاتصال `100*`:

1. تتطابق (الطول 4): تم اختيارها `"*100"`
2. تتطابق (الطول 3): تم تخطيها، أقصر `"*10"`
3. تتطابق (الطول 1): احتياطي `"*"`

إذا لم تتطابق أي توجيه، ترجع البوابة خطأ MAP وتقوم بتسجيل تحذير إلى الهاتف المحمول.

بروتوكول رد HTTP

يتلقى تطبيقك طلبات HTTP POST من البوابة ويرد بتعليمات JSON.

الطلب من البوابة إلى تطبيقك

Content-Type: `application/json`

الدور الأول (بدء الجلسة):

```
{
  "session_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "msisdn": "+254712345678",
  "type": "initiation",
  "text": "*100#",
  "turn": 1
}
```

الأدوار اللاحقة (رد المشترك):

```
{
  "session_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "msisdn": "+254712345678",
  "type": "response",
  "text": "1",
  "turn": 2
}
```

حقول الطلب

الحقل	النوع	الوصف
`session_id`	سلسلة	UUID تم إنشاؤه بواسطة البوابة. فريد لكل حوار USSD. استخدم هذا لربط الأدوا��.
`msisdn`	سلسلة	MSISDN المشترك (إذا كان متاحًا من رسالة MAP). قد تكون فارغة لبعض الشبكات.
`type`	سلسلة	`"initiation"` للدور الأول، `"response"` لردود المشترك اللاحقة.
`text`	سلسلة	سلسلة الاتصال (مثل `*100#` عند البدء، أو إدخال المشترك (مثل `1`) عند الرد).
`turn`	عدد صحيح	عداد الدور يبدأ من 1. يزيد مع كل تفاعل للمشترك.

الرد من تطبيقك

يجب على تطبيقك الرد بـ JSON يحتوي على `action` و `text`:

استمرار (عرض القائمة، الانتظار لرد المشترك):

```
{
  "action": "continue",
  "text": "1. Balance\n2. Top up\n3. Transfer"
}
```

إنهاء (عرض الرسالة النهائية، إغلاق الجلسة):

```
{
  "action": "end",
  "text": "Your balance is $5.00"
}
```

حقول الرد

الحقل	النوع	مطلوب	الوصف
`action`	سلسلة	نعم	`"continue"` للحفاظ على الجلسة مفتوحة والانتظار لرد المشترك، أو `"end"` لعرض رسالة نهائية وإغلاق الجلس��.
`text`	سلسلة	نعم	النص الذي سيتم عرضه على جهاز المشترك. الحد الأقصى للطول يتم تحديده بواسطة `max_text_length` (الافتراضي 182). استخدم `\n` لفواصل الأسطر.

USSD المنشأ من الشبكة (Push API)

ادفع رسالة USSD إلى مشترك من تطبيقك.

نقطة النهاية

`POST /api/ussd/send`

الطلب

```
{
  "msisdn": "+254712345678",
  "text": "لديك فاتورة معلقة. رد 1 للدفع.",
  "callback_url": "http://billing-app:9000/ussd"
}
```

حقول الطلب

الحقل	النوع	مطلوب	الوصف
`msisdn`	سلسلة	نعم	MSISDN المشترك الوجهة بصيغة دولية.
`text`	سلسلة	نعم	النص الأولي لـ USSD الذي سيتم عرضه. مشفر كـ GSM 7-bit.
`callback_url`	سلسلة	نعم	عنوان URL لاستقبال رد المشترك عبر بروتوكول الرد القياسي.

الرد

نجاح (200 OK):

```
{
  "session_id": "xyz-789-abc-123",
  "status": "sent"
}
```

ردود الأخطاء:

حالة HTTP	الجسم	السبب
400	`{"error": "invalid request", "required": ["msisdn", "text", "callback_url"]}`	مفقود واحد أو أكثر من الحقول المطلوبة، أو الجسم ليس صالح JSON
500	`{"error": "{:gsm7_encode_failed, ...}"}`	النص يحتوي على أحرف ليست في أبجدية GSM 7-bit. يتم عرض فشل الترميز من خلال فرع الخطأ العام، لذا فإن الحالة هي **500**، وليس 400.
500	`{"error": "..."}`	مثل الاتصال) أي فشل آخر في الإرسال بـ M3UA). يحمل حقل `error` السبب الداخلي الذي تم فحصه.
503	`{"error": "USSD gateway not enabled"}`	`ussd_gateway_enabled` هو `false`

> **ملاحظة حول أخطاء التحقق مقابل أخطاء الإرسال:** مشاكل التحقق من الحقول (المفاتيح المفقودة، JSON غير صالح) تعيد **400** `invalid request`. بمجرد اجتياز التحقق، يتم الإبلاغ عن أي فشل من مسار الإرسال (بما في ذلك أخطاء ترميز GSM 7-bit) كـ **500** مع السبب الذي تم فحصه في حقل `error`.

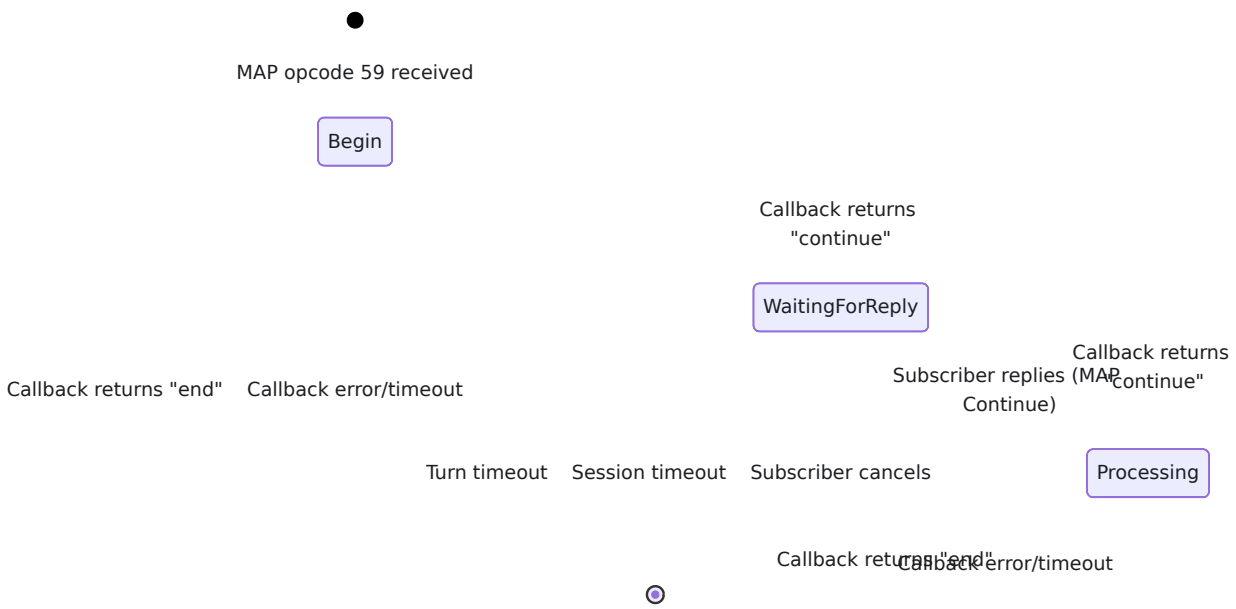
مثال cURL

```
curl -X POST http://localhost:9090/api/ussd/send \
  -H "Content-Type: application/json" \
  -d '{
    "msisdn": "+254712345678",
    "text": "لديك فاتورة معلقة. رد 1 للدفع.",
    "callback_url": "http://billing-app:9000/ussd"
  }'
```

دورة حياة الجلسة

يتم تتبع كل حوار USSD مع **جلسة** كـ `session_id` فريد.

حالات الجلسة



جلسة ذات دور واحد

إذا كان ردك يعيد `"end"` ترسل البوابة MAP End في الدور الأول، فلا يتم إنشاء جلسة مستمرة. مع النتيجة وتعود على الفور.

جلسة متعددة الأدوار

إذا كان ردك يعيد `"continue"`، فإن البوابة:

1. تنشئ **خادم جلسة** **GenServer** مسجل في `UssdGateway.Registry`
2. ترسل MAP Continue مع opcode 60 (unstructuredSS-Request) إلى الهاتف المحمول
3. تنتظر رد المشترك (حتى `turn_timeout_ms`)
4. تعيد توجيه الرد إلى ردك
5. تكرر حتى يعود ردك بـ `"end"` أو يحدث مهلة

سلوك المهلة

المهلة	الافتراضي	التأثير
مهلة الدور	30 ثانية	إذا لم يرد المشترك ضمن هذه النافذة، يتم إنهاء الجلسة بخطأ MAP.
مهلة الجلسة	3 دقائق	إجمالي مدة الجلسة. تنهي الجلسة بغض النظر عن النشاط.
مهلة رد HTTP	5 ثواني	إذا لم يستجب تطبيقك في الوقت المحدد، ترسل البوابة خطأ MAP إلى الهاتف المحمول وتنهى الجلسة.

معالجة الأخطاء

تتعامل البوابة مع الفشل بشكل سلس وتحاول دائمًا إرسال رد خطأ MAP إلى الهاتف المحمول حتى يرى المشترك رسالة ذات معنى بدلاً من مهلة الشبكة.

السيناريو	إجراء البوابة	رمز خطأ MAP
مهلة رد HTTP 5xx أو	إنهاء الجلسة، إرسال MAP End مع خطأ	34 (systemFailure)
غير صالح من الرد JSON	إنهاء الجلسة، إرسال MAP End مع خطأ	34 (systemFailure)
نص USD يتجاوز `max_text_length`	اقتطاع النص، تسجيل تحذير، الاستمرار بشكل طبيعي	N/A (مقتطع، ليس خطأ)
مهلة المشترك (لا رد)	إنهاء الجلسة، إرسال MAP End مع خطأ	34 (systemFailure)
لا تطابق توجيه الرمز القصير	إرسال MAP End، مع خطأ تسجيل تحذير	34 (systemFailure)
تعطل خادم GenServer الجلسة	تموت الجلسة، يرى المشترك مهلة الشبكة	N/A (خروج العملية)
بوابة USD غير مفعلة	إرجاع Facility Not Supported	21 (facilityNotSupported)

المقاييس والمراقبة

تقوم بوابة USD بتعريض مقاييس Prometheus على نقطة النهاية القياسية `/metrics` (المنفذ 9090).

مقاييس USD

المعالجة USD **النوع**: عداد **الوصف**: إجمالي طلبات `ussd_requests_total` **المقياس**: **العلامات**:

- `direction`: `"inbound"` (منشأ من الهاتف المحمول) أو `"outbound"` (منشأ من الشبكة)

المقياس: `ussd_active_sessions` **النوع**: مقياس **الوصف**: تم إعلانه كمقياس لعدد جلسات USSD النشطة.

> **تحذير:** هذا المقياس **معلن ولكن لم يتم تحديثه** في البناء الحالي: يقرأ دائمًا `0`. لا تعتمد عليه لعد الجلسات الحية. تتبع نشاط الجلسة عبر `ussd_requests_total` بدلاً من ذلك.

المقياس: `map_request_duration_milliseconds` **النوع**: هيستوجرام **الوصف**: مدة عمليات إرسال USSD بالمللي ثانية **العلامات**:

- `operation`: `"ussd_send"` لطلبات الدفع الخارجي

المثال Prometheus استعلامات

```
# معدل طلبات USSD من منشأ من الهاتف المحمول مقابل منشأ من الشبكة (حسب الاتجاه)
rate(ussd_requests_total[5m])

# زمن استجابة USSD الخارجي (p95)
histogram_quantile(0.95,
rate(map_request_duration_milliseconds_bucket{operation="ussd_send"}[5m]))
```

خادم ردود المثال

بايثون (Flask)

```
from flask import Flask, request, jsonify

app = Flask(__name__)
sessions = {}

@app.route('/ussd', methods=['POST'])
def ussd():
    data = request.json
    session_id = data['session_id']
    text = data['text']
    turn = data['turn']

    if data['type'] == 'initiation':
        sessions[session_id] = {'state': 'main_menu'}
        return jsonify({
            'action': 'continue',
            'text': 'مرحبًا!\n1. تحقق من الرصيد\n2. شراء رصيد\n3. تحويل'
        })

    state = sessions.get(session_id, {}).get('state')

    if state == 'main_menu':
        if text == '1':
            del sessions[session_id]
            return jsonify({
                'action': 'end',
                'text': 'رصيدك هو 5.00 دولار'
            })
        elif text == '2':
            sessions[session_id]['state'] = 'buy_airtime'
            return jsonify({
                'action': 'continue',
                'text': 'أدخل المبلغ:'
            })
        else:
            del sessions[session_id]
            return jsonify({
```

```
                'action': 'end',
                'text': 'خيار غير صالح. وداعًا.'
            })

    elif state == 'buy_airtime':
        del sessions[session_id]
        return jsonify({
            'action': 'end',
            'text': f'لقد اشتريت رصيدًا بقيمة {text} دولار. شكرًا لك!'
        })

    return jsonify({'action': 'end', 'text': 'انتهت الجلسة.'})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=9000)
```

Node.js (Express)

```
const express = require('express');
const app = express();
app.use(express.json());

const sessions = new Map();

app.post('/ussd', (req, res) => {
  const { session_id, text, type } = req.body;

  if (type === 'initiation') {
    sessions.set(session_id, { state: 'main_menu' });
    return res.json({
      action: 'continue',
      text: 'مرحبًا!\n1. تحقق من الرصيد\n2. شراء رصيد'
    });
  }

  const session = sessions.get(session_id);
  if (!session) {
    return res.json({ action: 'end', text: 'انتهت الجلسة.' });
  }

  if (session.state === 'main_menu' && text === '1') {
    sessions.delete(session_id);
    return res.json({ action: 'end', text: 'رصيدك هو 5.00 دولار'
});
  }

  sessions.delete(session_id);
  return res.json({ action: 'end', text: 'وداعًا.' });
});

app.listen(9000, () => console.log('ردود USSD 9000 على المنفذ'));
```

استكشاف الأخطاء

يعود USSD "الخدمة غير متاحة"

الأعراض: يقوم المشترك بالاتصال برمز قصير ويحصل على خطأ شبكة على الفور.

الأسباب المحتملة:

- `ussd_gateway_enabled` هو `false`
- لا يوجد توجيه يتطابق مع الرمز القصير المدخل
- اتصال M3UA معطل

الحل:

1. تحقق من `ussd_gateway_enabled: true` في التكوين
2. تحقق من أن نمط التوجيه يتطابق مع الرمز القصير (تذكر تضمين `*` كخيار احتياطي)
3. تحقق من حالة نظير M3UA في واجهة الويب (صفحة النظائر)

لم يتلق ردود التطبيق الطلبات

الأعراض: تظهر سجلات البوابة بدء USSD لكن تطبيقك لا يتلقى HTTP POST أبدًا.

الأسباب المحتملة:

- عنوان URL للرد غير قابل للوصول من مضيف OmniSS7
- جدار حماية يحظر HTTP الصادر من OmniSS7
- تطبيق الرد غير قيد التشغيل

الحل:

1. اختبار الاتصال: `curl -v http://your-app:9000/ussd` من مضيف OmniSS7
2. تحقق من قواعد جدار الحماية لـ HTTP الصادر
3. تحقق من أن تطبيق الرد الخاص بك يستمع على المنفذ المكون

انتهاء الجلسات قبل الأوان

الأعراض: تنتهي الجلسات متعددة الأدوار مع "systemFailure" من المشترك بل أن يتمكن الرد.

الأسباب المحتملة:

- `turn_timeout_ms` قصير جدًا لقاعدة المشتركين لديك
- `http_timeout_ms` قصير جدًا لوقت معالجة تطبيقك
- تأخر الشبكة بين OmniSS7 وخادم ردك

الحل:

1. زيادة `turn_timeout_ms` (30 ثانية افتراضية يجب أن تكون كافية لمعظم الحالات)
2. زيادة `http_timeout_ms` إذا كان تطبيقك يحتاج إلى مزيد من وقت المعالجة
3. نشر خادم الرد بالقرب من OmniSS7 لتقليل التأخر

أخطاء ترميز GSM 7-bit

الأعراض: أخطاء `gsm7_encode_failed` في السجلات، أو استجابة **500** من `/api/ussd/send` التي تحتوي على حقل `error` يحتوي على `{:gsm7_encode_failed, ...}`.

الأسباب المحتملة:

- النص يحتوي على أحرف خارج الأبجدية الافتراضية GSM 7-bit (مثل الرموز التعبيرية، أحرف CJK)

الحل:

- تقييد نص USSD إلى مجموعة الأحرف الأساسية GSM: أحرف ASCII، أرقام، علامات الترقيم الشائعة، وبعض الأحرف اليونانية/النوردية
- راجع 3GPP TS 23.038 القسم 6.2.1 للحصول على جدول الأحرف الكامل

الوثائق ذات الصلة

- **دليل API**: مرجع كامل لـ REST API (جميع نقاط النهاية بما في ذلك `/api/ussd/send`)
- **دليل عميل MAP**: إعداد اتصال M3UA المطلوب لـ USSD
- **مرجع التكوين**: جميع معلمات التكوين
- **دليل الميزات الشائعة**: واجهة الويب، المراقبة، وإعداد Prometheus

دليل واجهة المستخدم على الويب

← العودة إلى الوثائق الرئيسية

يوفر هذا الدليل وثائق شاملة لاستخدام **واجهة المستخدم على الويب** (واجهة Phoenix LiveView) الخاصة بـ OmniSS7.

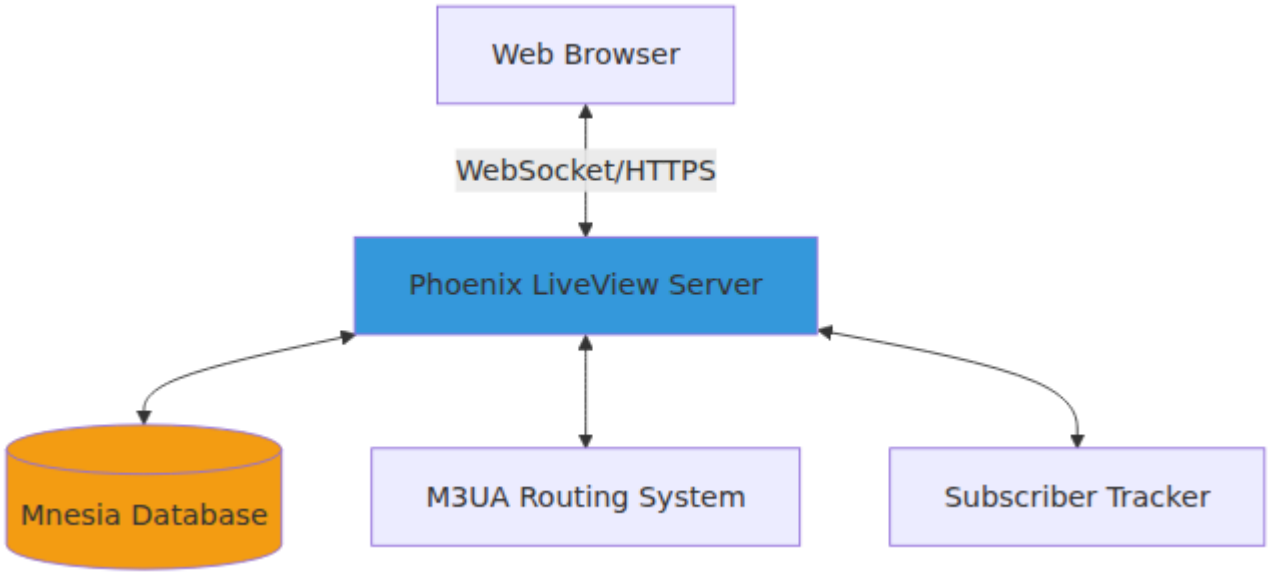
جدول المحتويات

1. نظرة عامة
2. الوصول إلى واجهة المستخدم على الويب
3. صفحة إدارة التوجيه
4. صفحة المشتركين النشطين
5. العمليات الشائعة
6. سلوك التحديث التلقائي

نظرة عامة

تعد واجهة المستخدم على الويب الخاصة بـ OmniSS7 تطبيق **Phoenix LiveView** يوفر قدرات المراقبة والإدارة في الوقت الفعلي. تعتمد الصفحات المتاحة على وضع التشغيل النشط (SMSc أو HLR أو STP).

هيكل واجهة المستخدم على الويب



تكوين الخادم

المعلمة	القيمة
البروتوكول	HTTPS (`force_ssl` مفعل)
المنفذ	`8087` (مكون تحت `config :control_panel, ControlPanelWeb.Endpoint` → `https: [port: 8087]`)
عنوان IP الافتراضي	`0.0.0.0` (يستمع على جميع الواجهات)
الشهادة	`priv/cert/omnitouch.crt`
المفتاح الخاص	`priv/cert/omnitouch.pem`
المهايئ	Bandit (Phoenix)

رابط الوصول: `https://[server-ip]:8087`

الوصول إلى واجهة المستخدم على الويب

المتطلبات الأساسية

1. **شهادات SSL**: تأكد من وجود شهادات SSL صالحة في `priv/cert/`:
   - `omnitouch.crt` - ملف الشهادة
   - `omnitouch.pem` - ملف المفتاح الخاص
2. **تشغيل التطبيق** `iex -S mix`: ابدأ التطبيق باستخدام
3. **جدار الحماية**: ت��كد من فتح المنفذ `8087` لحركة مرور HTTPS

الصفحات المتاحة حسب الوضع

تظهر تسمية الصفحة هنا بالضبط كما هو مسجل في `config/config.exs` (`use_builtin_pages` / `use_additional_pages`). تشير أعمدة الوضع إلى وضع التشغيل الذي تكون فيه الصفحة ذات صلة عادةً؛ تقوم تكوينات وقت التشغيل بتصفية (STP/HLR/SMSc) قائمة الصفحات حسب النشر.

الصفحة (المسار)	التسمية	وضع STP	وضع HLR	وضع SMSc	الوصف
`/application`	**الموارد**	✗	✗	✗	موارد النظام والمراقبة
`/configuration`	**التكوين**	✗	✗	✗	عارض التكوين
`/license`	**الرخصة**	✗	✗	✗	حالة الرخصة
`/events`	**أحداث SS7**	✗	✗	✗	تسجيل الأحداث والتقاط رسائل SCCP
`/logs`	**سجلات النظام**	✗	✗	✗	عارض سجلات التطبيق الحية
`/client`	**عميل SS7**	✗	✗	✗	اختبار العمليات اليدوية لـ MAP
`/m3ua`	**الأقران**	✗	✗	✗	حالة اتصال الأقران / ASP لـ M3UA
`/sctp`	**اتصالات SCTP**	✗	✗	✗	حالة ارتباط الحية SCTP
`/routing`	**التوجيه**	✗	✗	✗	إدارة جدول توجيه M3UA
`/routing_test`	**اختبار التوجيه**	✗	✗	✗	اختبار وتحقق من المسار

الصفحة (المسار)	التسمية	وضع STP	وضع HLR	وضع SMSc	الوصف
`/hlr_links`	**روابط HLR**	✗	✓	✗	حالة واجهة برمجة التطبيقات وإدارة HLR المشتركين
`/subscribers`	**المشتركين النشطين**	✗	✓	✗	تتبع موقع المشتركين في الوقت الفعلي (HLR)
`/smsc_links`	**روابط SMSc**	✗	✗	✓	حالة واجهة برمجة التطبيقات وإدارة SMSc الطوابير
`/smsc_subscribers`	**مشتركي SMSc**	✗	✗	✓	تتبع المشتركين في الوقت الفعلي (SMSc)
`/camel_sessions`	**جلسات CAMEL**	✗	✗	✓	مراقبة حوار CAMEL/CAP الحي
`/camel_request`	**منشئ طلب CAMEL**	✗	✗	✓	منشئ طلب تفاعلي CAP
`/msisdn_imsi_test`	**اختبار MSISDN/IMSI**	✗	✗	✗	اختبار البحث / التحويل بين

الوصف	وضع SMSc	وضع HLR	وضع STP	التسمية	الصفحة (المسار)
MSISDN ↔ IMSI					

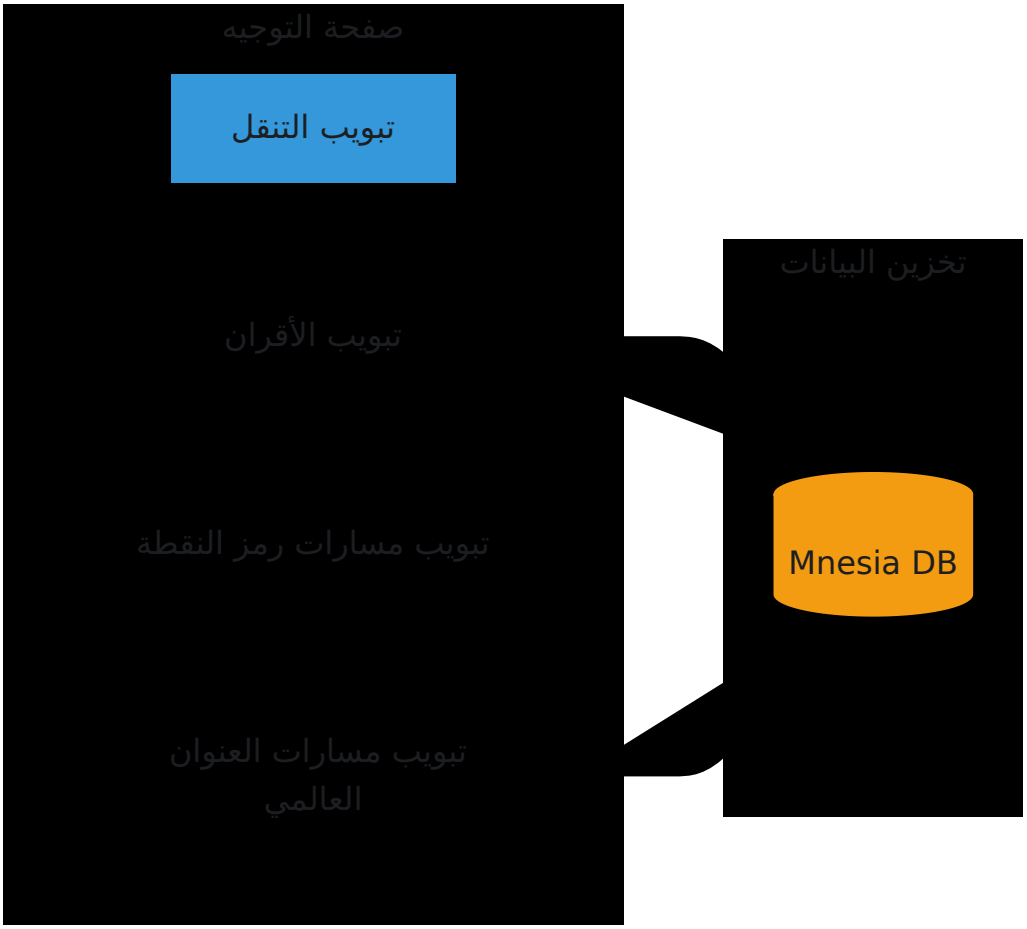
> يتم تسمية صفحة M3UA في التنقل "**الأقران**" (المسار `/m3ua`).

إدارة التوجيه

الصفحة: `/routing`
الأوضاع: STP، SMSc
التحديث التلقائي: كل 5 ثوانٍ

توفر صفحة إدارة التوجيه واجهة تبويب لإدارة جداول توجيه M3UA.

تخطيط الصفحة



تبويب الأقران

إدارة اتصالات الأقران M3UA (STPs الأخرى، HLRs، MSCs، SMSCs).

أعمدة جدول الأقران

العمود	الوصف	المثال
ID	معرف فريد للقرين	`1`
الاسم	اسم القرين قابل للقراءة البشرية	`"STP_West"`
الدور	دور الاتصال	`client`, `server`, `stp`
رمز النقطة	رمز النقطة SS7 للقرين	`100`
عن بُعد	IP:Port عن بُعد	`10.0.0.10:2905`
الحالة	حالة الاتصال	`active`, `aspup`, `down`
الإجراءات	أزرار التحرير / الحذف	-

إضافة قرين

1. **انقر** على تبويب الأقران
2. **املأ** حقول النموذج:
   - **معرف القرين**: يتم إنشاؤه تلقائيًا إذا ترك فارغًا
   - **اسم القرين**: اسم وصفي (مطلوب)
   - **الدور**: اختر `client` أو `server` أو `stp`
   - **رمز النقطة**: رمز النقطة SS7 (مطلوب)
   - **IP المحلي**: عنوان IP الخاص بنظامك
   - **المنفذ المحلي**: 0 لتعيين منفذ ديناميكي
   - **IP عن بُعد**: عنوان IP للقرين
   - **المنفذ عن بُعد**: منفذ القرين (عادةً 2905)
   - **سياق التوجيه**: معرف سياق توجيه M3UA
   - **مؤشر الشبكة**: `international` أو `national`
3. **انقر** على "إضافة قرين"

الاستمرارية: يتم حفظ القرين على الفور في Mnesia ويستمر بعد إعادة التشغيل.

تحرير قرين

1. **انقر** على زر "ت��رير" في صف القرين
2. **قم بتعديل** حقول النموذج حسب الحاجة
3. **انقر** على "تحديث القرين"

ملاحظة: إذا قمت بتغيير معرف القرين، سيتم حذف القرين القديم وإنشاء قرين جديد.

حذف قرين

1. **انقر** على زر "حذف" في صف القرين
2. **أكد** الحذف (سيتم أيضًا إزالة جميع المسارات التي تستخدم هذا القرين)

مؤشرات حالة القرين

الحالة	اللون	الوصف
`active`	أخضر	القرين متصل ويقوم بتوجيه الرسائل
`aspup`	أصفر	متصل ولكنه ليس نشطًا بعد ASP
`down`	أحمر	القرين غير متصل

تبويب مسارات رمز النقطة

تكوين قواعد التوجيه بناءً على رموز النقطة الوجهة.

أعمدة جدول المسار

العمود	الوصف	المثال
رمز النقطة الوجهة	رمز النقطة المستهدفة (بتنسيق zone.area.id)	`1.2.3` (100)
القناع	قناع الشبكة لمطابقة رمز النقطة	`/8`، (مط��بقة دقيقة) `/14` (نطاق)
معرف القرين	القرين المستهدف لهذا المسار	`1`
اسم القرين	اسم القرين المستهدف	`"STP_West"`
الأولوية	أولوية المسار (1 = الأعلى)	`1`
الشبكة	مؤشر الشبكة	`international`
الإجراءات	أزرار التحرير / الحذف	-

إضافة مسار رمز النقطة

1. **انقر** على تبويب "مسارات رمز النقطة"
2. **املأ** حقول النموذج:
   - **رمز النقطة الوجهة**: أدخل كـ `zone.area.id` (مثل `1.2.3`) أو عدد صحيح (0-16383)
   - **القناع**: اختر القناع `/14` للمطابقة الدقيقة، والقيم المنخفضة للنطاقات
   - **معرف القرين**: اختر القرين المستهدف من القائمة المنسدلة
   - **الأولوية**: أدخل الأولوية (1 = الأعلى، الافتراضي)
   - **مؤشر الشبكة**: اختر `international` أو `national`
3. **انقر** على "إضافة مسار"

تنسيق رمز النقطة: يمكنك إدخال رموز النقطة بتنسيقين:

- **تنسيق 3-8-3**: `zone.area.id` (مثل `1.2.3`)
- **تنسيق عدد صحيح**: `0-16383` (مثل `1100`)

يقوم النظام تلقائيًا بتحويل التنسيقات.

فهم الأقنعة

رموز النقطة هي قيم 14 بت (0-16383). يحدد القناع عدد بتات الأكثر أهمية التي يجب أن تتطابق:

القناع	رموز النقطة المطابقة	حالة الاستخدام
`/14`	1 (مطابقة دقيقة)	توجيه إلى وجهة محددة
`/13`	2 رموز نقطة	نطاق صغير
`/8`	64 رمز نقطة	نطاق متوسط
`/0`	جميع 16,384 رمز نقطة	**المسار الافتراضي / الاحتياطي**

أمثلة:

- `PC 1000 /14` → يتطابق فقط مع PC 1000
- `PC 1000 /8` → يتطابق مع PC 1000-1063 (64 رمز نقطة متتالية)
- `PC 0 /0` → يتطابق مع جميع رموز النقطة (المسار الافتراضي)

بطاقة مرجعية لقناع رمز النقطة

تتضمن صفحة الويب مرجعًا تفاعليًا يظهر جميع قيم الأقنعة ونطاقاتها.

تبويب مسارات العنوان العالمي

تكوين قواعد التوجيه بناءً على عناوين العنوان العالمي SCCP.

المتطلب: يجب تمكين توج��ه العنوان العالمي في التكوين:

```
config :omniss7,
  enable_gt_routing: true
```

أعمدة جدول المسار

العمود	الوصف	المثال
بادئة GT	بادئة GT للطرف المتصل (فارغ = احتياطي)	`"1234"`, `""`
SSN المصدر	SSN للطرف المتصل المطابقة على (اختياري)	`6` (HLR)، `any`
معرف القرين	القرين المستهدف	`1`
القرين	اسم القرين	`"HLR_West (1)"`
SSN الوجهة	SSN عند التوجيه (اختياري) إعادة كتابة	`6`, `preserve`
الأولوية	أولوية المسار	`1`
الوصف	وصف المسار	`"أرقام الولايات المتحدة"`
الإجراءات	أزرار التحرير / الحذف	-

إضافة مسار عنوان عالمي

1. انقر على تبويب "مسارات العنوان العالمي"
2. املأ حقول النموذج:
   - **بادئة GT**: اترك فارغًا للمسار الاحتياطي، أو أدخل أرقام (مثل `"1234"`)
   - **SSN المصدر**: اختياري - تصفية حسب SSN للطرف المتصل
   - **معرف القرين**: اختر القرين المستهدف
   - **SSN الوجهة**: اختياري - إعادة كتابة SSN عند التوجيه
   - **الأولوية**: أولوية المسار (1 = الأعلى)
   - **الوصف**: وصف قابل للقراءة البشرية
3. انقر على "إضافة مسار"

المسارات الاحتياطية: إذا كانت بادئة GT فارغة، يعمل المسار كمسار شامل لعناوين GT التي لا تتطابق مع أي مسار آخر.

قيم SSN الشائعة

تتضمن الصفحة بطاقة مرجعية بقيم SSN الشائعة:

SSN	عنصر الشبكة
6	HLR (سجل الموقع المنزلي)
7	VLR (سجل الموقع الزائر)
8	MSC (مركز التبديل المحمول)
9	EIR (سجل هوية المعدات)
10	AUC (مركز المصادقة)
142	RANAP
145	gsmSCF (وظيفة التحكم في الخدمة)
146	SGSN

إعادة كتابة SSN

- **المصدر** SSN للطرف المتصل في الرسائل الواردة: المطابقة على **SSN**
- **الوجهة** SSN للطرف المتصل عند التوجيه: إذا تم تعيينه، يعيد كتابة **SSN**
  - الأصلي SSN فارغ = الحفاظ على
  - القيمة SSN = الاستبدال بهذا

حالة الاستخدام: توجيه الرسائل مع SSN=7 إلى قرين، وإعادة كتابتها إلى SSN=6 (HLR) على الجانب الخارج (VLR).

استمرارية جدول التوجيه

تُخزن جميع المسارات في Mnesia وتستمر عبر إعادة تشغيل التطبيق.

كيفية استمرار المسارات

1. **تغييرات واجهة المستخدم على الويب**: يتم حفظ جميع عمليات الإضافة / التحرير / الحذف على الفور في Mnesia
2. **إعادة تشغيل التطبيق**: يتم تحميل المسارات من Mnesia عند بدء التشغيل
3. **دمج runtime.exs**: يتم دمج المسارات الثابتة من `config/runtime.exs` مع مسارات Mnesia (بدون تكرارات)

أولوية المسار

عند تطابق عدة مسارات مع وجهة:

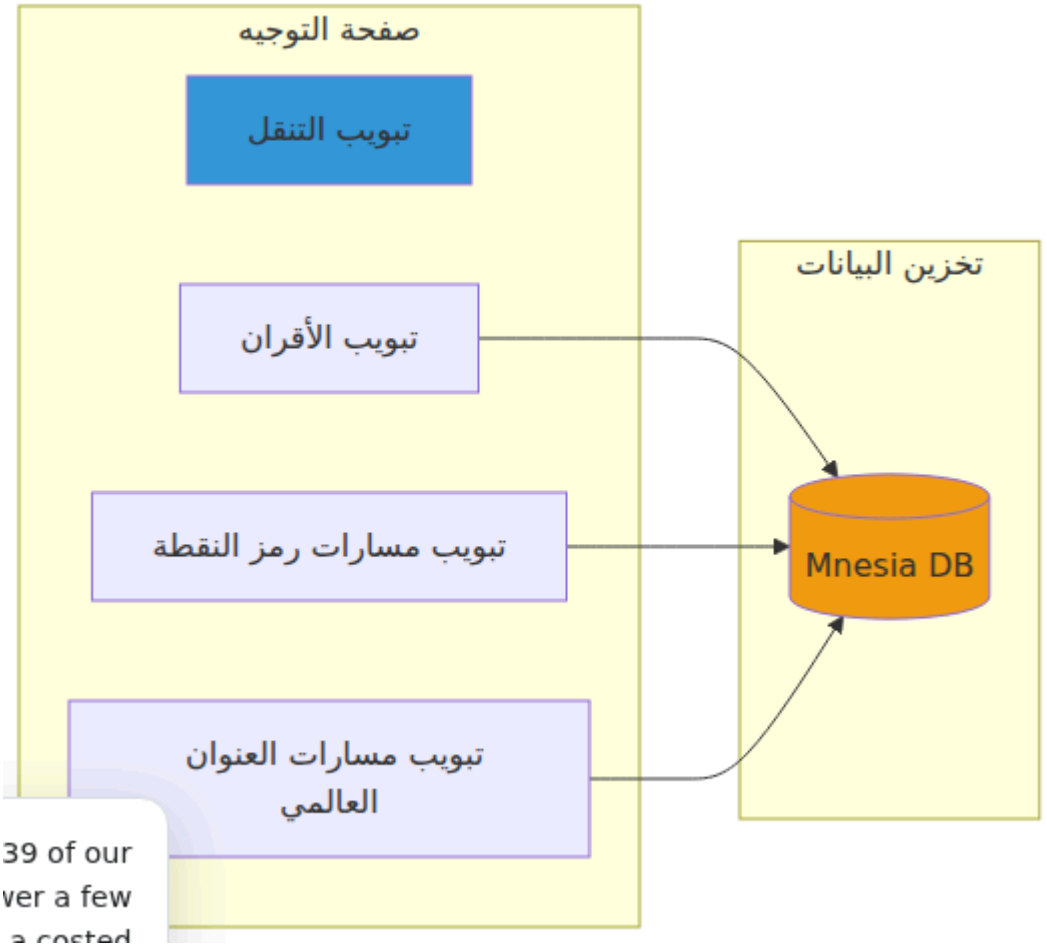
1. **الأكثر تحديدًا أولاً**: تأخذ القيم الأعلى للقناع (الأكثر تحديدًا) الأولوية
2. **حقل الأولوية**: يتم توجيه الأرقام ذات الأولوية المنخفضة أولاً (1 = أعلى أولوية)
3. **حالة القرين**: يتم استخدام المسارات فقط إلى الأقر��ن `active`

المشتركين النشطين

الصفحة: `/subscribers`
الوضع: فقط HLR
التحديث التلقائي: كل 2 ثانية

يعرض تتبعًا في الوقت الفعلي للمشتركين الذين أرسلوا طلبات UpdateLocation.

ميزات الصفحة



أعمدة جدول المشتركين

العمود	الوصف	المثال
IMSI	IMSI المشترك	`"505571234567 89"`
رقم VLR	عنوان GT الحالي لـ VLR	`"555123155"`
رقم MSC	عنوان GT الحالي لـ MSC	`"555123155"`
تم التحديث في	آخر طابع زمني لـ UpdateLocation	`"2025-10-25 14:23:45 UTC"`
المدة	الوقت منذ التسجيل	`"2h 15m 34s"`

ملخص الإحصائيات

عندما يكون هناك مش��ركين، تعرض بطاقة ملخص:

- **الإجمالي النشط**: إجمالي عدد المشتركين المسجلين
- **VLRs الفريدة**: عدد عناوين VLR المتميزة
- **MSCs الفريدة**: عدد عناوين MSC المتميزة

مسح المشتركين

زر مسح الكل: يزيل جميع المشتركين النشطين من المتعقب.

التأكيد: يتطلب تأكيدًا قبل المسح (لا يمكن التراجع عنه).

حالة الاستخدام: مسح سجلات المشتركين القديمة بعد صيانة الشبكة أو الاختبار.

التحديث التلقائي

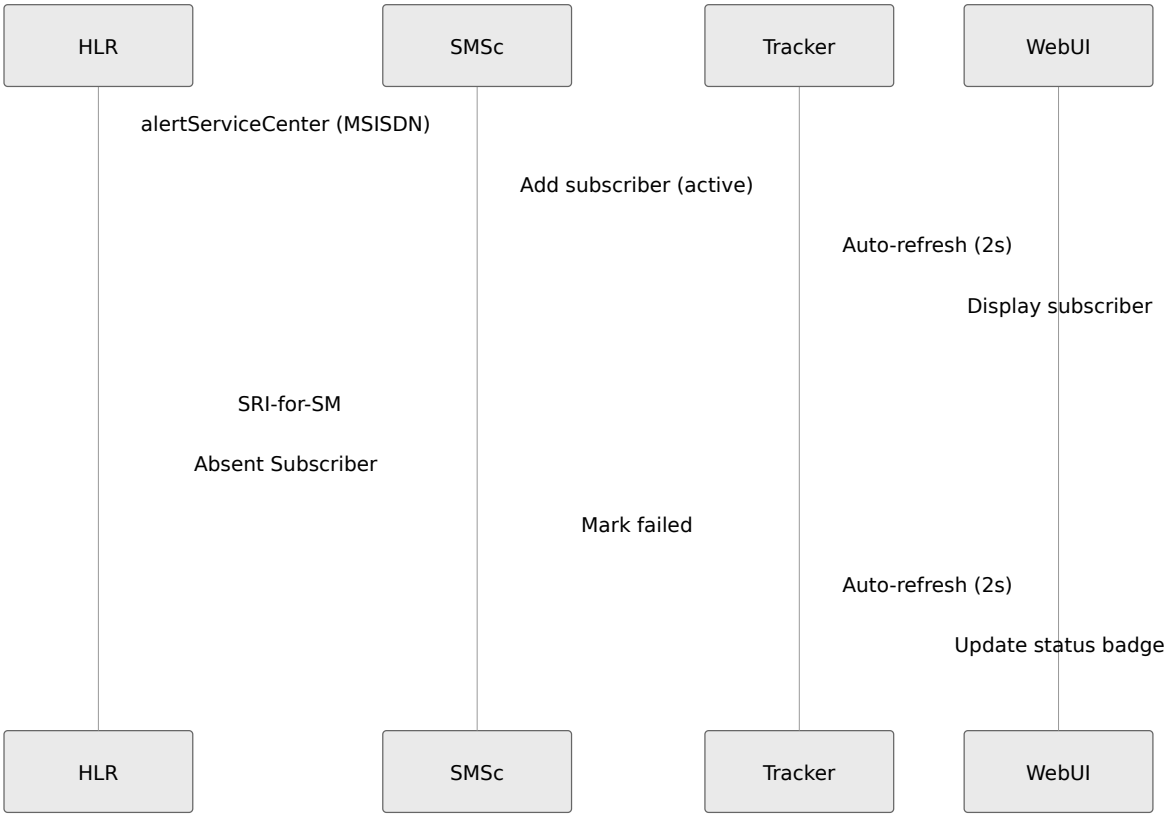
تقوم الصفحة بتحديث نفسها تلقائيًا كل **2 ثانية** لعرض تحديثات المشتركين في الوقت الفعلي.

SMSc مشتركي

الصفحة: `/smsc_subscribers`
الوضع: SMSc فقط
التحديث التلقائي: كل 2 ثانية

يعرض تتبعًا في الوقت الفعلي للمشتركين بناءً على رسائل alertServiceCenter المستلمة من HLRs، وحالة تسليم الرسائل، وتتبع الفشل.

ميزات الصفحة



أعمدة جدول المشتركين

العمود	الوصف	المثال
MSISDN	رقم هاتف المشترك	`"15551234567"`
IMSI	IMSI المشترك	`"001010123456789"`
HLR GT	HLR GT الذي أرسل alertServiceCenter	`"15551111111"`
عدد الرسائل المرسلة	عدد رسائل MT-FSM المرسلة	`5`
عدد الرسائل المستلمة	عدد رسائل MO-FSM المستلمة	`2`
الحالة	`نشط` أو `فشل` (ملونة)	`● نشط`
آخر تحديث	طابع زمني آخر تحديث	`"2025-10-30 14:23:45 UTC"`
المدة	الوقت منذ آخر تحديث	`"15m 34s"`

مؤشرات الحالة

- ● **نشط** (أخضر): المشترك قابل للوصول، آخر alertServiceCenter تم استلامه بنجاح
- ○ **فشل** (أحمر): فشلت آخر محاولة تس��يم (أو مشترك غائب SRI-for-SM خطأ)

ملخص الإحصائيات

عندما يكون هناك مشتركين، تعرض بطاقة ملخص:

- **الإجمالي المتعقب**: إجمالي عدد المشتركين المتعقبين
- **النشط**: عدد المشتركين بحالة نشطة

- **الفشل**: عدد المشتركين بحالة فشل
- **الفريدة HLRs**: عدد HLRs المتميزة التي ترسل التنبيهات

إدارة المشتركين

زر إزالة: يزيل مشتركًا فرديًا من المتعقب.

زر مسح الكل: يزيل جميع المشتركين المتعقبين.

التأكيد: يتطلب مسح الكل تأكيدًا قبل المسح (لا يمكن التراجع عنه).

حالة الاستخدام:

- إزالة الإدخالات القديمة بعد مشكلات الشبكة
- مسح بيانات الاختبار بعد التطوير
- مراقبة HLRs التي ترسل التنبيهات

عدادات الرسائل

يقوم المتعقب بزيادة العدادات تلقائيًا:

- **الرسائل المرسلة**: تزداد عند نجاح MT-FSM وإرسال SRI-for-SM
- **الرسائل المستلمة**: تزداد عند استلام MO-FSM من المشترك

الت��ديث التلقائي

تقوم الصفحة بتحديث نفسها تلقائيًا كل **2 ثانية** لعرض تحديثات المشتركين وحالتهم في الوقت الفعلي.

العمليات الشائعة

البحث والتصفية

حاليًا، لا تتضمن واجهة المستخدم على الويب وظيفة بحث / تصفية مدمجة. للعثور على مسارات محددة:

1. استخدم وظيفة البحث في متصفحك (Ctrl+F / Cmd+F)
2. ابحث عن أسماء الأقران أو رموز النقطة أو بادئات GT

العمليات الجماعية

لإجراء تغييرات جماعية على المسارات:

1. **الخيار 1**: استخدم واجهة برمجة التطبيقات REST للوصول البرمجي
2. **الخيار 2**: تحرير `config/runtime.exs` وإعادة تشغيل التطبيق
3. **الخيار 3**: استخدم واجهة المستخدم على الويب لتغييرات المسار الفردية

التصدير / الاستيراد

ملاحظة: لا تدعم واجهة المستخدم على الويب حاليًا تصدير أو استيراد جداول التوجيه. المسارات هي:

- مخزنة في ملفات قاعدة بيانات Mnesia
- مكونة في `config/runtime.exs`

لنسخ احتياطي للمسارات:

1. **Mnesia**: نسخ احتياطي لمجلد `Mnesia.{node_name}/`
2. **التكوين**: التحكم في إصدار `config/runtime.exs`

سلوك التحديث التلقائي

تمتلك الصفحات المختلفة فترات تحديث مختلفة:

الصفحة	فترة التحديث	السبب
إدارة التوجيه	5 ثوانٍ	تغييرات المسار نادرة
المشتركين النشطين	2 ثانٍ	تتغير حالة المشتركين بشكل متكرر
حالة M3UA	تختلف حسب الصفحة	مراقبة حالة الاتصال

اتصال WebSocket: تستخدم جميع الصفحات اتصالات WebSocket الخاصة بـ Phoenix LiveView للتحديثات في الوقت الفعلي.

انقطاع الشبكة: إذا فقد اتصال WebSocket، ستقوم الصفحة بمحاولة إعادة الاتصال تلقائيًا.

استكشاف الأخطاء وإصلاحها

الصفحة لا تُحمَل

1. **تحقق من شهادة HTTPS**: تأكد من وجود `priv/cert/omnitouch.crt` و `priv/cert/omnitouch.pem`
2. **تحقق من المنفذ 8087**: تحقق من أن قواعد جدار الحماية تسمح بحركة مرور HTTPS على المنفذ `8087`
3. **تشغيل التطبيق**: تأكد من تشغيل التطبيق باستخدام `iex -S mix`
4. **وحدة التحكم في المتصفح**: تحقق من أخطاء شهادة SSL (تحذيرات الشهادات الموقعة ذاتيًا)

المسارات لا تستمر

1. **تحقق من تخزين Mnesia**: تحقق من `mnesia_storage_type: :disc_copies` في التكوين
2. **مجلد Mnesia**: تأكد من أن مجلد Mnesia قابل للكتابة
3. **تحقق من السجلات**: ابحث عن أخطاء Mnesia في سجلات التطبيق

التحديث التلقائي لا يعمل

1. **اتصال WebSocket**: تحقق من وحدة التحكم في المتصفح بحثًا عن أخطاء WebSocket
2. **الشبكة**: تحقق من اتصال الشبكة المستقر
3. **إعادة تحميل الصفحة**: حاول تحديث الصفحة (F5)

الوثائق ذات الصلة

- **دليل STP** - تكوين التوجيه بالتفصيل
- **دليل HLR** - إدارة المشتركين
- **دليل API** - واجهة برمجة التطبيقات REST للوصول البرمجي
- **مرجع التكوين** - جميع معلمات التكوين

الملخص

إدارة بديهية وفي الوقت الفعلي لجداول OmniSS7 توفر واجهة المستخدم على الويب الخاصة بـ التوجيه وتتبع المشتركين:

 تحديثات في الوقت الحقيقي - التحديث التلقائي يحافظ على البيانات محدثة
 تخزين دائم - تضمن Mnesia بقاء المسارات عبر إعادة التشغيل
 واجهة مستخدم قائمة على الدور - تتكيف الصفحات مع وضع التشغيل (STP/HLR/SMSc)
 إدارة تفاعلية - إضافة وتحرير وحذف المسارات دون إعادة التشغيل
 مراقبة الحالة - حالة الاتصال والقرين الحية

لعمليات متقدمة أو أتمتة، راجع دليل API.

أمثلة متقدمة على توجيه العنوان العالمي

← العودة إلى الوثائق الرئيسية

تقدم هذه الوثيقة أمثلة عملية لاستخدام نوع الترجمة (TT) ومؤشر خطة الترقيم (NPI) ومؤشر طبيعة العنوان (NAI) لسيناريوهات توجيه GT المتقدمة.

جدول المحتويات

1. المفاهيم الأساسية
2. أمثلة على نوع الترجمة (TT)
3. أمثلة على مؤشر خطة الترقيم (NPI)
4. أمثلة على طبيعة العنوان (NAI)
5. أمثلة مجمعة
6. حالات الاستخدام في العالم الحقيقي

المفاهيم الأساسية

مرجع قيم المؤشرات

نوع الترجمة (TT):

- `0` = غير معروف
- `1` = دولي
- `2` = وطني
- `3` = محدد الشبكة

مؤشر خطة الترقيم (NPI):

- `0` = غير معروف

- `1` = ISDN/الهاتف (E.164)
- `3` = بيانات (X.121)
- `4` = تيليكس (F.69)
- `6` = الهاتف المحمول (E.212)

مؤشر طبيعة العنوان (NAI):

- `0` = غير معروف
- `1` = رقم المشترك
- `2` = محجوز للاستخدام الوطني
- `3` = رقم وطني مهم
- `4` = رقم دولي

المطابقة مقابل التحويل

- **مؤشرات المصدر** (source_tt, source_npi, source_nai): تستخدم لـ **مطابقة** الرسائل الواردة
- **مؤشرات الوجهة** (dest_tt, dest_npi, dest_nai): تستخدم لـ **تحويل** القيم عند إعادة التوجيه
- اترك الحقول فارغة (`nil`) للمطابقة العامة أو للحفاظ على القيم الأصلية

أمثلة على نوع الترجمة (TT)

المثال 1: تحويل TT دولي إلى غير معروف

السيناريو: شبكة الشركاء ترسل جميع GTs مع TT=0 (غير معروف)، لكن HLR الخاص بك يتوقع TT=1 (دولي).

```
%{
  gt_prefix: "44",
  peer_id: 1,
  priority: 1,
  description: "تحويل :المملكة المتحدة TT إلى دولي معروف غير",
  source_tt: 0,      # مطابقة TT 0=الوارد
  dest_tt: 1         # تحويل إلى TT=1
}
```

تدفق الحركة:

- الوارد: GT=441234567890، TT=0
- التوجيه: يتطابق مع gt_prefix="44"، source_tt=0
- الصادر: GT=441234567890، TT=1 (تم تحويله)

المثال 2: طرق منفصلة حسب نوع الترجمة

السيناريو: توجيه قيم TT المختلفة إلى شبكات مختلفة.

```
# توجيه TT=0 إلى الشبكة A
%{
  gt_prefix: "33",
  peer_id: 1,
  priority: 1,
  description: "الأرقام الفرنسية مع TT=0 إلى الشبكة A",
  source_tt: 0
},

# توجيه TT=2 إلى الشبكة B
%{
  gt_prefix: "33",
  peer_id: 2,
  priority: 1,
  description: "الأرقام الفرنسية مع TT=2 إلى الشبكة B",
  source_tt: 2
}
```

تدفق الحركة:

- GT="33123456789"، TT=0 → 1 يتم توجيهها إلى النظير
- GT="33123456789"، TT=2 → 2 يتم توجيهها إلى النظير

أمثلة على مؤشر خطة الترقيم (NPI)

المثال 1: تحويل ISDN من الهاتف المحمول إلى NPI

السيناريو: تستخدم شبكة الهاتف المحمول NPI=6 (E.212)، ويتوقع PSTN NPI=1 (E.164).

```
%{
  gt_prefix: "49",
  peer_id: 1,
  priority: 1,
  description: "تحويل الهاتف المحمول الألماني→ISDN",
  source_npi: 6,     # مطابقة NPI=6 (الهاتف المحمول/E.212)
  dest_npi: 1        # تحويل إلى NPI=1 (ISDN/E.164)
}
```

حالة الاستخدام: الربط بين الشبكات المحمولة والثابتة مع خطط ترقيم مختلفة.

المثال 2: توجيه بناءً على خطة الترقيم

السيناريو: فصل حركة مرور الشبكة البيانات عن حركة مرور الصوت.

```
# توجيه حركة مرور الشبكة البيانات (NPI=3)
%{
  gt_prefix: "1",
  peer_id: 1,
  priority: 1,
  description: "حركة مرور الشبكة البيانات",
  source_npi: 3      # مطابقة NPI=3 (بيانات/X.121)
},

# توجيه حركة مرور الشبكة الصوتية (NPI=1)
%{
  gt_prefix: "1",
  peer_id: 2,
  priority: 1,
  description: "حركة مرور الشبكة الصوتية",
  source_npi: 1      # مطابقة NPI=1 (ISDN)
}
```

أمثلة على طبيعة العنوان (NAI)

المثال 1: تحويل NAI من دولي إلى وطني

السيناريو: تنسيق وارد دولي (NAI=4)، تحو��ل إلى تنسيق وطني (NAI=3) للتوجيه المحلي.

```
%{
  gt_prefix: "1",
  peer_id: 1,
  priority: 1,
  description: "الولايات المتحدة: تحويل دولي→وطني",
  source_nai: 4,     # مطابقة NAI=4 (دولي)
  dest_nai: 3        # تحويل إلى NAI=3 (وطني)
}
```

تدفق الحركة:

- الوارد: GT=12125551234، NAI=4 (تنسيق دولي)
- الصادر: GT=12125551234، NAI=3 (تنسيق وطني)

المثال 2: توجيه أرقام المشتركين مقابل أرقام الشبكة NAI

السيناريو: توجيه بناءً على ما إذا كان رقم مشترك أو رقم شبكة.

```
# توجيه أرقام المشتركين إلى HLR
%{
  gt_prefix: "44",
  peer_id: 1,
  priority: 1,
  description: "أرقام المشتركين في المملكة المتحدة إلى HLR",
  source_nai: 1      # مطابقة NAI=1 (رقم المشترك)
},

# توجيه أرقام الشبكة إلى MSC
%{
  gt_prefix: "44",
  peer_id: 2,
  priority: 1,
  description: "عناوين الشبكة في المملكة المتحدة إلى MSC",
  source_nai: 3      # مطابقة NAI=3 (رقم وطني مهم)
}
```

أمثلة مجمعة

المثال 1: تطبيع حركة مرور SMS

السيناريو: تطبيع جميع حركة مرور SMS من مصادر مختلفة قبل توجيهها إلى HLR.

```
%{
  gt_prefix: "86",
  source_ssn: 8,           # مطابقة SSN=8 (SMSC)
  peer_id: 1,
  dest_ssn: 6,             # إعادة كتابة إلى SSN=6 (HLR)
  priority: 1,
  description: "تطبيع SMS الصينية",
  source_tt: 0,            # مطابقة TT=0 (غير معروف)
  dest_tt: 2,              # تحويل إلى TT=2 (وطني)
  source_npi: 6,           # مطابقة NPI=6 (الهاتف المحمول)
  dest_npi: 1,             # تحويل إلى NPI=1 (ISDN)
  source_nai: 4,           # مطابقة NAI=4 (دولي)
  dest_nai: 3              # تحويل إلى NAI=3 (وطني)
}
```

ما يفعله هذا:

- يتطابق مع حركة مرور SMS (SSN=8) للأرقام الصينية (GT prefix "86")
- يطبع جميع المؤشرات إلى القيم القياسية
- يعيد كتابة SSN الوجهة من SMSC (8) إلى HLR (6)
- مثالي لسيناريوهات الربط مع استخدام غير متسق للمؤشرات

المثال 2: توجيه متعدد الناقلين مع تحويل التنسيقات

السيناريو: توجيه إلى ناقلين مختلفين بناءً على TT، مع تح��يل التنسيقات حسب الحاجة.

```
# الناقل A: يتوقع TT وطني و NPI ISDN
%{
  gt_prefix: "44",
  peer_id: 1,
  priority: 1,
  description: "الناقل A تطبيع إلى وطني: المملكة المتحدة/ISDN",
  source_tt: 0,           # أي مطابقة TT=0
  dest_tt: 2,             # تحويل إلى TT=2 (وطني)
  source_npi: 6,          # مطابقة NPI المحمول
  dest_npi: 1             # تحويل إلى NPI ISDN
},

# الناقل B: يتوقع TT دولي، الحفاظ على NPI
%{
  gt_prefix: "44",
  peer_id: 2,
  priority: 2,            # أولوية أقل (رقم أعلى)
  description: "الناقل B تحويل إلى: المملكة المتحدة في TT دولي",
  source_tt: 2,           # مطابقة TT وطني
  dest_tt: 1              # تحويل إلى TT دولي
  # تم ترك NPI فارغًا = الحفاظ على الأصل
}
```

حالات الاستخدام في العالم الحقيقي

حالة الاستخدام 1: تكامل النظام القديم

المشكلة: دمج الشبكة الحديثة مع النظام القديم الذي يستخدم مؤشرات مختلفة.

الحل: تحويل جميع المؤشرات إلى تنسيق النظام القديم.

```
%{
  gt_prefix: "555",
  peer_id: 1,
  priority: 1,
  description: "تطبيع HLR القديم",
  # مطابقة أي تنسيق حديث
  source_tt: 1,            # حديث: دولي
  dest_tt: 0,              # قديم: غير معروف
  source_npi: 1,           # حديث: ISDN
  dest_npi: 0,             # قديم: غير معروف
  source_nai: 4,           # حديث: دولي
  dest_nai: 0              # قديم: غير معروف
}
```

حالة الاستخدام 2: توحيد الربط الدولي

المشكلة: يرسل العديد من الشركاء الدوليين حركة مرور بمؤشرات غير متسقة.

الحل: تطبيع جميع حركة المرور الواردة إلى تنسيق متسق.

```
# الشريك A يرسل TT=0، تحويل إلى TT 1=القياسي
%{
  gt_prefix: "1",
  peer_id: 1,
  priority: 1,
  description: "تطبيع الشريك A في الولايات المتحدة",
  source_tt: 0,
  dest_tt: 1,
  dest_npi: 1,            # ضمان NPI ISDN
  dest_nai: 4             # ضمان NAI دولي
},

# الشريك B يرسل TT=3 (محدد الشبكة)، تحويل إلى TT=1
%{
  gt_prefix: "44",
  peer_id: 2,
  priority: 1,
  description: "تطبيع الشريك B في المملكة ا��متحدة",
  source_tt: 3,
  dest_tt: 1,
  dest_npi: 1,
  dest_nai: 4
}
```

حالة الاستخدام 3: إدارة حركة المرور أثناء التجوال

المشكلة: تستخدم الشبكة المنزلية والشركاء المتجولين تنسيقات عنوان مختلفة.

الحل: تحويل حركة المرور المتجولة إلى تنسيق الشبكة المنزلية.

```
# التجوال الوارد: تحويل إلى تنسيق الشبكة المنزلية
%{
  gt_prefix: "12025",      # منطقة واشنطن العاصمة
  peer_id: 1,
  priority: 1,
  description: "تطبيع التجوال الوارد",
  source_nai: 4,           # دولي من الشريك المتجول
  dest_nai: 3,             # وطني للشبكة المنزلية
  source_npi: 6,           # شبكة الهاتف المحمول
  dest_npi: 1              # المنزلية HLR لـ ISDN
}
```

حالة الاستخدام 4: ترحيل الشبكة

المشكلة: الانتقال من نظام العناوين القديم إلى نظام جديد بشكل تدريجي.

الحل: توجيه بناءً على مؤشرات المصدر، وتحويل للوجهة.

```
# حركة المرور بتنسيق قديم: تحويل إلى تنس��ق جديد
%{
  gt_prefix: "999",
  peer_id: 1,
  priority: 1,
  description: "الانتقال من التنسيق القديم إلى الجديد",
  source_tt: 0,           # قديم: غير معروف
  dest_tt: 2,             # جديد: وطني
  source_npi: 0,          # قديم: غير معروف
  dest_npi: 1,            # جديد: ISDN
  source_nai: 0,          # قديم: غير معروف
  dest_nai: 3             # جديد: وطني
},

# حركة المرور بتنسيق جديد: تمرير بدون تغيير
%{
  gt_prefix: "999",
  peer_id: 1,
  priority: 2,
  description: "تمرير التنسيق الجديد",
  source_tt: 2,           # بالفعل بتنسيق جديد
  # الحفاظ على جميع القيم = dest_ لا توجد حقول
}
```

اختبار طرقك

عبر واجهة الويب

1. انتقل إلى **التوجيه ← طرق العنوان العالمي**
2. انقر على **"عرض التوجيه المتقدم (TT/NPI/NAI)"**
3. املأ مؤشرات المصدر والوجهة الخاصة بك
4. انقر على **إضافة طريق**
5. انتقل إلى صفحة **اختبار التوجيه** لاختبار الطريق

عبر API

نقطة نهاية طريق GT موجودة في `/routing/gt-routes` على مستمع API (المنفذ **8445**، مع تمكين TLS). يستخدم جسم JSON الحقل `prefix` (ليس `gt_prefix`)، و `prefix` و `peer_id` مطلوبان. تعتبر مطابقة TT/NPI/NAI اختيارية وإعادة الكتابة.

```
curl -k -X POST https://localhost:8445/routing/gt-routes \
  -H "Content-Type: application/json" \
  -d '{
    "prefix": "44",
    "peer_id": 1,
    "priority": 1,
    "description": "تحويل TT في المملكة المتحدة",
    "source_tt": 0,
    "dest_tt": 1
  }'
```

الحقل	النوع	مطلوب	الوصف
`prefix`	سلسلة	نعم	بادئة GT للمطابقة
`peer_id`	عدد صحيح	نعم	معرف النظير المستهدف (`0` = إسقاط)
`priority`	عدد صحيح	لا	أولوية الطريق (افتراضي `1`، الأقل يفوز)
`description`	سلسلة	لا	وصف قابل للقراءة البشرية
`source_ssn` / `dest_ssn`	عدد صحيح	لا	مطابقة / إعادة كتابة SSN للمتصل
`source_tt` / `dest_tt`	عدد صحيح	لا	مطابقة / إعادة كتابة نوع الترجمة
`source_npi` / `dest_npi`	عدد صحيح	لا	مطابقة / إعادة كتابة مؤشر خطة الترقيم
`source_nai` / `dest_nai`	عدد صحيح	لا	مطابقة / إعادة كتابة مؤشر طبيعة العنوان

لحذف طريق، قم بإصدار `DELETE` إلى `/routing/gt-routes/{prefix}:{peer_id}` (على سبيل المثال `/routing/gt-routes/44:1`).

اختبار مطابقة الطريق

استخدم صفحة اختبار التوجيه لمحاكاة حركة المرور الواردة مع مؤشرات محددة:

1. أدخل بادئة GT (مثل "441234567890")
2. قم بتعيين قيم NAI و NPI و TT
3. انقر على "اختبار الطريق"
4. راجع أي طريق يتطابق وما هي التحويلات التي تنطبق

استكشاف الأخطاء وإصلاحها

الطرق غير المطابقة

تحقق من ترتيب التخصيص:

1. الفائز هو أطول بادئة GT
2. ثم المصدر المحدد على العامة SSN
3. ثم المصدر المحدد على العامة TT
4. ثم المصدر المحدد على العامة NPI
5. ثم المصدر المحدد على العامة NAI
6. أخيرًا، الفائز هو أقل رقم أولوية

التحويلات غير المطبقة

تحقق من تعيين حقول dest_:

- تعني القيم الفارغة dest_tt/dest_npi/dest_nai "الحفاظ على القيمة الأصلية"
- يجب تعيين dest_* صراحة لتطبيق التحويل

مشاكل الطرق المكررة

يسمح بالتكرارات bag: جدول

- هو نوع GT جدول طرق `:bag` (يسمح بسجلات متعددة بنفس المفتاح)
- المفتاح هو `{gt_prefix, source_ssn}`
- استخدم TT/NPI/NAI لتمييز الطرق ذات نفس البادئة + SSN
- احذف الطرق المكررة غير المرغوب فيها

أفضل الممارسات

1. **وثق تحويلاتك:** استخدم أوصاف طرق وصفية
2. **اختبر بدقة:** استخدم صفحة اختبار التوجيه قبل الإنتاج
3. **راقب السجلات:** تحقق من سجلات INFO للتحويلات المطبقة

4. **ابدأ ببساطة:** أضف طرق بادئة أساسية أولاً، ثم أضف قواعد المؤشر
5. **استخدم الأحرف العامة بحكمة:** الحقول الفارغة source_* تتطابق مع أي قيمة
6. **احفظ عند الإمكان:** قم بتحويل ما يحتاج إلى تغيير فقط
7. **اعتبر التخصيص:** الطرق الأكثر تخصيصًا تت��اوز الأقل تخصيصًا

لمزيد من المعلومات، راجع:

- ← العودة إلى الوثائق الرئيسية
- دليل STP - الوثائق الكاملة للتوجيه
- مرجع التكوين - جميع خيارات التكوين
- دليل API - وثائق REST API

OmniSS7 بواسطة Omnitouch Network Services

توجيه العنوان العالمي M3UA STP

← العودة إلى الوثائق الرئيسية

نظرة عامة

يدعم M3UA STP **توجيه العنوان العالمي (GT)** بالإضافة إلى توجيه رمز النقطة (PC). يقوم بتوجيه رسائل GT بناءً على عنوان الطرف المدعو العالمي في طبقة SS7 بتوجيه SCCP، مع التراجع التلقائي إلى توجيه PC.

كيفية العمل

أولوية التوجيه

1. **توجيه GT (إذا تم تمكينه)**: استخراج GT المدعو من SCCP → البحث في جدول توجيه GT
2. **توجيه PC (تراجع)**: استخراج DPC من MTP3 → البحث في جدول توجيه PC

مطابقة بادئة GT

- يستخدم خوارزمية **مطابقة البادئة الأطول**
- مثال: GT "447123456789"
  - يطابق "447" (الطول 3) على "44" (الطول 2)
  - يتم التوجيه إلى النظير المكون لـ "447" بادئة

التكوين

تمكين توجيه GT

في `config/runtime.exs`:

```
config :omniss7,
  enable_gt_routing: true,  # تعيين إلى false توجيه لاستخدام PC فقط
```

تعريف طرق GT

```
config :omniss7,
  m3ua_gt_routes: [
    # توجيه أرقام الولايات المتحدة (تبدأ بـ 1234) إلى النظير 1
    %{gt_prefix: "1234", peer_id: 1, priority: 1, description:
"أرقام الولايات المتحدة"},

    # توجيه أرقام المملكة المتحدة (تبدأ بـ 44) إلى النظير 2
    %{gt_prefix: "44", peer_id: 2, priority: 1, description:
"أرقام المملكة المتحدة"},

    # مسار أكثر تحديدًا للهواتف المحمولة في المملكة المتحدة
(447xxx)
    %{gt_prefix: "447", peer_id: 2, priority: 1, description:
"الهواتف المحمولة في المملكة المتحدة"},

    # توجيه أرقام أسترالية (تبدأ بـ 61) إلى النظير 3
    %{gt_prefix: "61", peer_id: 3, priority: 1, description:
"أرقام أسترالية"}
  ]
```

> يتم شحن تكوين STP كامل وجاهز للتعديل في `config/stp_example_runtime.exs`. راجع دليل STP لمعرفة كيفية تمكينه.

أمثلة الاستخدام

المثال 1: توجيه GT الأساسي

الرسالة: SCCP UDT مع GT "المدعو" = "447700123456"

قرار التوجيه:

- GT "المستخرج": "447700123456"

- أفضل مطابقة: "447" (الهواتف المحمولة في المملكة المتحدة) → النظير 2
- تم توجيه الرسالة إلى النظير 2

المثال 2: GT غير موجود، التراجع إلى PC

الرسالة: SCCP UDT مع GT 331234567890" = المدعو"، DPC = 100

قرار التوجيه:

- GT 331234567890" :المستخرج"
- لم يتم العثور على مسار GT 33" لـ"
- التراجع إلى توجيه PC
- DPC 100 → النظير 1 (من جدول توجيه PC)
- تم توجيه الرسالة إلى النظير 1

المثال 3: GT موجود لا يوجد

الرسالة: M3UA DATA بدون SCCP، فقط MTP3 مع DPC = 200

قرار التوجيه:

- GT لا يوجد، لا توجد طبقة SCCP
- استخدم توجيه PC فقط
- DPC 200 → النظير 2
- تم توجيه الرسالة إلى النظير 2

إدارة وقت التشغيل

تُعرض هذه الوظائف بواسطة `M3UARouting` ويمكن استدعاؤها من وحدة تحكم `iex -S mix`.

استعلام طرق GT

```
# الحصول على جميع طرق GT
{:ok, routes} = M3UARouting.get_all_gt_routes()

# كل سجل هو عبارة عن مجموعة من 14 عنصر. الحقول هي:
# {:m3ua_gt_route, gt_prefix, source_ssn, peer_id, dest_ssn,
priority,
#  description, enabled, source_tt, dest_tt, source_npi, dest_npi,
#  source_nai, dest_nai}
#
# مثال على المخرجات (تكون source_ssn المؤشر/وحقول إعادة الكتابة
افتراضيًا nil):
[
  {:m3ua_gt_route, "1234", nil, 1, nil, 1, "أرقام الولايات
المتحدة", true, nil, nil, nil, nil, nil, nil},
  {:m3ua_gt_route, "44", nil, 2, nil, 1, "أرقام المملكة المتحدة",
true, nil, nil, nil, nil, nil, nil},
  {:m3ua_gt_route, "447", nil, 2, nil, 1, "الهواتف المحمولة في
المملكة المتحدة", true, nil, nil, nil, nil, nil, nil},
  {:m3ua_gt_route, "61", nil, 3, nil, 1, "أرقام أسترالية", true,
nil, nil, nil, nil, nil, nil}
]
```

إضافة مسار GT ديناميكيًا

`add_gt_route/3` تأخذ البادئة و ID آخر كل شيء؛ موضعي النظير بشكل (`priority`, `description`, `source_ssn`, `dest_ssn`, المطابقات وإعادة الكتابة TT/NPI/NAI، ��`enabled`) هو خيار كلمة رئيسية.

```
# التوقيع: add_gt_route(gt_prefix, peer_id, opts \ [])

# إضافة مسار لأرقام ألمانية
M3UARouting.add_gt_route("49", 4, priority: 1, description: "أرقام
ألمانيا")

# إضافة مسار SMS يتطابق مع SSN 8 (SMSC) ويعيد كتابة SSN 6 إلى
(HLR)
M3UARouting.add_gt_route("55", 4,
  source_ssn: 8,
  dest_ssn: 6,
  priority: 1,
  description: "حركة SMS إلى HLR"
)
```

إزالة مسار GT

```
# إزالة المسار (gt_prefix, peer_id, source_ssn اختياري)
M3UARouting.remove_gt_route("49", 4)
```

اختبار بحث GT

```
# اختبار أي نظير سيتعامل مع هذا GT
# التوقيع: lookup_peer_by_gt(global_title, source_ssn \ nil, opts
\ [])
{:ok, peer, rewrites} =
M3UARouting.lookup_peer_by_gt("447700123456")

# `peer` هو سجل m3ua_peer عنصر 13 المكون من:
# {:m3ua_peer, peer_id, peer_name, protocol, role, local_ip,
local_port,
#  remote_ip, remote_port, routing_context, point_code,
network_indicator, status}
#
# `rewrites` هو خريطة من إعادة الكتابة للمؤشرات التي يجب تطبيقها
عند التوجيه.
# القيم nil تعني "احتفظ بالأصلي":
#   %{dest_ssn: nil, dest_tt: nil, dest_npi: nil, dest_nai: nil}
```

عندما يحدد مسار إعادة الكتابة، تظهر في الخريطة. على سبيل المثال، مسار `"55"` مع `dest_ssn: 6`:

```
{:ok, peer, rewrites} =
M3UARouting.lookup_peer_by_gt("5512345678", 8)
# rewrites => %{dest_ssn: 6, dest_tt: nil, dest_npi: nil,
dest_nai: nil}
```

مسار `peer_id: 0` (DROP) يعيد `{:error, :dropped}`؛ GT غير المطابق يعيد `{:error, :no_gt_route}`.

السجلات

يسجل STP قرارات توجيه GT:

```
[info] توجيه M3UA DATA: OPC=1000, DPC=200, GT
(1 من النظير) "447700123456"=المدعو
[info] التوجيه بواسطة GT: 447700123456
[info] 2 تم توجيه الرسالة من النظير 1 إلى النظير
```

إذا فشل توجيه GT:

```
[info] توجيه M3UA DATA: OPC=1000, DPC=200, GT "331234567"=المدعو
(1 من النظير)
[info] لم يتم العثور على مسار GT 331234567 لـ
[info] 2 تم توجيه الرسالة من النظير 1 إلى النظير (عبر توجيه PC)
```

دعم SCCP

رسائل SCCP المدعومة

الرسالة	الرمز	الدعم
UDT (Unitdata)	0x09	كامل: استخراج GT، إ��ادة كتابة المؤشر، التوجيه
UDTS (Unitdata Service)	0x0A	كامل: يتم إنشاؤه كرسالة عائدة عند فشل التوجيه
XUDT (Extended Unitdata)	0x11	كامل: استخراج GT، إعادة كتابة المؤشر، تقليل عداد القفزات، رسالة عائدة
XUDTS (Extended Unitdata Service)	0x12	كامل: يتم إنشاؤه كرسالة عائدة عند فشل التوجيه
LUDT (Long Unitdata)	0x13	كامل: استخراج GT، إعادة كتابة المؤشر، تقليل عداد القفزات، رسالة عائدة
LUDTS (Long Unitdata Service)	0x14	كامل: يتم إنشاؤه كرسالة عائدة عند فشل التوجيه

تحمل XUDT و LUDT عداد قفزات SCCP. يقوم STP بتقليله في كل قفزة عبور، وإذا وصل إلى الصفر، يتخلص من الرسالة ويعيدها إلى المرسل (عند طلب العودة عند الخطأ). راجع دليل STP للحصول على التفاصيل.

تنسيقات GT المدعومة

- **GTI 0-4**: جميع أنواع مؤشرات GT
- **ترميز BCD**: ترميز الأرقام الهاتفية القياسية
- **طبيعة العنوان**: جميع الأنواع (دولية، وطنية، إلخ.)

الأداء

- **بحث GT**: O(n) حيث n = عدد طرق GT
- **مطابقة البادئة**: محسّنة مع فرز المطابقة الأطول أولاً
- **التوصية**: الحفاظ على عدد طرق GT أقل من 1000 لأفضل أداء

استكشاف الأخطاء وإصلاحها

GT غير مستخرج

المشكلة: السجلات تظهر `Called GT=nil`

الحلول:

1. تحقق من نوع رسالة SCCP هو UDT/XUDT/LUDT
2. تحقق من أن عنوان الطرف المدعو يحمل عنوانًا عالميًا
3. تمكين تسجيل الأخطاء لرؤية تفاصيل تحليل SCCP

النظير المختار بشكل خاطئ

المشكلة: تم توجيه الرسالة إلى نظير خاطئ

الحلول:

1. تحقق من تكوين بادئة GT (الفوز بأطول مطابقة)
2. تحقق من أن `peer_id` يتطابق مع النظائر المكونة
3. تحقق من قيم الأولوية (الأقل = أولوية أعلى)

لا يعمل PC التراجع إلى

المشكلة: تم إسقاط الرسالة عند عدم العثور على GT

الحلول:

1. تحقق من أن `enable_gt_routing` هو `true`
2. تحقق من أن جدول توجيه DPC يحتوي على مسار لـ PC

3. تأكد من أن النظير لـ PC نشط

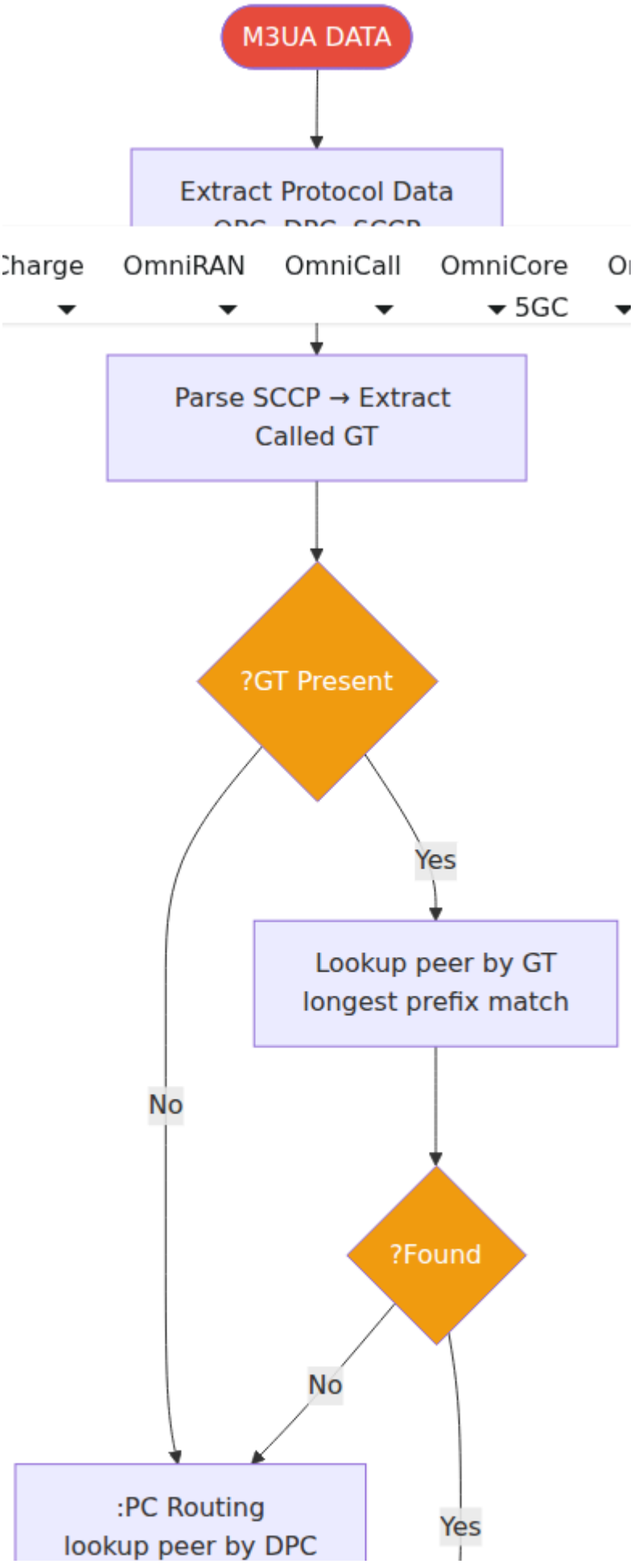
أفضل الممارسات

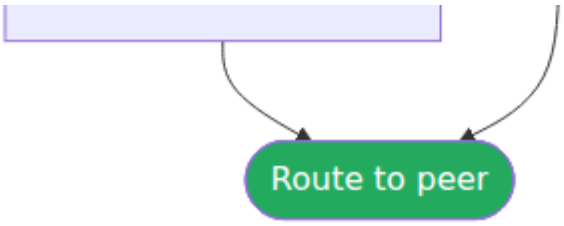
1. **استخدم بادئات محددة**: الطرق الأكثر تحديدًا تتجاوز الأقل تحديدًا
   - جيد: "447"، "448"، "449" لمشغلي المملكة المتحدة المختلفين
   - سيء: "44" واحدة لجميع المملكة المتحدة
2. **وثق الطرق**: استخدم حقل الوصف

```
M3UARouting.add_gt_route("1234", 1, priority: 1,
  description: "شركة ACME - الساحل الشرقي للولايات المتحدة")
```

3. **اختبر الطرق**: استخدم `lookup_peer_by_gt/1` للتحقق
4. **راقب**: تحقق من السجلات لفشل التوجيه
5. **التراجع**: قم دائمًا بتكوين طرق PC كنسخة احتياطية

الهيكلية


M3UA DATA
Extract Protocol Data
Charge
OmniRAN
OmniCall
OmniCore
5GC
Parse SCCP → Extract Called GT
?GT Present
Yes
No
Lookup peer by GT longest prefix match
?Found
No
Yes
:PC Routing lookup peer by DPC



OmniSS7 بواسطة Omnitouch Network Services