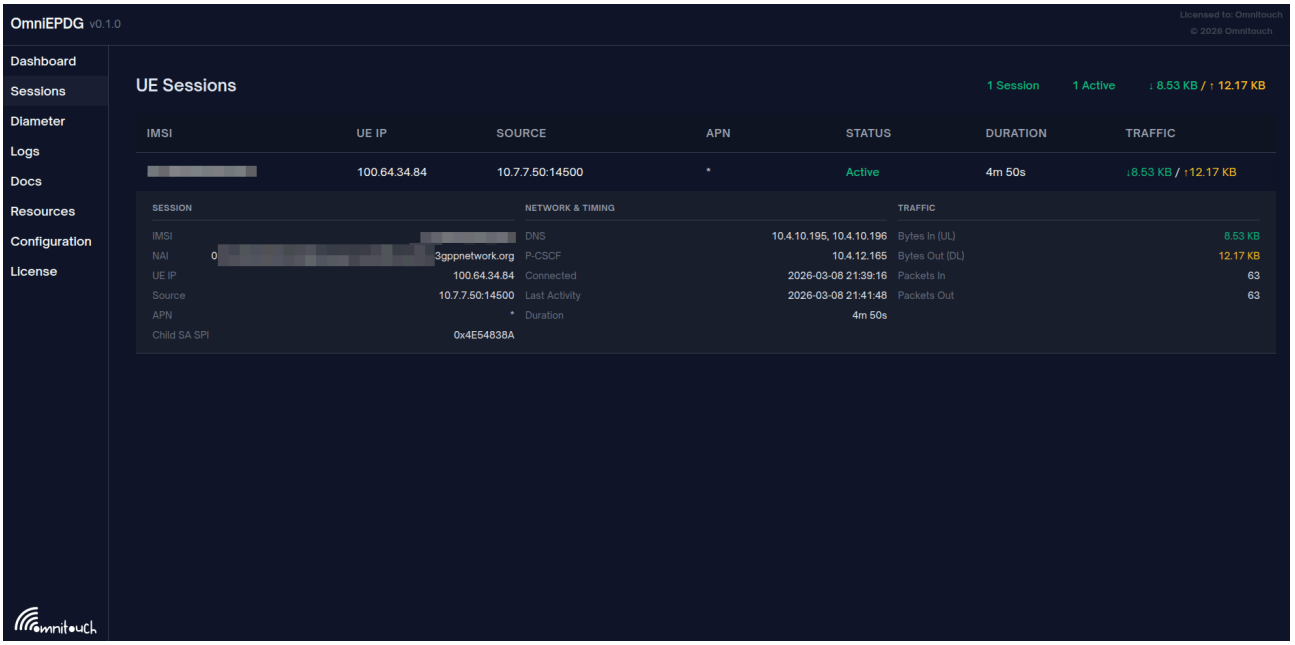


OmniEPDG □作指南

OmniEPDG □一个演进的数据包数据网关 (ePDG)，支持 WiFi 上的语音通话 (VoWiFi)。它通过 EAP-AKA 在不受信任的 WiFi 网络上对移动用户进行身份验证，并通过 Diameter 信令连接到移动核心网络，连接到 HSS 和 GTP 隧道到数据包网关 (PGW)。

OmniEPDG □□□□□□□□□□□□□□□□□□□□□□□□

OmniEPDG 支持两种操作模式:

- **GTP** 模式（默认） - 通过 GTPv2-C 和 GTP-U 通过 PGW 完全符合 3GPP 标准的隧道
- 简单 **VPN** 模式 - 本地突破，内置 IP 池和 Linux TUN 接口，无需 PGW

OmniEPDG 还可以作为 5G **N3IWF** 在同一 IKEv2 监听器上运行。请参见 N3IWF 模式。

文档

配置与□作

- 架构与呼叫流程 - 系统架构、协议接口、用户设备状态机和两种模式的消息序列图
- 配置参考 - Diameter、GTPv2-C、GTP-U、简单 VPN 和日志的完整参数文档

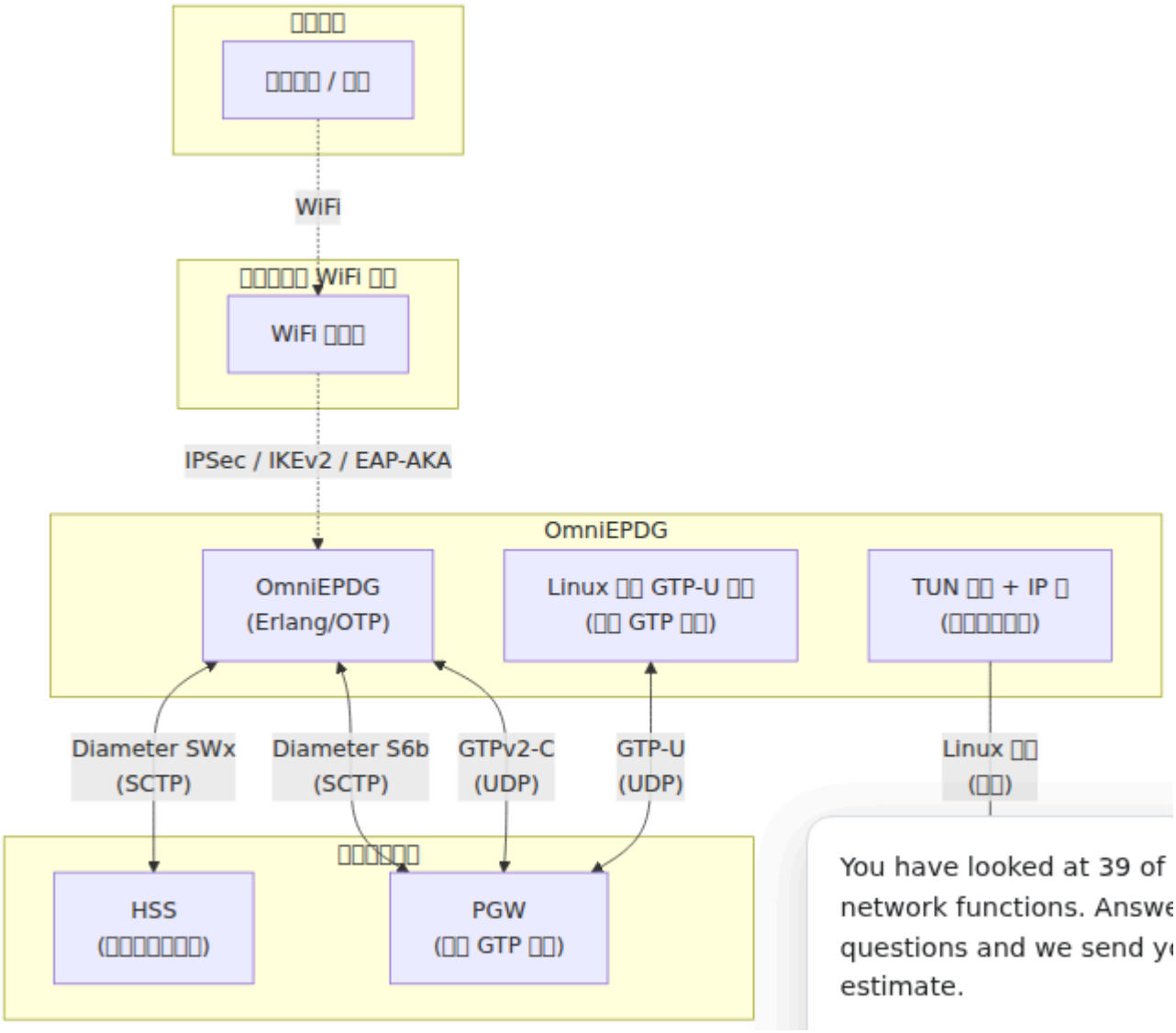
- 控制面板 - 用于会话、Diameter 对等体和日志的基于 Web 的监控 UI
- **N3IWF** 模式 - 5G 不受信任的非 3GPP 访问：NWu 上的 EAP-5G、到 AMF 的 NGAP 和到 UPF 的 GTP-U

安全

- 安全指南 - 身份验证速率限制和 GeoIP 国家阻止

监控与故障排除

- 指标参考 - 用于监控身份验证、会话、Diameter 信令和系统健康的 Prometheus 指标
- 故障排除 - 常见问题、诊断程序和解决步骤



□作模式

GTP 模式

默认模式。OmniEPDG 通过 PGW 使用 GTPv2-C 隧道所有用户流量进行会话控制，并通过 Linux 内核模块使用 GTP-U 进行用户面。这完全符合 3GPP 标准，适合具有现有 EPC 基础设施的运营商部署。

流量路径： UE → IPSec → OmniEPDG → GTPv2-C 创建会话 → GTP-U 隧道 → PGW → 互联网

所需基础设施： HSS, PGW

简单 **VPN** 模式

OmniEPDG 从本地池分配 IP 地址，并通过 Linux TUN 接口 (`tun_epdg`) 直接路由用户流量，使用标准内核路由。无需 PGW 或 GTP 基础设施。身份验证仍然通过 Diameter SWx 到 HSS 进行。

流量路径： UE → IPSec → OmniEPDG → 本地 IP 分配 → TUN 接口 → Linux 路由 → 互联网

所需基础设施：仅 HSS（无需 PGW）

可选优化： `skip_sar` 标志绕过 HSS 服务器分配请求/应答，减少连接设置时间。这意味着 HSS 不会跟踪哪个 ePDG 为用户服务，HSS 发起的程序（注销、配置文件推送）将无法正常工作。适合没有漫游要求的私人部署。

模式比较

功能	**GTP** 模式	简单 **VPN** 模式
符合 3GPP 标准	✅	否（使用 `skip_sar`），部分（不使用）
需要 PGW	✅	否
需要 HSS	✅	✅（仅身份验证）
IP 分配	来自 PGW	本地池（CIDR）
用户面	GTP-U 内核模块	Linux TUN + 路由
HSS 配置文件推送	✅（PPR/PPA）	否
HSS 注销	✅（RTR/RTA）	否（使用 `skip_sar`）
PGW 发起的拆除	✅	不适用
漫游支持	✅	否
IPv6 / 双栈	✅	仅 IPv4

协议接口

接口	协议	传输	模式	对等体	目的	参考
SWu	IKEv2 / IPSec	UDP	两者	UE	安全隧道和 EAP-AKA 身份验证	3GPP TS 33.402
SWx	Diameter	SCTP	两者	HSS	身份验证向量和服务器分配	3GPP TS 29.273 第 8 节
S6b	Diameter	SCTP	仅限 GTP	PGW	会话授权和策略	3GPP TS 29.273 第 9 节
S2b	GTPv2-C / GTP-U	UDP	仅限 GTP	PGW	控制和用户面隧道	3GPP TS 29.274

功能

核心功能

- **EAP-AKA** 身份验证 - 通过 HSS 完全符合 3GPP 的 EAP-AKA 用户身份验证
- **IPSec** 隧道管理 - UE 和 ePDG 之间基于 IKEv2 的安全隧道
- 双□作模式 - GTP 隧道到 PGW 或使用简单 VPN 的本地突破
- 每个 **UE** 状态机 - 每个用户的独立 Erlang FSM，用于会话生命周期管理
- 双栈支持 - IPv4、IPv6 和 IPv4v6 PDP 地址类型（GTP 模式）

GTP 模式功能

- **GTP** 隧道建立 - 通过 Linux 内核模块创建 GTPv2-C 会话和 GTP-U 用户面
- **PGW** 发起的拆除 - PGW 发送删除承载请求，ePDG 将拆除传递给 UE
- 网络发起的拆除 - HSS 通过 SWx RTR 触发注销，ePDG 拆除所有会话
- 重新身份验证 - HSS 触发的配置文件推送和重新授权，参见 3GPP TS 29.273 第 7.1.2.5.1 节

简单 **VPN** 模式功能

- 本地 **IP** 池 - 基于 CIDR 的 IPv4 地址分配，按 IMSI 跟踪
- **TUN** 接口路由 - 标准 Linux TUN 设备 (`tun_epdg`)，每个 UE 主机路由
- **DNS** 配置 - 通过 PCO 提供给 UE 的可配置 DNS 服务器
- 可选的 **SAR** 跳过 - 绕过 HSS 注册以加快连接设置

安全功能

- 身份验证速率限制 - 每个 IP 和每个 IMSI 的暴力破解保护，具有可配置的阈值
- **GeoIP** 国家阻止 - ���用 MaxMind GeoLite2 的基于国家的访问控制白名单或黑名单
- 死对等体检测 - 具有可配置探测的主动存活监控
- **ESP** 防重放保护 - 符合 RFC 4303 的 64 位滑动窗口

HSS 集成 **(SWx Diameter)**

- 多媒体认证请求/应答 **(MAR/MAA)** - 检索 EAP-AKA 身份验证向量（两种模式）
- 服务器分配请求/应答 **(SAR/SAA)** - 下载用户配置文件和 APN 配置（在简单模式中可跳过）
- 推送配置文件请求/应答 **(PPR/PPA)** - 从 HSS 接收更新的用户配置文件（GTP 模式）
- 注册终止请求/应答 **(RTR/RTA)** - HSS 发起的用户注销（GTP 模式）

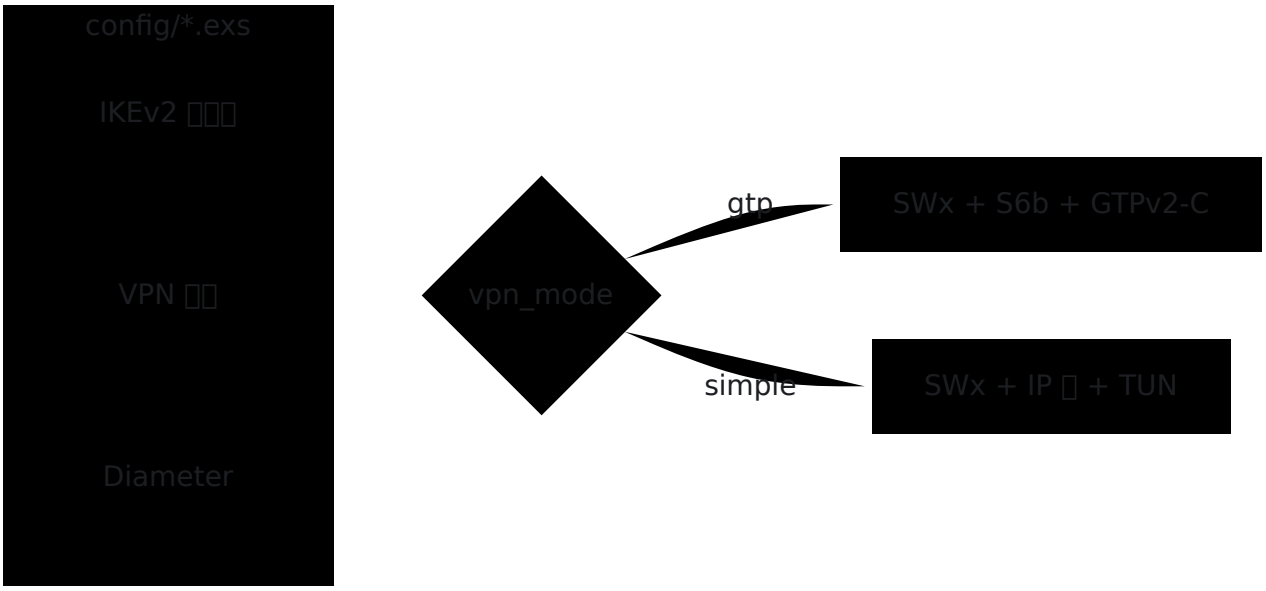
PGW 集成（仅限 **GTP** 模式）

S6b Diameter:

- **AA** 请求/应答 **(AAR/AAA)** - 授权 PGW 会话
- 会话终止请求/应答 **(STR/STA)** - 终止 PGW 会话
- 重新认证请求/应答 **(RAR/RAA)** - 重新授权活动会话
- 强制会话终止请求/应答 **(ASR/ASA)** - 强制终止会话

S2b GTPv2-C:

- 创建会话请求/响应 - 建立带有 TEID 分配的 GTP 隧道
- 删除会话请求/响应 - 拆除 GTP 隧道
- 删除承载请求/响应 - PGW 发起的承载管理



快速开始

配置结构

配置在 `config/runtime.exs` 中完成或通过环境变量进行。`vpn_mode` 参数选择 GTP 和简单 VPN 模式。请参见 配置参考 获取完整的参数文档。

典型网络地址分配（GTP 模式）

组件	IP 地址	端口	备注
OmniEPDG (GTP-U)	10.74.0.11	-	GTP-U 隧道端点
OmniEPDG (Diameter S6b)	10.74.0.12	3868	S6b Diameter 监听器
HSS	10.74.0.21	3868	SWx Diameter 对等体
PGW	10.74.0.23	2123	GTPv2-C 和 S6b 对等体

典型网络地址分配（简单 **VPN** 模式）

组件	**IP** 地址	备注
OmniEPDG (TUN 网关)	10.44.0.1	`tun_epdg` 接口上的网关 IP
UE IP 池	10.45.0.0/16	可配置的 CIDR 池，用于用户 IP
HSS	10.74.0.21:3868	SWx Diameter 对等体（仅身份验证）

3GPP 规范

规范	标题	相关性
TS 29.273	EPS AAA 接口 (SWx, S6b, SWm)	ePDG Diameter 接口的主要规范
TS 29.274	GTPv2-C 和 GTP-U	S2b 隧道控制和用户面（GTP 模式）
TS 33.402	非 3GPP 访问的安全性	不受信任的 WiFi 上的 EAP-AKA 身份验证
TS 23.402	非 3GPP 访问的架构增强	整体 ePDG 架构和程序
TS 23.003	编号、地址和标识	NAI 格式，IMSI 结构
TS 29.229	Cx/Dx Diameter（通用定义）	SWx 使用的服务器分配类型值
RFC 6733	Diameter 基础协议	Diameter 传输、对等体管理、看门狗
RFC 4187	EAP-AKA	通过 IKEv2 使用的身份验证方法

按角色文档

网络运营商：

1. 从 架构与呼叫流程 开始，以了解系统和两种操作模式
2. 查看 配置参考 以获取部署参数
3. 查看 安全指南 以配置速率限制和 GeoIP 阻止
4. 使用 指标参考 设置监控以集成 Prometheus
5. 保留 故障排除 指南以备操作使用

系统集成商:

1. 查看 架构与呼叫流程 以获取接口详细信息和状态机
2. 使用 配置参考 进行对等体连接设置
3. 使用 指标参考 配置警报
4. 参考上面的 3GPP 规范表以确保协议合规性

OmniEPDG 操作指南

OmniEPDG 是一个符合 3GPP 标准的 ePDG（演进数据包网关），通过将不受信任的 WiFi 接入与移动核心网络连接，支持 WiFi 通话。它支持两种操作模式：**GTP** 模式用于 PGW 隧道和 简单 **VPN** 模式用于本地 IP 突破。

快速链接

配置与部署

- 网络要求 - 部署所需的防火墙端口和 DNS 条目
- 配置参考 - IKEv2、Diameter、VPN 模式及所有运行时设置的完整参数文档
- 架构概述 - 系统架构、呼叫流程、状态机和协议参考

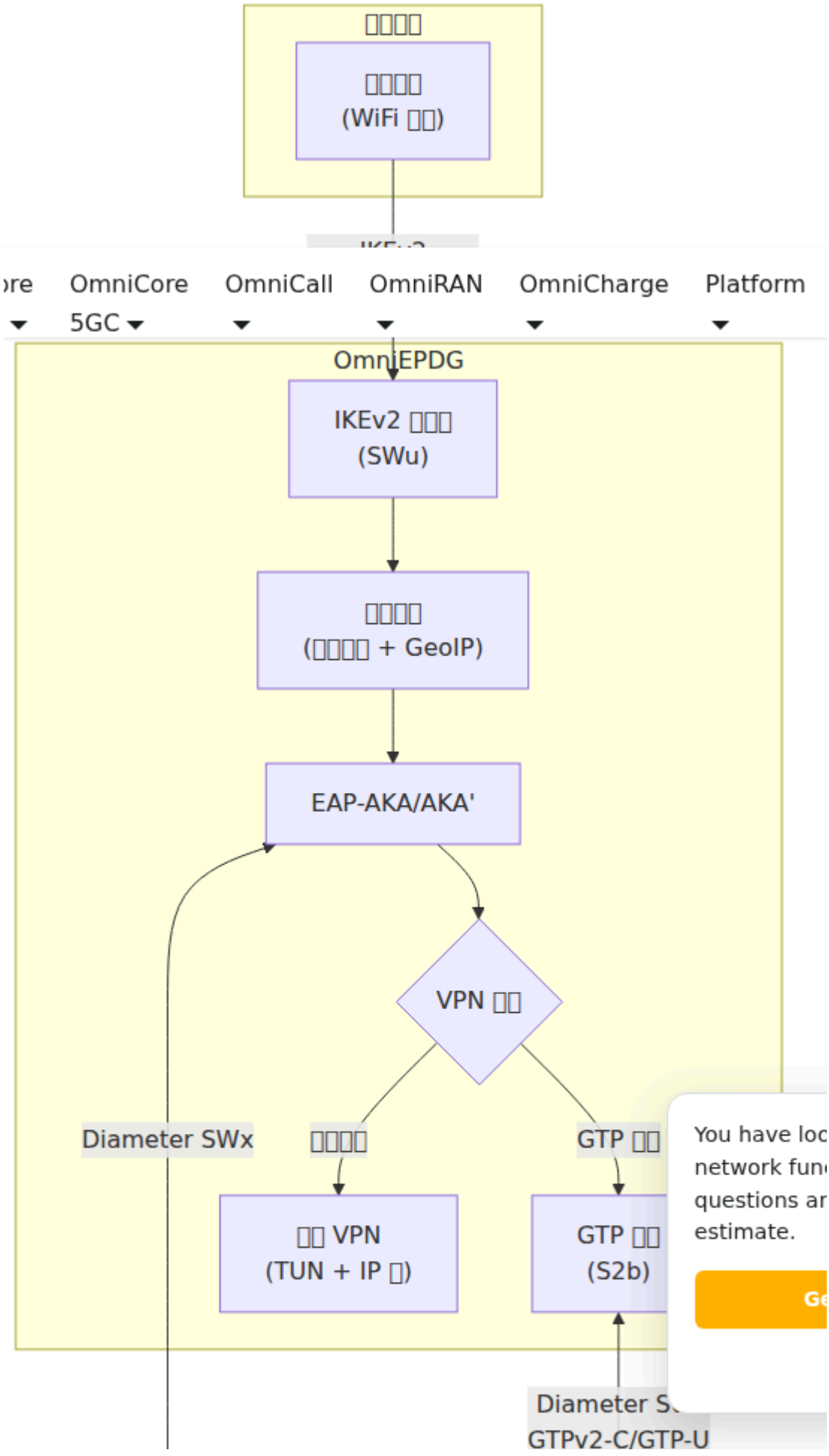
操作与监控

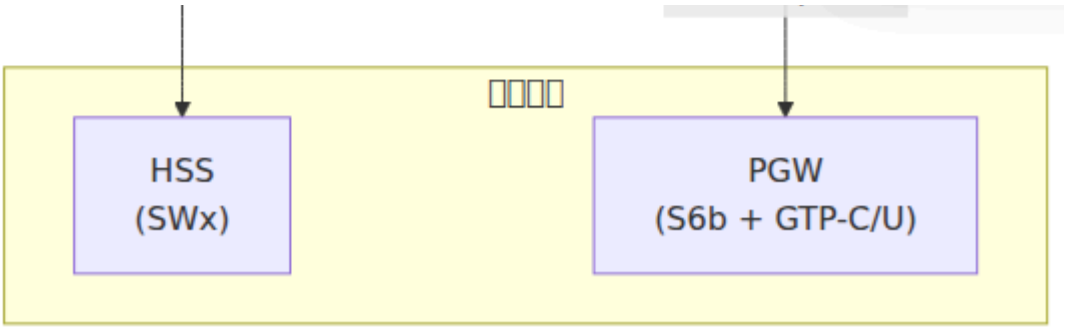
- 控制面板 - 用于会话、Diameter 对等体和日志的基于 Web 的监控界面
- 指标参考 - Prometheus 指标、示例查询和警报规则
- **Webhooks** - 附加和分离时的出站会话生命周期通知
- 故障排除 - 常见问题、诊断程序和解决步骤

安全

- 认证安全 - 速率限制和 GeoIP 国家阻止

□□□□


□□□□
□□□□
(WiFi □□)
OmniCore
5GC
OmniCall
OmniRAN
OmniCharge
Platform
OmniEPDG
IKEv2 □□□
(SWu)
□□□□
(□□□□ + GeoIP)
EAP-AKA/AKA'
VPN □□
Diameter SWx
□□□□
GTP □□
□□ VPN
(TUN + IP □)
GTP □□
(S2b)
You have loc
network fun
questions ar
estimate.
Ge
Diameter S
GTPv2-C/GTP-U



作模式

OmniEPDG 支持通过 `vpn_mode` 配置参数选择的两种操作模式。

GTP 模式

通过 PGW 完成完整的 3GPP 隧道。用户流量被封装在 GTP-U 中，并通过移动核心路由。

使用场景：

- 与现有移动核心基础设施集成
- 通过 PCRF/PCEF 进行策略和计费
- 需要漫游和运营商间切换

组件：

- 用于 PGW 会话授权的 Diameter S6b
- 用于会话管理的 GTPv2-C（创建/删除/修改会话）
- 用于用户平面的 Linux 内核 GTP-U 模块

简单 VPN 模式

通过 TUN 接口实现本地 IP 突破。用户流量直接通过 OmniEPDG 主机路由，无需 PGW 参与。

使用场景：

- 无需 PGW 的独立部署
- 测试和开发
- 本地突破场景

组件：

- 本地 IP 池管理（IPv4/IPv6）
- 带有每个用户主机路由的 TUN 接口（`omniepdg0`）
- 可选的 NAT/伪装以访问互联网

认证安全

OmniEPDG 包含内置安全功能，以防止暴力攻击并限制地理位置访问。有关详细信息，请参见 认证安全 指南。

速率限制

通过跟踪失败的认证尝试来防止暴力攻击：

- 每个 **IP** 限制 - 在 1 分钟内失败 10 次后阻止 IP（阻止 5 分钟）
- 每个 **IMSI** 限制 - 在 1 分钟内失败 5 次后阻止 IMSI（阻止 10 分钟）
- 滑动窗口算法，自动过期
- 成功认证清除失败历史

GeoIP 国家阻止

使用 MaxMind GeoLite2 数据库的可选地理访问控制：

- 白名单模式 - 仅允许来自指定国家的连接
- 黑名单模式 - 阻止来自指定国家的连接
- 可配置未知/私有 IP 的处理
- 数据库不可用时的失败开放或失败关闭行为

关键配置

最小配置（简单 VPN 模式）

```
config :omniepdg,
  vpn_mode: :simple,
  simple_vpn: [
    pool_ipv4: "10.45.0.0/16",
    dns_servers_ipv4: ["8.8.8.8", "8.8.4.4"]
  ]

config :diameter_ex,
  diameter: %{
    host: "epdg",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    peers: [
      %{host: "hss", ip: "10.74.0.21", port: 3868, transport:
:tcp}
    ]
  }
```

启用认证安全

```
config :omniepdg,
  # 率限制（默认启用这些值）
  auth_rate_limit_per_ip: 10,
  auth_rate_limit_ip_block_ms: 300_000,
  auth_rate_limit_per_imsi: 5,
  auth_rate_limit_imsi_block_ms: 600_000,

  # GeoIP 阻止（默认禁用）
  geoip_enabled: true,
  geoip_mode: :whitelist,
  geoip_countries: ["AU", "NZ"]
```

有关完整参数文档，请参见 配置参考。

监控

控制面板

访问 Web 控制面板 `http://<host>:4000/dashboard` 以获取：

- 实时会话监控
- Diameter 对等体状态
- 实时日志流
- 系统配置

有关详细信息，请参见 控制面板 指南。

Prometheus 指标

从 `http://<host>:9568/metrics` 抓取指标以获取：

- 认证成功/失败率
- 会话生命周期事件
- Diameter 信令延迟
- 安全事件（速率限制，GeoIP 阻止���
- IP 池利用率
- ESP 数据平面统计

有关查询和警报规则，请参见 指标参考。

故障排除

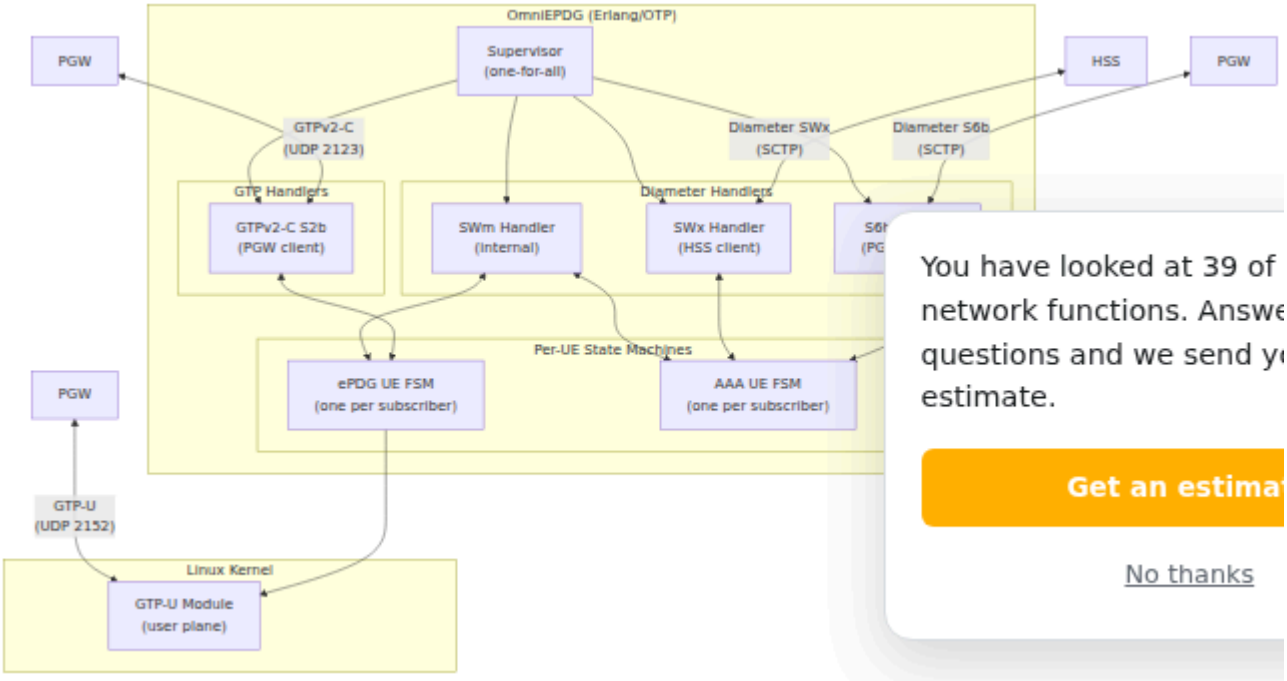
常见问题及解决步骤：

问题	快速检查	指南部分
认证失败	检查日志中的 SWx MAR/MAA	认证失败
Diameter 连接问题	检查控制面板中的对等体状态	Diameter 连接性
GTP 隧道失败	检查 GTPv2-C 原因代码	GTP 隧道失败
简单 VPN 问题	检查 TUN 接口和路由	简单 VPN 失败
速率限制误报	调整阈值	速率限制问题
GeoIP 阻止问题	检查数据库和国家代码	GeoIP 问题

有关详细诊断程序，请参见 故障排除 指南。

文档索引

文档	描述
架构	系统设计、状态机、呼叫流程、协议参考
配置	带有示例的完整配置参考
控制面板	带有屏幕截图的 Web ��面指南
指标	Prometheus 指标、查询和警报
网络要求	部署所需的防火墙端口和 DNS 条目
安全	速率限制和 GeoIP 国家阻止
故障排除	常见问题和诊断程序



OmniEPDG 架构与呼叫流程

本文档描述了 OmniEPDG 的内部架构、协议接口、UE 状态机以及关键过程的详细消息序列图。OmniEPDG 支持两种操作模式：**GTP** 模式（通过 PGW 完整 3GPP 隧道）和 简单 **VPN** 模式（通过 TUN 接口的本地突破）。有关高层次比较，请参见 操作指南。

系统架构

OmniEPDG 基于 Erlang/OTP 构建，并实现了 3GPP ePDG（演进数据包网关）网络功能。它将不可信的 WiFi 接入与移动核心网络连接起来，使 UE 能够进行和接听 VoWiFi 通话。

GTP 模式架构

在 GTP 模式下，用户流量通过 PGW 使用 GTPv2-C 进行会话控制，并使用 Linux 内核 GTP-U 模块进行用户平面隧道。

简单 **VPN** 模式架构

在简单 VPN 模式下，用户流量通过 Linux TUN 接口在本地路由。无需 PGW 或 GTP 基础设施。Diameter S6b、GTPv2-C 和 GTP-U 组件被简单 VPN 子系统替代。

监控树

OmniEPDG 使用 一对多 的监控策略，这意味着如果任何子进程崩溃，所有子进程都会被重启。监控器根据操作模式有条件地启动不同的子进程。

在两种模式下启动的进程：

进程	角色	描述
`aaa_diameter_swx`	Diameter SWx 客户端	连接到 HSS 进行身份验证和用户配置文件操作
`aaa_diameter_swm`	Diameter SWm（内部）	在 ePDG 和 AAA FSM 之间路由 Diameter EAP 和会话消息
`epdg_diameter_swm`	SWm ePDG 处理程序	处理内部 SWm Diameter 信令的 ePDG 侧

GTP 模式附加进程：

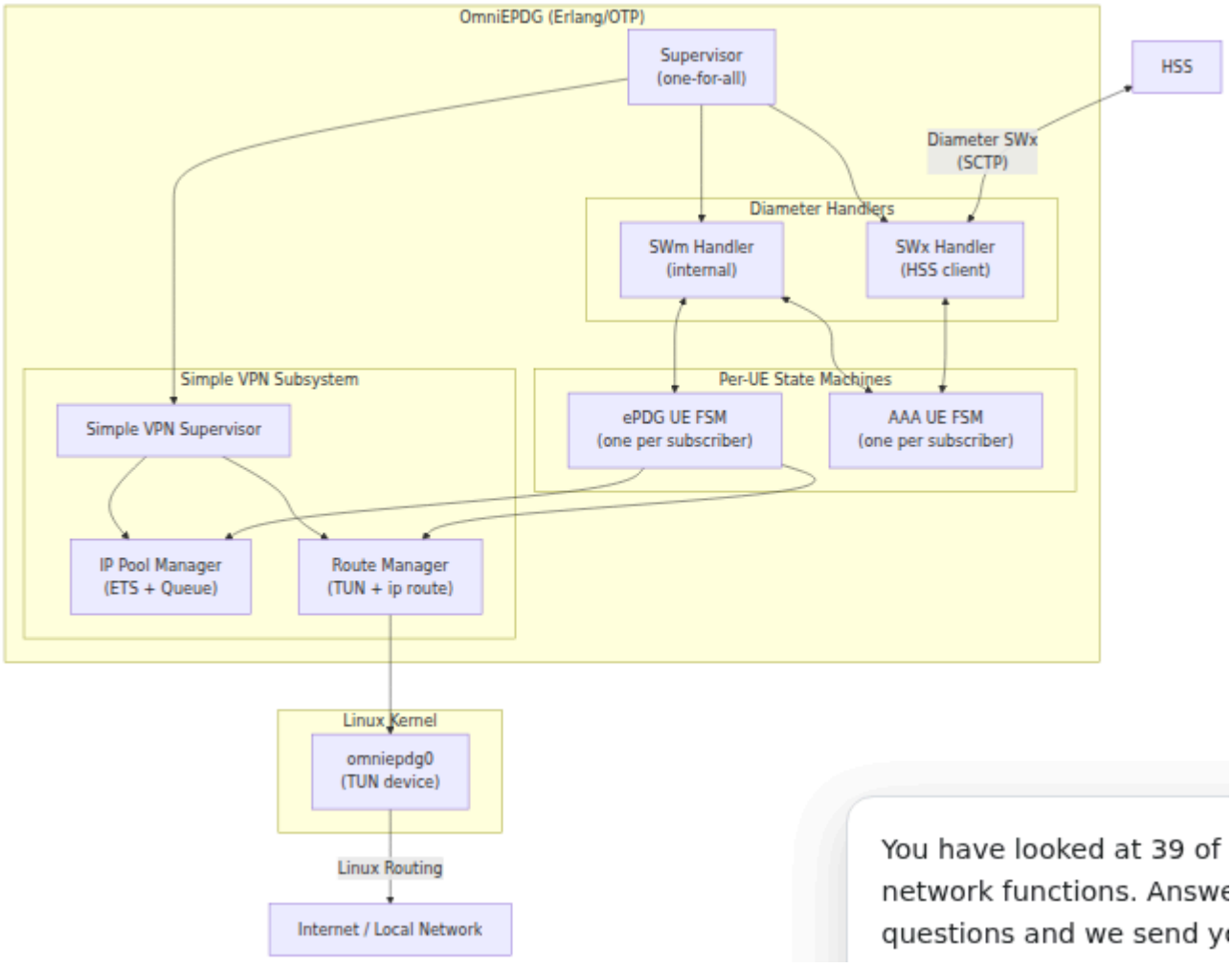
进程	角色	描述
`aaa_diameter_s6b`	Diameter S6b 服务器	接受来自 PGW 的会话授权连接
`epdg_gtpc_s2b`	GTPv2-C 客户端	通过 S2b 向 PGW 发送创建/删除会话请求
`gtp_u_kmod`	GTP-U 内核处理程序	管理 Linux 内核模块中的 GTP-U PDP 上下文

简单 **VPN** 模式附加进程：

进程	角色	描述
`simple_vpn_supervisor`	VPN 子系统监控器	监控 IP 池和路由管理器进程
`simple_vpn_pool`	IP 池管理器	使用 ETS 从配置的 CIDR 池中分配和释放 IPv4 地址
`simple_vpn_route`	路由管理器	创建 `omniepdg0` TUN 接口并管理每个用户的主机路由

每个 **UE** 状态机

对于每个活动的用户（通过 IMSI 识别），OmniEPDG 创建两个状态机实例：



- **ePDG UE FSM** (`epdg_ue_fsm`) - 从 ePDG 的角度管理用户的会话生命周期：身份验证、GTP 隧道创建和拆除协调
- **AAA UE FSM** (`aaa_ue_fsm`) - 管理 AAA 侧信令：与 HSS 的 Diameter SWx 交换和与 PGW 的 S6b 交换

这两个 FSM 都实现为 Erlang `gen_statem` 进程，具有状态函数回调模式。

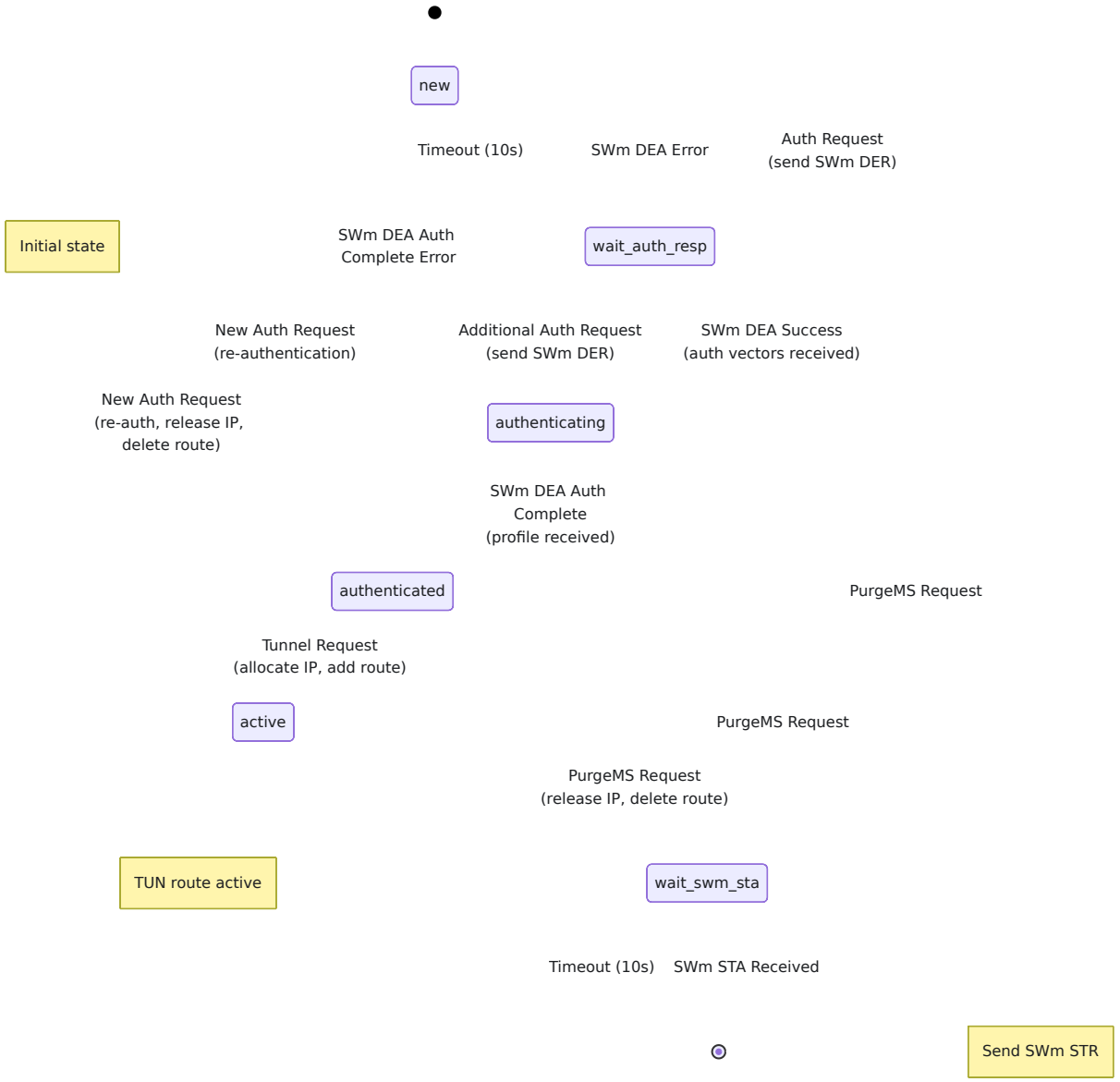
ePDG UE FSM 状态

ePDG UE FSM 跟踪用户的会话，从初始身份验证请求到活动隧道状态再到拆除。FSM 的行为在 `authenticated` 状态下根据操作模式有所不同。

GTP 模式 FSM

在 GTP 模式下，隧道建立通过 GTPv2-C 创建会话到 PGW，拆除涉及 GTPv2-C 删除会话、PGW 发起的删除承载和 HSS 发起的注销流程。

You have looked at 39 of
network functions. Answe
questions and we send y

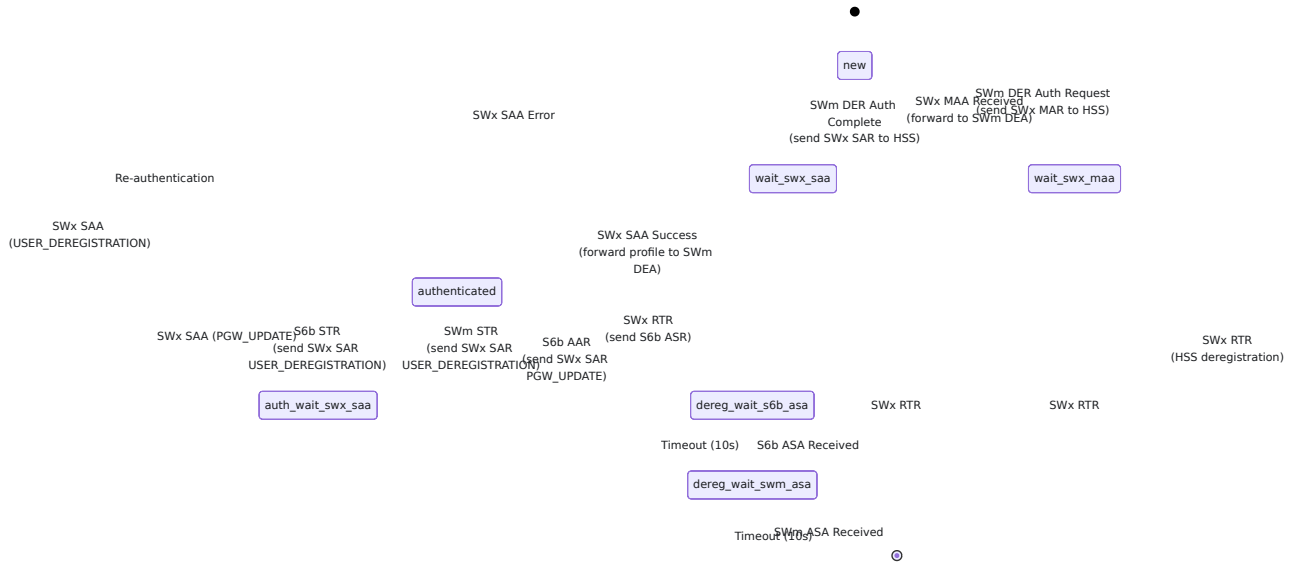


简单 VPN 模式 FSM

在简单 VPN 模式下，FSM 在 `authenticated` 状态下采取了捷径。FSM 从本地池分配一个 IP 地址，创建 TUN 接口上的主机路由，并直接过渡到 `active`。不使用特定于 GTP 的拆除状态（`wait_create_session_resp`、`wait_delete_session_resp`、`dereg_pgw_wait_cancel`、`dereg_net_wait_s2b_delete`）。

ePDG UE FSM 状态参考

状态	模式	描述	等待
`new`	两者	初始状态。没有活动会话。	UE 的身份验证请求
`wait_auth_resp`	两者	通过 SWm DER 发送身份验证请求。	SWm DEA 返回身份验证向量或错误
`authenticating`	两者	收到身份验证向量，EAP 交换进行中。	位置更新/身份验证完成
`authenticated`	两者	身份验证完成，用户配置文件已下载。	UE 的隧道请求
`wait_create_session_resp`	GTP	GTPv2-C 创建会话请求已发送到 PGW。	PGW 的创建会话响应
`active`	两者	隧道/路由正常工作。用户流量正在流动。	拆除触发
`wait_delete_session_resp`	GTP	GTPv2-C 删除会话请求已发送到 PGW（用户发起的拆除）。	PGW 的删除会话 ��应
`wait_swm_sta`	两者	发送 SWm 会话终止请求。	来自 AAA 的 SWm STA
`dereg_pgw_wait_cancel`	GTP	PGW 发起的注销。取消位置已发送到 UE。	取消位置结果
`dereg_net_wait_cancel`	GTP	网络/HSS 发起的注销。取消位置已发送到 UE。	取消位置结果
`dereg_net_wait_s2b_delete`	GTP	网络发起的注销。S2b 删除会话已发送到 PGW。	删除会话响应



AAA UE FSM 状态

AAA UE FSM 代表每个用户管理与 HSS（SWx）和 PGW（S6b）之间的 Diameter 信令。

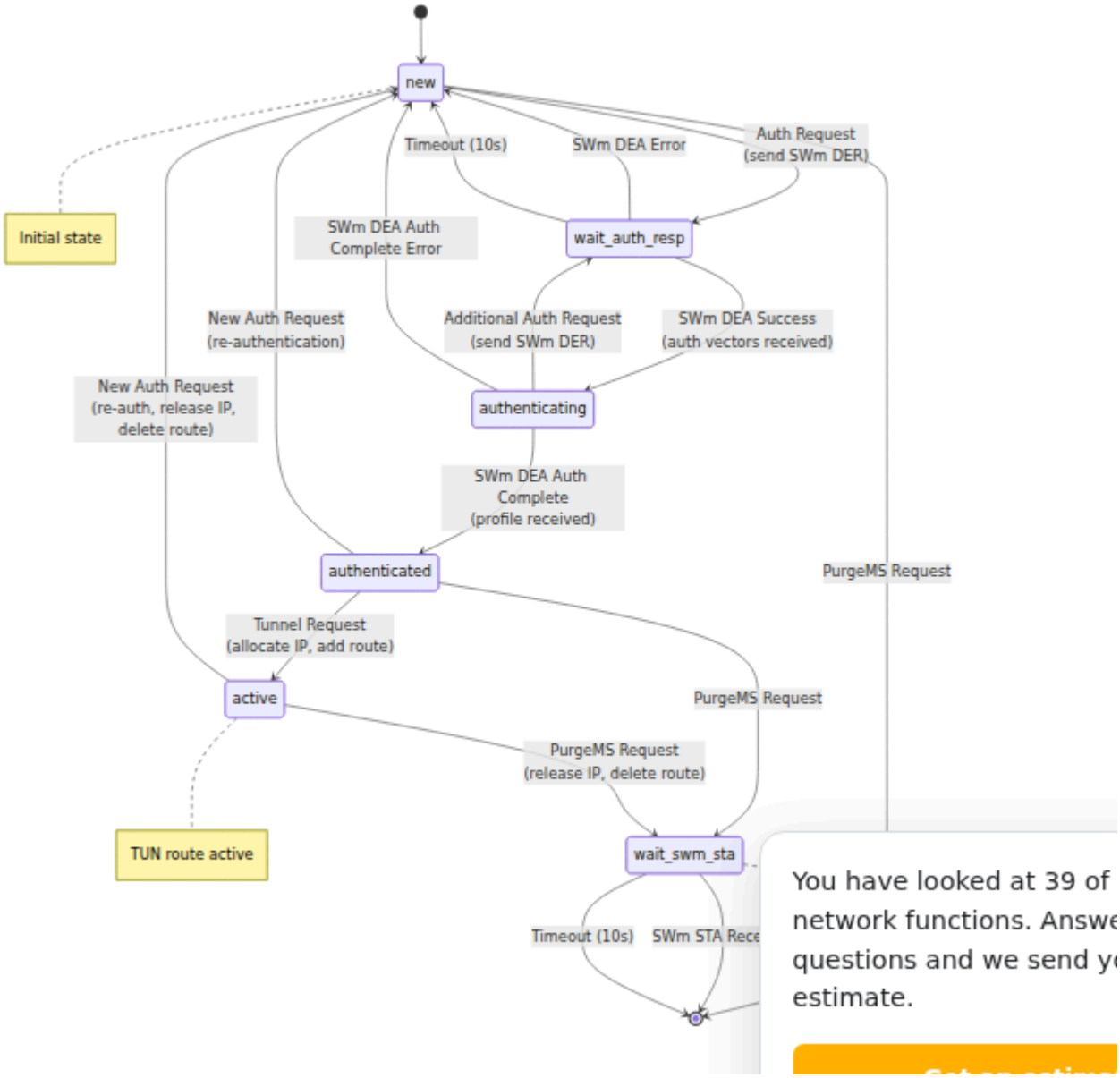
AAA UE FSM 状态参考

状态	描述	等待
`new`	初始状态。没有活动的 AAA 会话。	Diameter 身份验证请求
`wait_swx_maa`	SWx MAR 发送到 HSS 以获取 EAP-AKA 向量。	来自 HSS 的 SWx MAA
`wait_swx_saa`	SWx SAR 发送到 HSS 以获取服务器分配。	来自 HSS 的 SWx SAA
`authenticated`	ePDG 和 PGW 会话可能都处于活动状态。跟踪双重会话状态。	会话事件
`auth_wait_swx_saa`	SWx SAR 发送以进行 PGW 更新或用户注销。	来自 HSS 的 SWx SAA
`dereg_net_wait_s6b_asa`	HSS 发起的注销。S6b ASR 发送到 PGW。	来自 PGW 的 S6b ASA
`dereg_net_wait_swm_asa`	S6b 拆除完成。SWm ASR 发送到 ePDG。	来自 ePDG 的 SWm ASA

呼叫流程

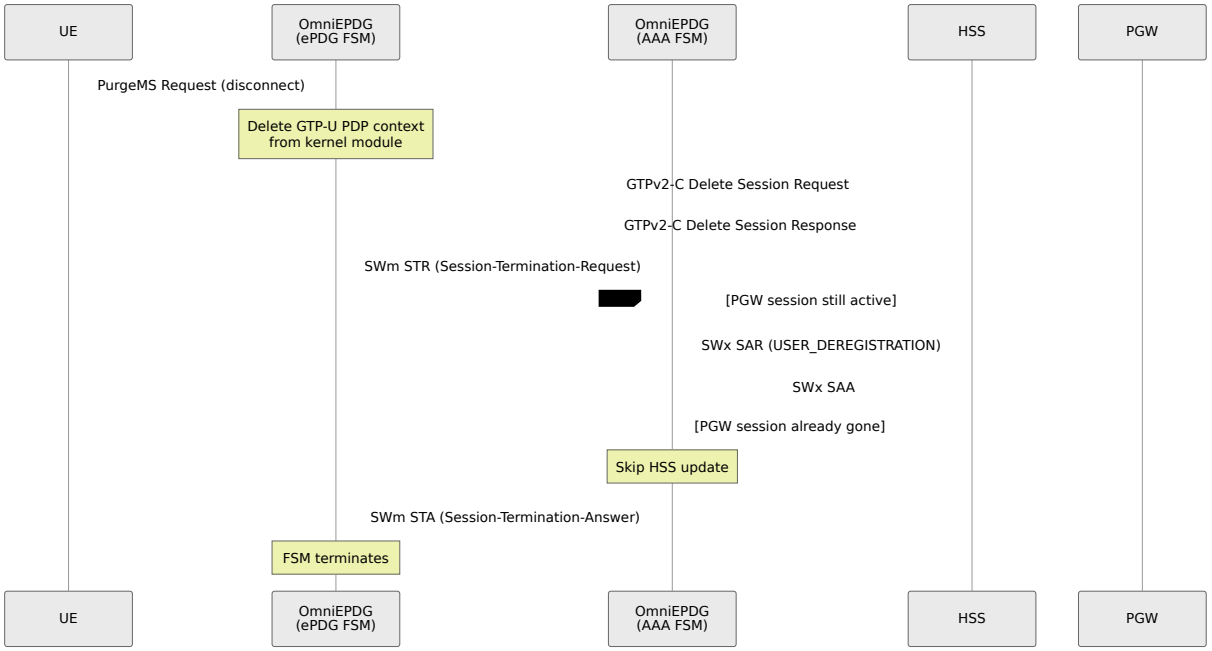
GTP 模式：成功的会话建立

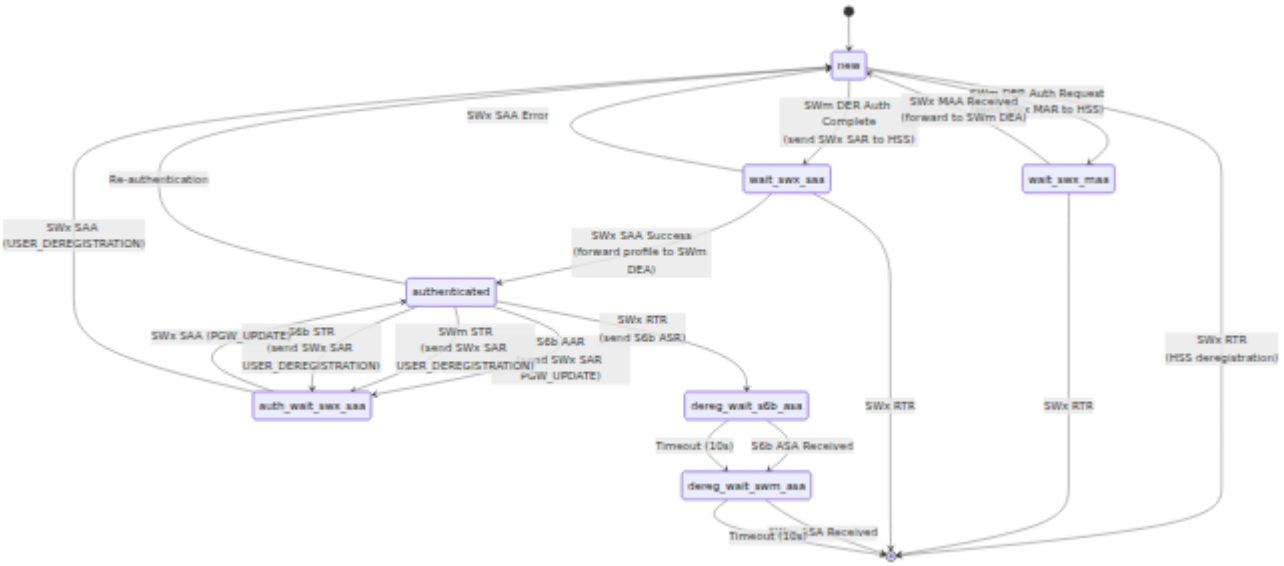
该序列显示了从 EAP-AKA 身份验证到活动 GTP 隧道的完整成功会话。


new
Initial state
Timeout (10s)
SWm DEA Error
Auth Request
(send SWm DER)
SWm DEA Auth
Complete Error
wait_auth_resp
New Auth Request
(re-authentication)
Additional Auth Request
(send SWm DER)
SWm DEA Success
(auth vectors received)
New Auth Request
(re-auth, release IP,
delete route)
authenticating
SWm DEA Auth
Complete
(profile received)
authenticated
PurgeMS Request
Tunnel Request
(allocate IP, add route)
active
PurgeMS Request
PurgeMS Request
(release IP, delete route)
TUN route active
wait_swm_sta
Timeout (10s)
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

GTP □□□□□□□□□□□□□

□ UE □□□□□□□□□□ WiFi □□□□□□□□□□□□



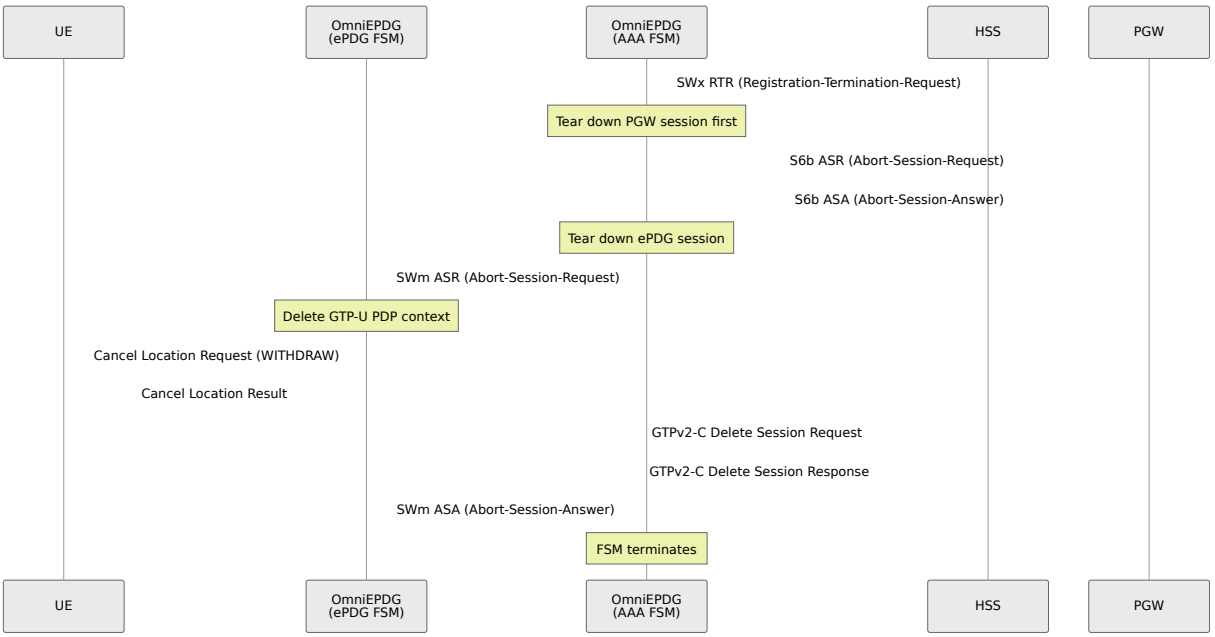


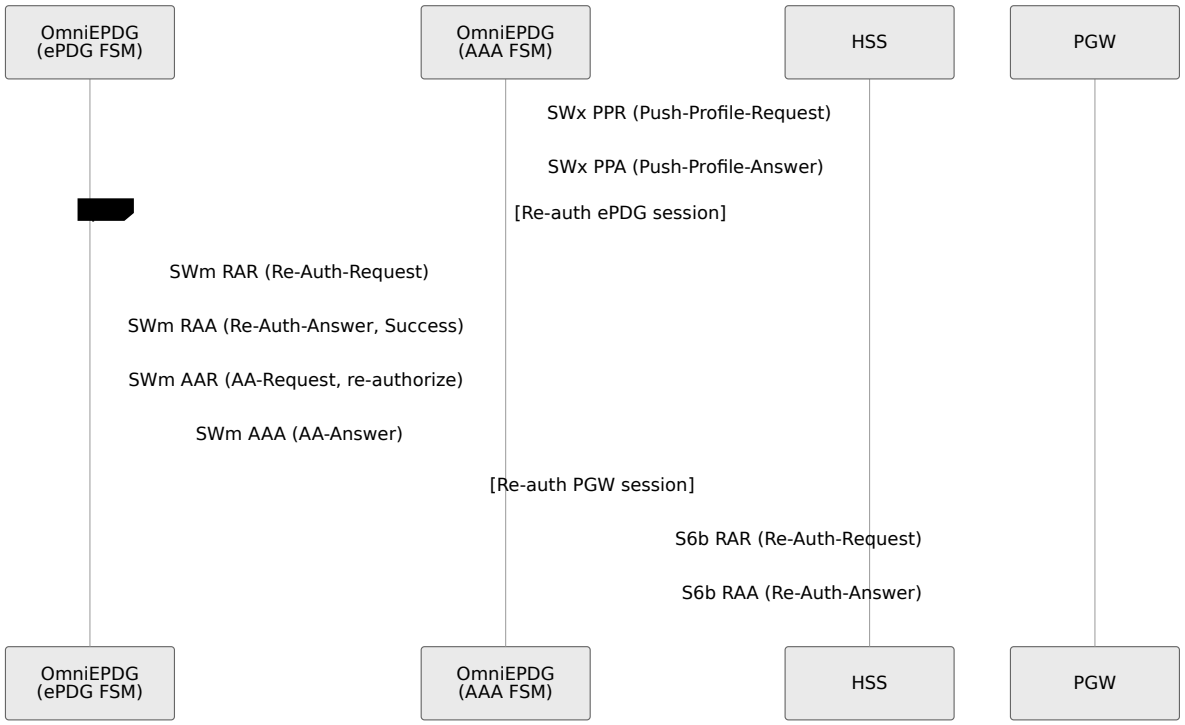
GTP 模式：PGW 发起的会话拆除

当 PGW 终止会话时（例如，策略违规、超时或管理操作）。

GTP 模式：网络（HSS）发起的注销

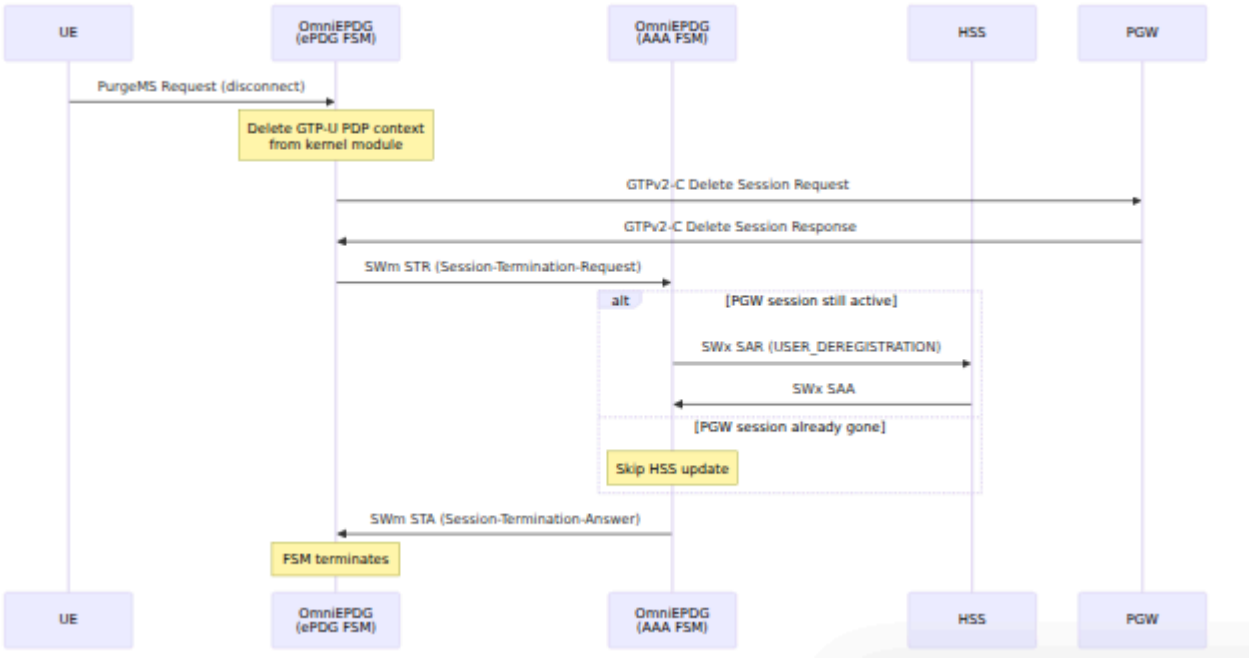
当 HSS 撤销用户的注册时（例如，订阅变更、欺诈检测或管理操作）。

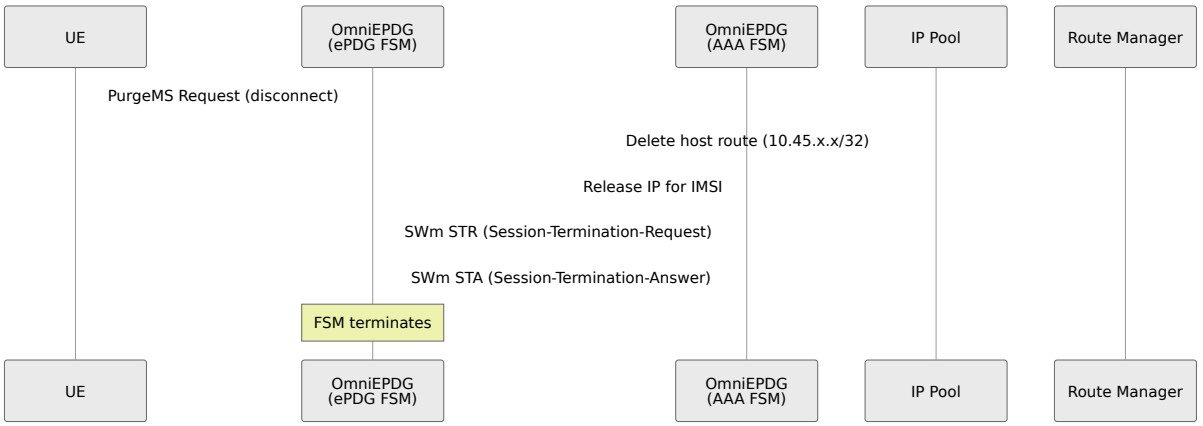




GTP 模式：HSS ��置文件推送和重新认证

当 HSS 推送更新的用户配置文件时，OmniEPDG 在 ePDG（SWm）和 PGW（S6b）会话上触发重新认证，按照 3GPP TS 29.273 第 8.1.2.3.3 节。





简单 VPN 模式：成功的会话建立

在简单 VPN 模式下，会话建立过程更短。在 EAP-AKA 身份验证后，ePDG FSM 从本地池分配一个 IP，并在 TUN 接口上设置主机路由，绕过所有 PGW 交互。如果启用了 `skip_sar`，则与 HSS 的 SAR/SAA 交换也会被跳过。

简单 VPN 模式：用户发起的会话拆除

在简单 VPN 模式下，当 UE ��开连接时，FSM 释放分配的 IP 地址并删除主机路由。

Diameter 应用程序 ID

应用程序 ID	接口	供应商 ID	描述	参考
16777265	SWx	10415 (3GPP)	ePDG ↔ HSS 身份验证和用户管理	3GPP TS 29.273
16777272	S6b	10415 (3GPP)	AAA ↔ PGW 会话授权	3GPP TS 29.273

Diameter 结果代码

OmniEPDG 将 Diameter 结果代码映射到内部原因值，以实现跨协议的错误传播。

标准结果代码

结果代码	名称	意义
2001	DIAMETER_SUCCESS	操作成功完成
2002	DIAMETER_LIMITED_SUCCESS	操作部分成功

3GPP 实验性结果代码

结果代码	名称	意义	参
4181	DIAMETER_AUTHENTICATION_DATA_UNAVAILABLE	HSS 暂时无法提供身份验证数据	3GP TS 29.2
5001	DIAMETER_ERROR_USER_UNKNOWN	用户 IMSI 在 HSS 中未找到	3GP TS 29.2
5002	DIAMETER_UNKNOWN_SESSION_ID	会话未找到（用于过期的 STR/AAR）	RFC 673
5003	DIAMETER_AUTHORIZATION_REJECTED	用户未被授权使用该服务	3GP TS 29.2
5004	DIAMETER_ERROR_ROAMING_NOT_ALLOWED	适用漫游限制	3GP TS 29.2
5005	DIAMETER_MISSING_AVP	消息中缺少所需的 AVP	RFC 673
5012	DIAMETER_UNABLE_TO_COMPLY	一般处理失败	RFC 673
5420	DIAMETER_ERROR_UNKNOWN_EPS_SUBSCRIPTION	未找到 EPS 订阅	3GP TS 29.2
5421	DIAMETER_ERROR_RAT_NOT_ALLOWED	不允许访问技术	3GP TS 29.2

结果代码	名称	意义	参
5422	DIAMETER_ERROR_EQUIPMENT_UNKNOWN	设备 IMEI 未被识别	3GP TS 29.2

GTPv2-C 原因代码（仅限 GTP 模式）

OmniEPDG 处理 PGW 返回的创建/删除会话响应中的以下 GTPv2-C 原因代码。代码 1-15 为信息性，16-63 表示成功，64 及以上表示错误。请参见 3GPP TS 29.274 第 8.4 节。

成功原因

代码	名称	描述
16	Request Accepted	操作成功完成
17	Request Accepted Partially	部分成功
18	New PDN Type (Network Preference)	由于网络偏好而更改 PDN 类型
19	New PDN Type (Single Address Bearer)	由于单地址承载限制而更改 PDN 类型

错误原因（选择）

代码	名称	描述
64	Context Not Found	在 PGW 上未找到会话上下文
73	No Resources Available	PGW 资源耗尽
78	Missing or Unknown APN	请求的 APN 未在 PGW 上配置
82	Denied in RAT	不允许访问技术
84	All Dynamic Addresses Occupied	PGW 上的 IP 地址池耗尽
92	User Authentication Failed	PGW 身份验证失败
93	APN Access Denied	用户未被授权使用 APN
96	IMSI/IMEI Not Known	用户身份未被识别
109	Invalid Peer	对等方验证失败
113	APN Congestion	APN 超载
120	GTP-C Entity Congestion	PGW 控制平面超载

NAI 格式

OmniEPDG 使用在 3GPP TS 23.003 第 19 节中定义的网络接入标识符（NAI）格式识别用户：

```
<prefix><IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
```

身份前缀和身份验证类型

NAI 前缀根据 3GPP TS 23.003 确定 EAP 身份验证方法：

前缀	身份验证类型	描述
`0`	EAP-AKA	标准 AKA 身份验证（WiFi 通话最常见）
`6`	EAP-AKA'	带有网络绑定的增强型 AKA 身份验证

OmniEPDG 根据 UE 的身份前缀自动选择身份验证方法。大多数 UE 使用前缀 `0`（EAP-AKA）进行 WiFi 通话。

OmniEPDG 通过解析前缀和 `@` 符号之间的所有内容，从 NAI 中提取 IMSI。然后，IMSI 被用作所有每个用户状态机和信令操作的主键。

加密算法

OmniEPDG 根据 3GPP TS 33.402 和 RFC 7296（IKEv2）实现加密算法。

IKEv2 加密算法

算法	**ID**	密钥大小	状态	参考
AES-CBC	12	128, 192, 256 bits	支持（默认：256）	RFC 3602
AES-GCM-16	20	128, 192, 256 bits	支持	RFC 5282
AES-GCM-12	19	128, 192, 256 bits	支持	RFC 5282
AES-GCM-8	18	128, 192, 256 bits	支持	RFC 5282
3DES	3	192 bits	支持（遗留）	RFC 2451

IKEv2 完整性算法

算法	ID	密钥大小	ICV 大小	状态	参考
HMAC-SHA2-256-128	12	256 bits	128 bits	支持（默认）	RFC 4868
HMAC-SHA2-384-192	13	384 bits	192 bits	支持	RFC 4868
HMAC-SHA2-512-256	14	512 bits	256 bits	支持	RFC 4868
AES-XCBC-96	5	128 bits	96 bits	支持	RFC 3566
HMAC-SHA1-96	2	160 bits	96 bits	支持（遗留）	RFC 2404
HMAC-MD5-96	1	128 bits	96 bits	支持（遗留）	RFC 2403

AES-XCBC-96 (`AUTH_AES_XCBC_96`) □某���手机（特别□某些三星 VoWiFi 客户端）所需的，它将其作为唯一的完整性变换。它根据 RFC 3566 实现，具有 RFC 4434 和 RFC 7296 第 2.14 节中定义的可变密钥处理和 SKEYSEED 随机数截断。

IKEv2 PRF 算法

算法	ID	输出大小	状态	参考
PRF-HMAC-SHA2-256	5	256 bits	支持（默认）	RFC 4868
PRF-HMAC-SHA2-384	6	384 bits	支持	RFC 4868
PRF-HMAC-SHA2-512	7	512 bits	支持	RFC 4868
PRF-AES128-XCBC	4	128 bits	支持	RFC 4434
PRF-HMAC-SHA1	2	160 bits	支持（遗留）	RFC 2104
PRF-HMAC-MD5	1	128 bits	支持（遗留）	RFC 2104

当协商 `PRF-AES128-XCBC` 时，SKEYSEED 仅从 Ni 和 Nr 的前 64 位派生（根据 RFC 7296 第 2.14 节），而完整的随机数用于 prf+ 密钥扩展。

IKEv2 Diffie-Hellman 组

组	ID	大小	状态	参考
MODP-2048	14	2048 bits	支持（默认）	RFC 3526
MODP-1024	2	1024 bits	支持（遗留）	RFC 2409
MODP-1536	5	1536 bits	支持	RFC 3526
MODP-3072	15	3072 bits	支持	RFC 3526
MODP-4096	16	4096 bits	支持	RFC 3526
ECP-256	19	256 bits	支持	RFC 5903
ECP-384	20	384 bits	支持	RFC 5903
ECP-521	21	521 bits	支持	RFC 5903
Curve25519	31	256 bits	支持	RFC 8031
Curve448	32	448 bits	支持	RFC 8031

ESP（子 SA）算法

ESP 隧道算法从 UE 在 IKE_AUTH 中提供的 SA 提案中协商的（在重新密钥时的 CREATE_CHILD_SA 中）。OmniEPDG 选择 UE 实际提供的变换，并使用匹配的密钥和 ICV 大小派生子 SA 密钥。它不强制固定套件。

类型	算法	备注
加密	AES-CBC-128 / 192 / 256	密码遵循协商的密钥长度
完整性	HMAC-SHA2-256-128, HMAC-SHA2-384-192, HMAC-SHA2-512-256, AES-XCBC-96, HMAC-SHA1-96	AES-XCBC-96 ICV 为 12 字节（与 HMAC-SHA2-256-128 的 16 字节相比）

回退（仅当无法解析 **UE** 的提案时）：

- 加密：AES-CBC-256（32 字节密钥，16 字节 IV）
- 完整性：HMAC-SHA2-256-128（32 字节密钥，16 字节 ICV）

EAP-AKA 加密函数

函数	算法	参考
MK 派生	SHA-1	RFC 4187 第 7 节
PRF+ 密钥扩展	FIPS 186-2 PRF (SHA-1)	RFC 4187 附录 D
AT_MAC	HMAC-SHA1-128	RFC 4187 第 10.15 节
Milenage (f1-f5)	AES-128	3GPP TS 35.206

EAP-AKA' 加密函数

函数	算法	参考
CK'/IK' 派生	HMAC-SHA-256	RFC 5448 第 3.3 节
MK 派生	SHA-256	RFC 5448 第 3.4 节
AT_MAC	HMAC-SHA256-128	RFC 5448 第 3.1 节

3GPP 合规性

OmniEPDG 实现了 3GPP TS 33.402 第 8 节中规定的所有强制加密算法:

要求	算法	状态
IKEv2 加密（强制）	AES-CBC-128	✓ 支持
IKEv2 完整性（强制）	HMAC-SHA2-256-128	✓ 支持（默认）
IKEv2 PRF（强制）	PRF-HMAC-SHA-256	✓ 支持（默认）
IKEv2 DH（强制）	组 14（MODP-2048）	✓ 支持（默认）
ESP 加密（强制）	AES-CBC-128/256	✓ 支持
ESP 完整性（强制）	HMAC-SHA2-256-128	✓ 支持（默认）
EAP-AKA	RFC 4187	✓ 实现
EAP-AKA'	RFC 5448	✓ 实现

除了强制集，OmniEPDG 还支持 AES-XCBC 完整性和 PRF 变换（RFC 3566 / RFC 4434），以便与仅提供这些变换的手机进行互操作：例如某些三星 VoWiFi 客户端，它们提议 AES-CBC-256 / AUTH_AES_XCBC_96 / PRF_AES128_XCBC / MODP-2048，并且不会回退到 HMAC-SHA2。

PDP 地址类型（仅限 GTP 模式）

OmniEPDG 支持 3GPP TS 29.274 第 8.14 节中定义的以下 PDP 地址类型。在简单 VPN 模式下，仅从本地池分配 IPv4 地址。

类型	描述	**GTPv2-C PAA** 格式
IPv4	仅 IPv4 承载	4 字节 IPv4 地址
IPv6	仅 IPv6 承载	1 字节前缀长度 + 16 字节 IPv6 地址
IPv4v6	双栈承载	1 字节前缀长度 + 16 字节 IPv6 + 4 字节 IPv4

OmniEPDG 配置参考

OmniEPDG 通过 `config/runtime.exs` 和环境变量进行配置。所有面向客户的配置在运行时进行 - 编译时默认值内置于发布中并未暴露。

对于容器化部署，请使用 环境变量参考 部分中记录的环境变量。

目录

- IKEv2 服务器参数
- 认证安全参数
- VPN 模式选择
- 简单 VPN 参数
- Diameter 参数
- 许可证客户端配置
- 控制面板配置
- Prometheus 指标配置
- Webhooks 配置
- 环境变量参考
- 超时参考

配置结构

配置文件

所有配置都在 **`config/runtime.exs`** 中完成。OmniEPDG 启动时会读取此文件。该文件支持容器化部署的环境变量替换。

配置示例

```
# config/runtime.exs
config :omniepdg,
  # IKEv2 服务器设置
  listen_ip: {0, 0, 0, 0},
  port_500: 500,
  port_4500: 4500,

  # VPN 模式: :simple (本地突破) 或 :gtp (通过 GTP-C 的 PGW)
  vpn_mode: :simple,

  # 简单 VPN 模式设置
  simple_vpn: [
    pool_ipv4: "10.45.0.0/16",
    pool_ipv6: "2001:db8::/32",
    dns_servers_ipv4: ["8.8.8.8", "8.8.4.4"],
    dns_servers_ipv6: ["2001:4860:4860::8888",
"2001:4860:4860::8844"]
  ]

# 控制面板配置
config :control_panel,
  parent_application: :omniepdg,
  parent_application_readable_name: "OmniEPDG",
  use_additional_pages: [
    {OmniEpdg.Web.DashboardLive, "/", "Dashboard"},
    {OmniEpdg.Web.SessionsLive, "/sessions", "Sessions"},
    {OmniEpdg.Web.DiameterLive, "/diameter", "Diameter"}
  ]

# Diameter 配置 (runtime.exs)
config :diameter_ex,
  diameter: %{
    service_name: :omniepdg,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    host: "epdg",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    product_name: "OmniEPDG",
    vendor_id: 10415,
    applications: [
      %{application_name: :swx, application_id: 16_777_265,
```

```
vendor_id: 10415},
      %{application_name: :s6b, application_id: 16_777_272,
vendor_id: 10415}
    ],
    peers: [
      %{host: "hss", ip: "127.0.0.1", port: 3868, transport: :tcp}
    ]
  }

# 许可证客户端配置 (runtime.exs)
config :license_client,
  server_url: "https://license.example.com/api",
  product: "omniepdg"
```

IKEv2 服务器参数

IKEv2 服务器处理 UEs 和 OmniEPDG 之间的 SWu 接口。它终止 IPSec 隧道并执行 EAP-AKA 认证。

□□	□□	□□	□□
`listen_ip`	□□	□	`{0, 0, 0, 0}`
`port_500`	□□	□	`500`
`port_4500`	□□	□	`4500`
`cert_file`	□□□	□	`/etc/omniepdg/certs/epdg.crt`

参数	类型	必需	默认
`key_file`	字符串	□	`/etc/omniepdg/certs/epdg.key`
`session_inactivity_timeout_ms`	整数	否	`300000`

认证安全参数

OmniEPDG 包含内置的防止暴力攻击和地理访问控制的保护。有关详细文档，请参见 安全指南。

速率限制参数

```
config :omniepdg,
  # 每个 IP 的□率限制
  auth_rate_limit_per_ip: 10,
  auth_rate_limit_ip_window_ms: 60_000,
  auth_rate_limit_ip_block_ms: 300_000,

  # 每个 IMSI 的□率限制
  auth_rate_limit_per_imsi: 5,
  auth_rate_limit_imsi_window_ms: 60_000,
  auth_rate_limit_imsi_block_ms: 600_000
```

参数	类型	必需	默认	描述
`auth_rate_limit_per_ip`	整数	否	`10`	每个 IP 的最大失败认证尝试次数，超过后将被阻止。
`auth_rate_limit_ip_window_ms`	整数	否	`60000`	计算 IP 失败次数的滑动窗口（毫秒）。
`auth_rate_limit_ip_block_ms`	整数	否	`300000`	超过阈值的 IP 的阻止持续时间（5 分钟）。
`auth_rate_limit_per_imsi`	整数	否	`5`	每个 IMSI 的最大失败认证尝试次数，超过后将被阻止。
`auth_rate_limit_imsi_window_ms`	整数	否	`60000`	计算 IMSI 失败次数的滑动窗口（毫秒）。
`auth_rate_limit_imsi_block_ms`	整数	否	`600000`	超过阈值的 IMSI 的阻止持续时间（10 分钟）。

GeoIP 参数

```
config :omniepdg,
  geoip_enabled: false,
  geoip_database_path: "/etc/omniepdg/GeoLite2-Country.mmdb",
  geoip_mode: :whitelist,
  geoip_countries: ["AU", "NZ"],
  geoip_allow_unknown: false,
  geoip_fail_open: true
```

□□	□□	□□	□□	□□
`geoip_enabled`	□□	□	`false`	□□□□ GeoIP □□□□□□□□ MaxMind GeoLite2 □□□□
`geoip_database_path`	□□□	□	`"/etc/omniepdg/GeoLite2-Country.mmdb"`	MaxMind GeoLite2 □□□□□□□ □MMDB □□□□
`geoip_mode`	□□	□	`:whitelist`	`:whitelist` □□□□□□□□□□□□ □ `:blacklist` □□□□□□□□□□□
`geoip_countries`	□□	□	`[]`	ISO 3166-1 alpha-2 □□□□□□□□□ `["AU", "NZ", "US"]` □□
`geoip_allow_unknown`	□□	□	□□□□	□□□□□□□□□□□□ □ IP□□□□□□□□ □ `false` □□□□□ `true` □
`geoip_fail_open`	□□	□	`true`	□□ GeoIP □□ □□□□□□□□□□□□ □□□□

VPN 模式选择

参数	类型	必需	默认	环境变量	描述
`vpn_mode`	原子	否	`:simple`	`EPDG_VPN_MODE`	操作模式：`:simple` 表示通过 TUN 接口的本地 IP 突破，`:gtp` 表示通过 GTPv2-C/GTP-U 的 PGW 隧道。有关详细比较，请参见 操作指南。

简单 VPN 参数

`simple_vpn` 配置块控制简单 VPN 模式下的 IP 分配和 DNS。OmniEPDG 支持 IPv4 和 IPv6 地址池。

```
config :omniepdg,
  simple_vpn: [
    pool_ipv4: "10.45.0.0/16",
    pool_ipv6: "2001:db8::/32",
    dns_servers_ipv4: ["8.8.8.8", "8.8.4.4"],
    dns_servers_ipv6: ["2001:4860:4860::8888",
"2001:4860:4860::8844"],
    p_cscf_ipv4: ["10.4.12.165"],
    p_cscf_ipv6: [],
    mtu: 1400,
    nat_enabled: true
  ]
```

□□	□□	□□	□□	□□
`pool_ipv4`	□□□	□	`"10.45.0.0/16"`	IPv4 □□□□□□ □ CIDR □□ □□□□□□□□□□ □ IP□
`pool_ipv6`	□□□	□	`"2001:db8::/32"`	□□□□□ IPv6 □□□□□□ CIDR □□□□
`dns_servers_ipv4`	□□	□	`["8.8.8.8", "8.8.4.4"]`	□□ PCO□□□ □□□□□□□□ UEs □ IPv4 DNS □□□□
`dns_servers_ipv6`	□□	□	`["2001:4860:4860::8888", "2001:4860:4860::8844"]`	□□ PCO □□□ UEs □ IPv6 DNS □□□□
`p_cscf_ipv4`	□□	□	`[]`	VoWiFi □ IPv4 P-CSCF□IMS □ □□□□□□□□ UEs □□□ IMS □□□
`p_cscf_ipv6`	□□	□	`[]`	VoWiFi □ IPv6 P-CSCF□IMS □ □□□□□
`mtu`	□□	□	`1400`	□□□□□ MTU □□□□□□□□□□ □□□

参数	类型	必需	默认	描述
`nat_enabled`	布尔	否	`true`	启用用户流量的 NAT。当为 true 时，将对出站流量应用伪装规则。

Diameter 参数

Diameter 配置控制 SWx（HSS）和 S6b（PGW）接口。当 `diameter_enabled` 为 `true` 时，OmniEPDG 启动 Diameter 栈并连接到配置的对等体。

```
config :diameter_ex,
  diameter: %{
    service_name: :omniepdg,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    host: "epdg",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    product_name: "OmniEPDG",
    vendor_id: 10415,
    applications: [
      %{application_name: :swx, application_id: 16_777_265,
vendor_id: 10415},
      %{application_name: :s6b, application_id: 16_777_272,
vendor_id: 10415}
    ],
    peers: [
      %{host: "hss", ip: "10.74.0.21", port: 3868, transport:
:tcp}
    ]
  }
```

服务参数

□□	□□	□□	□□	□□
`service_name`	□□	□	`:omniepdg`	-
`listen_ip`	□□□	□	`"0.0.0.0"`	`DIA_LIST`
`listen_port`	□□	□	`3868`	`DIA_LIST`
`host`	□□□	□	`"epdg"`	`DIA_HOST`
`realm`	□□□	□	`"epc.mnc001.mcc001.3gppnetwork.org"`	`DIA_REAL`

参数	类型	必需	默认	环境
`product_name`	字符串	否	`"OmniEPDG"`	-
`vendor_id`	整数	否	`10415`	-

对等体参数

`peers` 列表中的每个条目定义一个 Diameter 对等体连接（通常 HSS）。

参数	类型	必需	默认	环境变量	描述
`host`	字符串		-	`HSS_HOST`	对等体的 Diameter 身份（Origin-Host）。必须与对等体的配置身份匹配。
`ip`	字符串		-	`HSS_IP`	对等体的 IP 地址，用于 TCP/SCTP 连接。
`port`	整数	否	`3868`	`HSS_PORT`	对等体的 Diameter 端口。
`transport`	原子	否	`:tcp`	-	传输协议：`:tcp` 或 `:sctp`。

应用程序 **ID**

应用程序	ID	供应商 ID	接口	参考
SWx	16777265	10415	ePDG ↔ HSS	3GPP TS 29.273
S6b	16777272	10415	AAA ↔ PGW	3GPP TS 29.273

许可证客户端配置

许可证客户端验证 OmniEPDG 与许可证服务器的关系。

```
config :license_client,
  server_url: "https://license.example.com/api",
  product: "omniepdg"
```

参数	类型	必需	默认	环境变量	描述
`server_url`	字符串	□	-	`LICENSE_SERVER_URL`	许可证服务器 API 端点的 URL。
`product`	字符串	否	`"omniepdg"`	-	许可证验证的产品标识符。

控制面板配置

Web 控制面板提供监控和管理功能。

```
config :control_panel,
  port: 4000
```

参数	类型	必需	默认	环境变量	描述
`port`	整数	否	`4000`	`CONTROL_PANEL_PORT`	控制面板 Web 界面的 HTTP 端口。

Prometheus 指标配置

OmniEPDG 通过 HTTP 暴露 Prometheus 指标以进行监控和警报。

```
config :omniepdg,
  prometheus: %{
    port: 9568
  }
```

参数	类型	必需	默认	环境变量	描述
`port`	整数	否	`9568`	`PROMETHEUS_PORT`	Prometheus 指标端点（`/metrics`）的 HTTP 端口。

暴露的指标

计数器指标（事件驱动）：

- `epdg_ikev2_session_initiated_count` - 启动的 IKE_SA_INIT 交换
- `epdg_ikev2_session_established_count` - 成功建立的 IKE SAs
- `epdg_ikev2_session_failed_count` - IKE SA 建立失败（按原因）
- `epdg_eap_identity_count` - EAP 身份请求
- `epdg_eap_aka_challenge_count` - 发送的 EAP-AKA 挑战
- `epdg_eap_aka_success_count` - 成功的 EAP-AKA 认证
- `epdg_eap_aka_failure_count` - 失败的 EAP-AKA 认证（按原因）
- `epdg_eap_aka_sync_failure_count` - EAP-AKA SQN 同步失败
- `epdg_diameter_swx_mar_count` - 多媒体认证请求（按结果）
- `epdg_diameter_swx_sar_count` - 服务器分配请求（按结果）
- `epdg_diameter_s6b_aar_count` - 处理的 AA 请求（按结果）

- `epdg_diameter_s6b_str_count` - 会话终止请求
- `epdg_session_created_count` - 创建的会话（按 vpn_mode）
- `epdg_session_terminated_count` - 终止的会话（按原因）
- `epdg_esp_packets_in_count` - 解密的 ESP 数据包
- `epdg_esp_packets_out_count` - 加密的 ESP 数据包
- `epdg_ip_allocated_count` - 分配的 IP 地址（按类型）
- `epdg_ip_released_count` - 释放的 IP 地址（按类型）

仪表指标（每 **5** 秒轮询）：

- `epdg_sessions_active_count` - 当前活动会话总数
- `epdg_sessions_by_state_count` - 按 FSM 状态的会话
- `epdg_ip_pool_allocated_count` - 当前分配的 IP
- `epdg_ip_pool_available_count` - 池中可用的 IP
- `epdg_ip_pool_utilization_ratio` - 池的利用率（0.0-1.0）
- `epdg_diameter_swx_pending_count` - 待处理的 SWx 请求
- `epdg_diameter_s6b_active_sessions_count` - 活动的 S6b 会���

直方图指标（延迟跟踪）：

- `epdg_auth_duration_ms` - 完整认证流程持续时间
- `epdg_diameter_swx_mar_latency_ms` - MAR 响应时间
- `epdg_diameter_swx_sar_latency_ms` - SAR 响应时间
- `epdg_session_duration_seconds` - 会话生命周期

VM 指标:

- `vm_memory_total` - VM 总内存
- `vm_memory_processes` - 进程内存
- `vm_memory_binary` - 二进制内存
- `vm_memory_ets` - ETS 表内存
- `vm_system_info_process_count` - 正在运行的进程
- `vm_system_info_port_count` - 打开的端口
- `vm_statistics_run_queue` - 调度程序运行队列

Prometheus 抓取配置

```
scrape_configs:
  - job_name: 'omniepdg'
    static_configs:
      - targets: ['localhost:9568']
```

Webhooks 配置

OmniEPDG 可以发送出站 webhooks 以处理用户会话生命周期事件。当启用时，OmniEPDG 会在成功附加和分离时向每个配置的端点发送带有 JSON 主体的 HTTP `POST` 请求。有关交付模型、有效负载架构和示例有效负载，请参见 Webhooks。

```
config :omniepdg, :webhook,
  enabled: false,
  endpoints: ["https://127.0.0.1/webhook"],
  events: [:session_established, :session_terminated]
```

参数	类型	必需	默认	描述
`enabled`	布尔	否	`false`	启用或禁用出站 webhooks `false` 时，OmniEPDG 送任何 webhooks。
`endpoints`	列表	否	`["https://127.0.0.1/webhook"]`	端点 URL 列表。OmniEP 每个事件对每个端点发送一个 POS
`events`	列表	否	`[:session_established, :session_terminated]`	要发送的事件列表。有效值为 `:session_establis` （成功附加）和 `:session_terminat` （分离）。

OmniEPDG 以火忘记模型交付 webhooks，并且不会重试失败的交付。有关完整的交付模型和 JSON 有效负载架构，请参见 Webhooks。

超时参考

所有内部 FSM 超时都□硬编码的。这些超时控制状态机在认为响应失败之前等待的时间。

超时	值	模式	上下文	描述
GTP 答复	10,000 ms	GTP	ePDG UE FSM	最大等待 PGW 的 GTPv2-C 创建/删除会话响应的时间。
SWm 答复	10,000 ms	两者	ePDG UE FSM	最大等待内部 Diameter SWm 响应（DER/DEA, STR/STA）的时间。
S6b 答复	10,000 ms	GTP	AAA UE FSM	最大等待 Diameter S6b 响应（ASR/ASA）的时间。

环境变量参考

环境变量在 `config/runtime.exs` 中读取，并覆盖编译时默认值。

IKEv2 服务器

变量	默认	描述
`EPDG_LISTEN_IP`	`"0.0.0.0"`	IKEv2 服务器绑定地[（点分十进制格式，例如，`"10.0.0.1"`
`EPDG_PORT_500`	`"500"`	IKE 协议端口。
`EPDG_PORT_4500`	`"4500"`	IKE NAT-Traversal 端口。
`EPDG_CERT_FILE`	`"/etc/omniepdg/certs/epdg.crt"`	IKEv2 服务器证书的[（PEM）。
`EPDG_KEY_FILE`	`"/etc/omniepdg/certs/epdg.key"`	IKEv2 服务器私钥的[（PEM）。
`EPDG_SESSION_TIMEOUT`	`"300000"`	会话不活动超时时间（毫[

VPN 模式

变量	默认	描述
`EPDG_VPN_MODE`	`"simple"`	VPN 模式: `"simple"` 或 `"gtp"`。

Diameter

变量	默认	描述
`DIA_LISTEN_IP`	`"0.0.0.0"`	Diameter 监听器绑定地址。
`DIA_LISTEN_PORT`	`"3868"`	Diameter 监听器端口。
`DIA_HOST`	`"epdg"`	Diameter Origin-Host（不带领域）。
`DIA_REALM`	`"epc.mnc001.mcc001.3gppnetwork.org"`	Diameter Origin-Realm。

HSS 对等体

变量	默认	描述
`HSS_HOST`	`"hss"`	HSS Diameter 身份（Origin-Host）。
`HSS_IP`	`"127.0.0.1"`	HSS IP 地址。
`HSS_PORT`	`"3868"`	HSS Diameter 端口。

许可证、控制面板和指标

变量	默认	描述
`LICENSE_SERVER_URL`	-	许可证服务器 API URL（必需）。
`CONTROL_PANEL_PORT`	`"4000"`	控制面板 HTTP 端口。
`PROMETHEUS_PORT`	`"9568"`	Prometheus 指标 HTTP 端口（`/metrics` 端点）。

□□□Docker Compose

```yaml
services:
  omniepdg:
    image: omniepdg:latest
    environment:
      # IKEv2
      EPDG_LISTEN_IP: "0.0.0.0"
      EPDG_CERT_FILE: "/certs/epdg.crt"
      EPDG_KEY_FILE: "/certs/epdg.key"

      # VPN □□
      EPDG_VPN_MODE: "simple"

      # Diameter
      DIA_HOST: "epdg"
      DIA_REALM: "epc.mnc001.mcc001.3gppnetwork.org"
      HSS_HOST: "hss"
      HSS_IP: "10.74.0.21"
      HSS_PORT: "3868"

      # □□□
      LICENSE_SERVER_URL: "https://license.example.com/api"

      # □□□□
      CONTROL_PANEL_PORT: "4000"

      # Prometheus □□
      PROMETHEUS_PORT: "9568"
    ports:
      - "500:500/udp"
      - "4500:4500/udp"
      - "4000:4000"
      - "9568:9568"
    volumes:
      - ./certs:/certs:ro
    cap_add:
      - NET_ADMIN
```

OmniEPDG 控制面板

OmniEPDG 包含一个基于网页的控制面板，用于实时监控会话、Diameter 对等体和系统日志。控制面板提供无刷新页面的实时更新视图。

目录

- 访问控制面板
- 仪表板
- 会话视图
- Diameter 对等体视图
- 日志视图
- 文档视图
- 资源视图
- 配置视图

访问控制面板

控制面板在配置的 HTTP 端口上提供（默认 4000）：

```
http://<host>:4000/dashboard
```

导航

控制面板提供一个侧边栏，包含所有视图的链接：

路径	视图	描述
`/dashboard`	仪表板	系统概览和快速链接
`/sessions`	会话	活跃的 UE 会话列表
`/diameter`	Diameter 对等体	Diameter 对等体连接状态
`/logs`	日志	实时日志流
`/docs`	文档	集成文档浏览器
`/resources`	资源	BEAM VM 和应用程序信息
`/configuration`	配置	系统配置查看器

仪表板

仪表板提供 OmniEPDG 状态的高层次概述，包括关键指标和快速导航。

统计卡片

仪表板显示四个主要统计信息：

统计信息	描述
活跃会话	当前已建立的 UE 会话数量
接收数据 **(UL)**	从 UE 接收的总字节数（上行方向）
发送数据 **(DL)**	发送到 UE 的总字节数（下行方向）
Diameter 对等体	已连接对等体 / 总配置对等体

数据值会自动缩放到适当的单位（B, KB, MB, GB）。

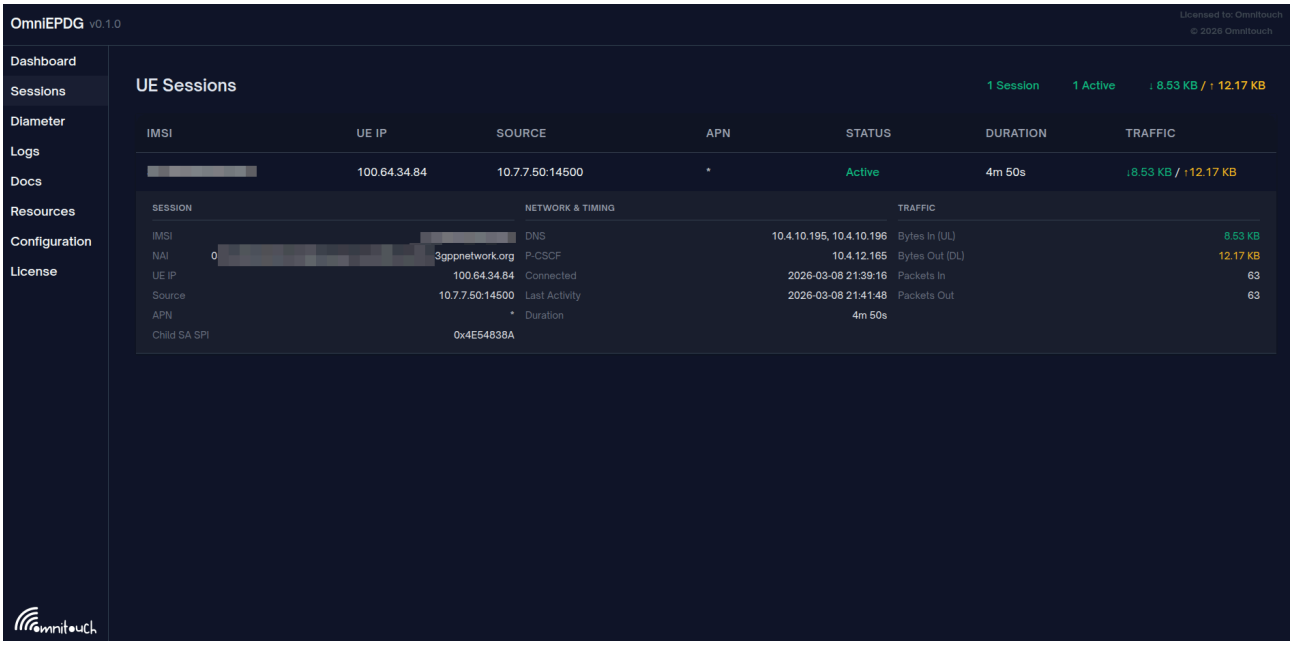
快速链接

直接导航到详细视图：

- 查看会话 - 导航到会话视图以获取详细的 UE 信息
- **Diameter** 对等体 - 导航到 Diameter 对等体视图以获取连接状态
- 系统日志 - 导航到日志视图以获取实时日志流
- 配置 - 导航到配置视图以获取系统设置

系统信息

显示当前操作配置：



字段	描述
VPN 模式	当前模式：`GTP` 或 `SIMPLE`
IKEv2 端口	标准端口：500 (IKE), 4500 (NAT-T)
Diameter 状态	□否启用了 Diameter 信令
IP 池 **(IPv4)**	配置的 IP 池 CIDR（仅限简单 VPN 模式）

自动刷新

仪表板每秒自动刷新一次，以显示当前统计信息。

会话视图

会话视图显示所有活跃的 UE 会话，并为每个订阅者提供详细信息。

□□□□□□□□□ UE □□□□□□□□□□□□□□□□□□□

会话列表

每个会话行显示：

列	描述
IMSI	订阅者的国际移动用户身份
UE IP	分配的 IPv4/IPv6 地址
SOURCE	UE 的外部 IP 和端口（NAT 地址）
APN	连接的接入点名称
STATUS	当前会话状态（活跃/非活跃）
DURATION	自会话建立以来的时间
TRAFFIC	接收/发送的字节（UL/DL）

状态指示器

会话通过颜色编码的徽章显示状态：

状态	颜色	描述
活跃	绿色	会话完全建立并正常运行
连接中	黄色	会话建立中
非活跃	红色	会话已终止或失败

会话详细信息

点击任何会话行以展开详细信息：

□□□□□□□□□ IMSI□NAI□□□□□□□□□□□

会话部分

字段	描述
IMSI	完整的 IMSI 值
NAI	网络接入标识符（3GPP 格式）
UE IP	分配的 IPv4/IPv6 地址
Source	UE 的外部 IP 和端口（NAT 地址）
APN	PDN 连接的接入点名称
Child SA SPI	IPSec 子 SA 安全参数索引

网络与时序部分

字段	描述
DNS	提供给 UE 的 DNS 服务器
P-CSCF	IMS 信令的代理-CSCF 服务器
Connected	会话建立的时间戳
Last Activity	最近数据包活动的时间戳
Duration	自会话建立以来的时间

流量部分

字段	描述
Bytes In (UL)	从 UE 接收的总字节数（上行）
Bytes Out (DL)	发送到 UE 的总字节数（下行）
Packets In	从 UE 接收的总数据包数
Packets Out	发送到 UE 的总数据包数

空状态

当没有活跃会话时，视图显示：

- "没有活跃会话" 消息
- 指示□否应尝试 UE 连接

自动刷新

会话列表每秒自动刷新一次。

Diameter 对等体视图

Diameter 对等体视图显示所有配置的 Diameter 对等体的状态（SWx 的 HSS，S6b 的 PGW）。

对等体列表

每个对等体行显示：

列	描述
Peer	Diameter 身份（Origin-Host）
Realm	Diameter 领域（Origin-Realm）
IP 地址	传输地址，格式为 `protocol://ip:port`
Status	连接状态

状态指示器

状态	颜色	描述
已连接	绿色	Diameter 对等体连接已建立
未连接	红色	对等体未连接
未知	灰色	状态无法确定

对等体计数摘要

标题显示汇总计数：

- **X** 已连接 - 活动连接的对等体数量
- **Y** 未连接 - 没有连接的对等体数量

对等体详细信息

点击任何对等体行以展开详细信息：

字段	描述
连接发起	谁发起连接：`local`（我们连接到对等体）或 `remote`（对等体连接到我们）
传输	协议：`tcp` 或 `sctp`
产品名称	对等体在 CER/CEA 中宣传的产品名称
宣传的应用程序	对等体支持的 Diameter 应用程序 ID

空状态

当没有配置对等体时，视图显示：

- "未配置 Diameter 对等体" 如果启用了 Diameter
- "Diameter 已禁用" 并提供配置提示如果禁用

自动刷新

对等体列表每秒自动刷新一次。

日志视图

日志视图提供系统日志的实时流式传输，具有过滤和搜索功能。

日志显示

日志以滚动容器的形式出现，最新条目在底部。每个日志条目显示：

元素	描述
时间戳	日志条目生成的时间
级别	带有颜色编码的严重性级别
消息	日志消息内容

日志级别

日志按严重性进行颜色编码：

级别	颜色	描述
debug	灰色	详细的诊断信息
info	蓝色	一般信息性消息
warning	黄色	警告条件
error	红色	错误条件

级别过滤

使用下拉菜单按最低严重性级别过滤日志：

过滤器	显示
所有级别	debug, info, warning, error
Info+	info, warning, error
Warning+	warning, error
Error Only	error

搜索

搜索框实时过滤日志：

- 输入任何文本以过滤日志消息
- 匹配不区分大小写
- 清空搜索框时会清除

控制

控制	描述
暂停/恢复	切换日志流的开/关
清除	移除所有显示的日�?
自动滚动	切换自动滚动到最新条目

日志缓冲区

- 最多保留 1000 条日志条目
- 达到限制时将移除最旧的条目
- 清除日志将从显示中移除所有条目

空状态

当没有日志与当前过滤器匹配时：

- "没有日志可显示" 消息
- 如果预期有日志，请验证过滤器设置

自动刷新

新日志会在生成时自动出现（当未暂停时）。

文档视图

文档视图提供集成的文档浏览器，允许操作员直接从控制面板访问所有 OmniEPDG 文档。

文档选择

使用按钮栏从可用文档文件中选择：

文档	描述
OPERATIONS.md	包含快速入门和程序的操作指南
README.md	项目概述和设置说明
architecture.md	系统架构和调用流程
configuration.md	完整的配置参考
control-panel.md	本控制面板指南
metrics.md	Prometheus 指标参考
troubleshooting.md	常见问题和解决步骤

Markdown 渲染

文档以完整的 Markdown 支持进行渲染，包括：

- 标题和文本格式
- 带语法高亮的代码块
- 表格
- 链接（内部和外部）
- 列表和引用块

资源视图

资源视图显示 BEAM VM 统计信息和正在运行的 OTP 应用程序。

系统指标

指标	描述
内存使用	BEAM VM 使用的总内存
BEAM 进程	正在运行的 Erlang/Elixir 进程数量
运行时间	自应用程序启动以来的时间

正在运行的应用程序

按类别列出所有加载的 OTP 应用程序：

类别	描述
主	OmniEPDG 应用程序
系统	核心 Erlang/OTP 和 Elixir 应用程序

点击应用程序以查看其详细信息，包括版本、描述和依赖关系。

配置视图

配置视图显示运行时配置和加载的应用程序。

环境信息

字段	描述
环境	当前 Mix 环境（开发/生产）
Elixir 版本	正在运行的 Elixir 版本

应用程序列表

显示所有加载的 OTP 应用程序及其版本。选择一个应用程序以查看：

- 应用程序描述
- 版本信息
- 依赖关系
- 配置参数

控制面板配置

HTTP 端口

在 `config/runtime.exs` 中配置控制面板端口:

```
config :omniepdg, OmniEpdg.Web.Endpoint,
  http: [port: 4000]
```

参数	类型	���认	描述
`port`	整数	4000	控制面板的 HTTP 端口

禁用控制面板

如果不需要，可以通过不在生产中启动网络端点来禁用控制面板。请联系您的系统集成商以获取特定于部署的配置。

OmniEPDG 指标参考

OmniEPDG 暴露 Prometheus 指标，以监控身份验证流程、会话生命周期、Diameter 信令和系统健康。指标通过 HTTP 提供，以供 Prometheus 抓取。

目录

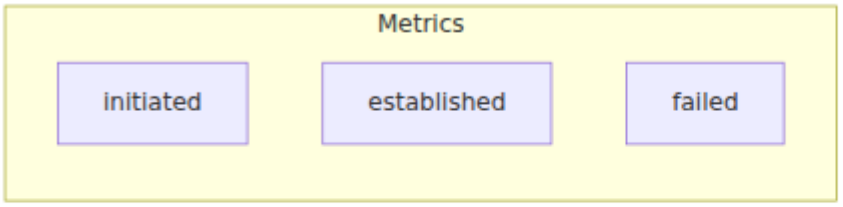
- 指标端点
- 配置
- 指标类别
  - IKEv2 会话指标
  - EAP 身份验证指标
  - 身份验证安全指标
  - Diameter SWx 指标
  - Diameter S6b 指标
  - 会话生命周期指标
  - ESP 数据平面指标
  - IP 池指标
  - VM 指标
- Prometheus 集成
- 示例查询
- 告警规则

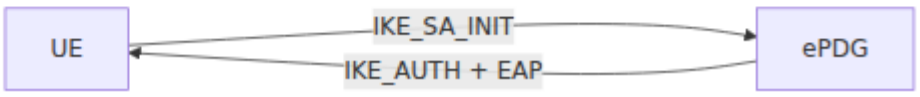
指标端点

OmniEPDG 在以下地址暴露指标：

```
http://<host>:9568/metrics
```

该端点以 Prometheus 展示格式返回指标，兼容 Prometheus、Grafana 和其他监控工具。





配置

`config/runtime.exs` 中配置指标端点：

```
config :omniepdg,
  prometheus: %{
    port: 9568
  }
```

参数	类型	默认值	环境变量	描述
`port`	整数	`9568`	`PROMETHEUS_PORT`	`/metrics` 端点的 HTTP 端口

指标类别

IKEv2 会话指标

跟踪 SWu 接口上 IKEv2 隧道建立的指标。

指标：`epdg_ikev2_session_initiated_count`

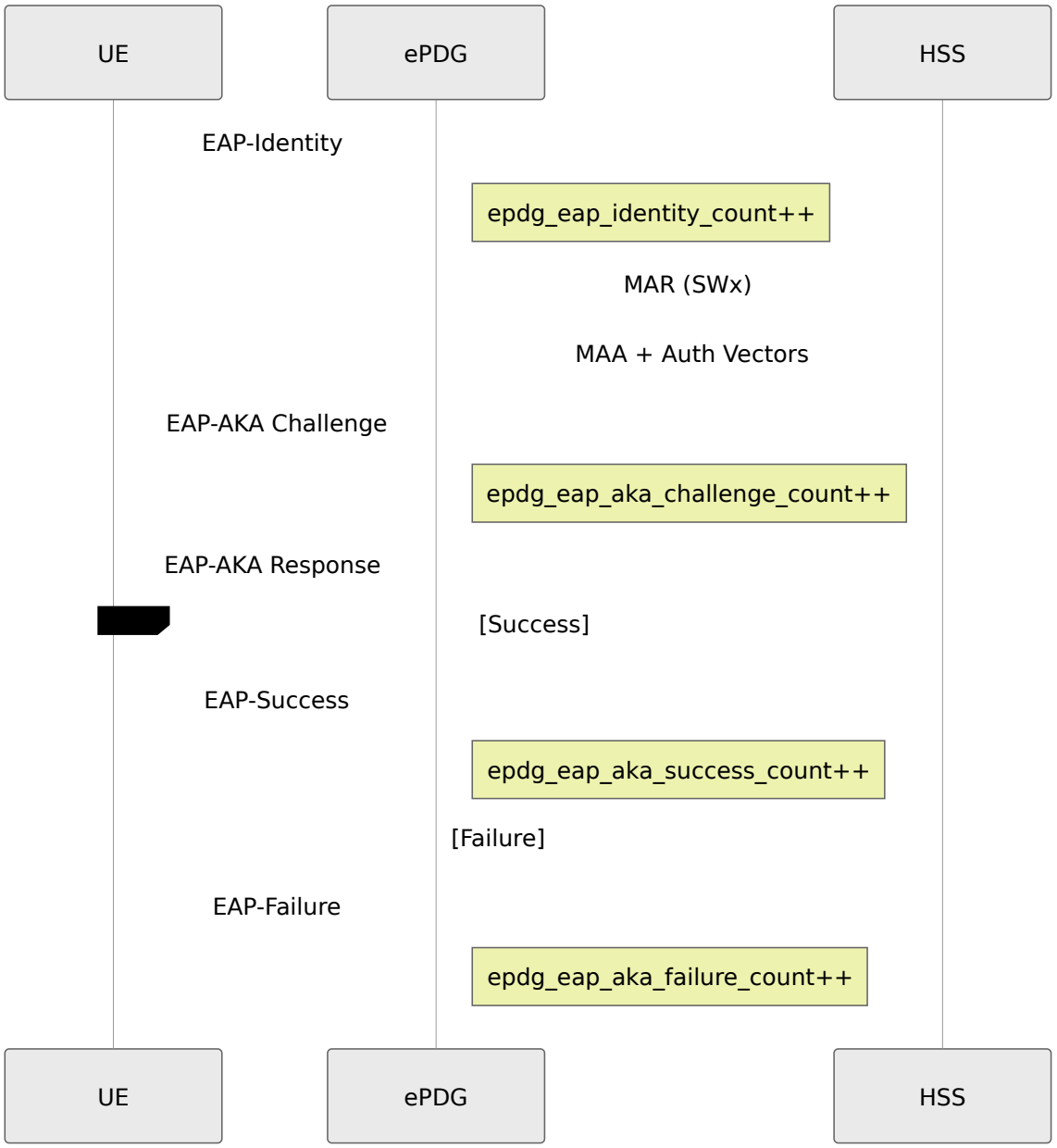
类型：计数器

描述：启动的 IKE_SA_INIT 交换总数。当 UE 启动隧道建立时递增。

指标：`epdg_ikev2_session_established_count`

类型：计数器

描述：成功建立的 IKE SA 总数。在成功的 EAP-AKA 身份验证和子 SA 创建后递增。



指标：`epdg_ikev2_session_failed_count`

类型：计数器

描述：IKE SA 建立失败的总数

标签：

- `reason` - 失败原因（例如，`auth_failed`，`timeout`，`invalid_proposal`）

EAP 身份验证指标

跟踪每个 RFC 4187 的 EAP-AKA 身份验证流程。OmniEPDG 还[illegible]持 RFC 5448 中的 EAP-AKA'，身份验证类型根据 UE 的 NAI 身份前缀自动选择。

指标：`epdg_eap_identity_count`

类型：计数器

描述：从 UE 接收到的 EAP-Identity 请求总数

指标：`epdg_eap_aka_challenge_count`

类型：计数器

描述：发送给 UE 的 EAP-AKA 挑战总数

指标：`epdg_eap_aka_success_count`

类型：计数器

描述：成功的 EAP-AKA 身份验证总数

指标：`epdg_eap_aka_failure_count`

类型：计数器

描述：失败的 EAP-AKA 身份验证总数

标签：

- `reason` - 失败原因（例如，`res_mismatch`，`invalid_identity`，`authentication_rejected`）

指标：`epdg_eap_aka_sync_failure_count`

类型：计数器

描述：EAP-AKA 序列号 (SQN) 同步失败的总数。指示 USIM/HSS 序列号不匹配，需要重新同步。

身份验证安全指标

身份验证安全层的指标。有关配置详细信息，请参见 安全指南。

指标：`epdg_auth_verification_failed_count`

类型：计数器

描述：AUTH 有效负载验证失败的总数。指示潜在的中间人攻击或实现错误。

指标：`epdg_auth_rate_limited_count`

类型：计数器

描述：被速率限制阻止的身份验证尝试总数

标签：

- `type` - 阻止原因：`ip`（每个 IP 阈值超出）或 `imsi`（每个 IMSI 阈值超出）

示例查询：

```
# 每分钟被限制的尝试
rate(epdg_auth_rate_limited_count[1m])

# 按类型限制
sum by (type) (rate(epdg_auth_rate_limited_count[5m]))
```

指标：`epdg_auth_geoip_blocked_count`

类型：计数器

描述：被 GeoIP 国家过滤阻止的身份验证尝试总数

标签：

- `country` - ISO 3166-1 alpha-2 国家代码（例如，`CN`，`RU`），或 `UNKNOWN` 对于无法进行地理定位的 IP

示例查询：

```
# 每分钟 GeoIP 阻止
rate(epdg_auth_geoip_blocked_count[1m])

# 被阻止的国家[illegible]十名
topk(10, sum by (country) (epdg_auth_geoip_blocked_count))
```

指标：`epdg_esp_replay_detected_count`

类型：计数器

描述：由于重放检测而被拒绝的 ESP 数据包总数（根据 RFC 4303）。指示潜在的重放攻击或网络问题导致的数据包重排序。

Diameter SWx 指标

根据 3GPP TS 29.273 的 ePDG 和 HSS 之间的 SWx 接口的指标。

指标: `epdg_diameter_swx_mar_count`

类型: 计数器

描述: 发送到 HSS 的多媒体身份验证请求总数，用于检索身份验证向量

标签:

- `result` - 请求结果: `success` 或 `failure`

指标: `epdg_diameter_swx_sar_count`

类型: 计数器

描述: 发送到 HSS 的服务器分配请求总数，用于注册/注销

标签:

- `result` - 请求结果: `success` 或 `failure`

指标: `epdg_diameter_swx_mar_latency_ms`

类型: 直方图

描述: MAR 响应时间（以毫秒为单位）

桶: 50, 100, 250, 500, 1000, 2500 ms

指标: `epdg_diameter_swx_sar_latency_ms`

类型: 直方图

描述: SAR 响应时间（以毫秒为单位）

桶: 50, 100, 250, 500, 1000, 2500 ms

指标: `epdg_diameter_swx_pending_count`

类型: 计量���

描述: 当前等待响应的 SWx 请求数量。高值表示 HSS 拥堵或连接问题。

Diameter S6b 指标

根据 3GPP TS 29.273 的 AAA 服务器和 PGW 之间的 S6b 接口的指标。仅适用于 GTP 模式。

指标：`epdg_diameter_s6b_aar_count`

类型：计数器

描述：处理的 AA 请求总数，用于会话授权

标签：

- `result` - 请求结果：`success` 或 `failure`

指标：`epdg_diameter_s6b_str_count`

类型：计数器

描述：处理的会话终止请求总数

指标：`epdg_diameter_s6b_active_sessions_count`

类型：计量器

描述：当前活动的 S6b 会话数量

会话生命周期指标

跟踪 PDN 会话的创建和终止。

指标：`epdg_session_created_count`

类型：计数器

描述：创建的会话总数

标签：

- `vpn_mode` - VPN 模式：`simple` 或 `gtp`

指标：`epdg_session_terminated_count`

类型：计数器

描述：终止的会话总数

标签：

- `reason` - 终止原因：`user_request`，`timeout`，`error`，`admin`

指标：`epdg_sessions_active_count`

类型：计量器

描述：当前活动会话的数量。每 5 秒轮询一次。

指标：`epdg_sessions_by_state_count`

类型：计量器

描述：按 FSM 状态分组的会话

标签：

- `state` - 会话状态（例如，`init`，`eap_identity`，`eap_aka_challenge`，`authenticated`，`established`）

指标：`epdg_auth_duration_ms`

类型：直方图

描述：从 IKE_SA_INIT 到会话建立的完整身份验证流程持续时间

桶：100, 250, 500, 1000, 2500, 5000, 10000 ms

指标：`epdg_session_duration_seconds`

类型：直方图

描述：从建立到终止的会话生命周期

桶：60, 300, 900, 1800, 3600, 7200, 14400 seconds (1 min 到 4 hours)

ESP 数据平面指标

根据 RFC 4303 的 ESP 数据包处理指标。

指标：`epdg_esp_packets_in_count`

类型：计数器

描述：成功解密的 ESP 数据包总数（UE → 网络方向）

指标：`epdg_esp_packets_out_count`

类型：计数器

描述：加密的 ESP 数据包总数（网络 → UE 方向）

指标：`epdg_esp_bytes_in_total`

类型：计量器

描述：从 ESP 数据包解密的总字节数

指标：`epdg_esp_bytes_out_total`

类型：计量器

描述：加密到 ESP 数据包中的总字节数

IP 池指标

用于简单 VPN 模式的 IP 地址池管理指标。

指标：`epdg_ip_allocated_count`

类型：计数器

描述：分配的 IP 地址总数

标签：

- `type` - 地址类型：`ipv4` 或 `ipv6`

指标：`epdg_ip_released_count`

类型：计数器

描述：释放的 IP 地址总数

标签：

- `type` - 地址类型：`ipv4` 或 `ipv6`

指标：`epdg_ip_pool_allocated_count`

类型：计量器

描述：当前分配的 IP 地址数量

指标：`epdg_ip_pool_available_count`

类型：计量器

描述：当前可用的 IP 地址数量

指标：`epdg_ip_pool_utilization_ratio`

类型：计量器

描述：IP 池的利用率，范围从 0.0 到 1.0。接近 1.0 的值表示池耗尽的风险。

VM 指标

用于系统健康监控的 Erlang/BEAM 虚拟机指标。

指标：`vm_memory_total`

类型：计量器

单位：字节

描述：VM 分配的总内存

指标：`vm_memory_processes`

类型：计量器

单位：字节

描述：Erlang 进程使用的内存

指标：`vm_memory_binary`

类型：计量器

单位：字节

描述：用于二进制文件的内存（包括数据包缓冲区）

指标：`vm_memory_ets`

类型：计量器

单位：字节

描述：ETS 表使用的内存（会话状态、注册表）

指标：`vm_system_info_process_count`

类型：计量器

描述：正在运行的 Erlang 进程数量

指标：`vm_system_info_port_count`

类型：计量器

描述：打开的端口数量（套接字、文件句柄）

指标：`vm_statistics_run_queue`

类型：计量器

描述：调度程序运行队列的总长度。高值表示 CPU 饱和。

Prometheus 集成

抓取配置

将 OmniEPDG 添加到您的 Prometheus `prometheus.yml`：

```
scrape_configs:
  - job_name: 'omniepdg'
    scrape_interval: 15s
    static_configs:
      - targets: ['epdg-host:9568']
        labels:
          instance: 'epdg-01'
          environment: 'production'
```

服务发现

对于 Kubernetes 部署，使用服务发现：

```yaml
scrape_configs:
  - job_name: 'omniepdg'
    kubernetes_sd_configs:
      - role: pod
    relabel_configs:
      - source_labels: [__meta_kubernetes_pod_label_app]
        action: keep
        regex: omniepdg
      - source_labels:
[__meta_kubernetes_pod_annotation_prometheus_io_port]
        action: replace
        target_label: __address__
        regex: (.+)
        replacement: ${1}:9568
```

示例查询

身份验证成功率

```
# 5分钟内的成功率
sum(rate(epdg_eap_aka_success_count[5m]))
/
(sum(rate(epdg_eap_aka_success_count[5m])) +
sum(rate(epdg_eap_aka_failure_count[5m])))
```

会话建立率

```
# 每秒建立的会话
rate(epdg_ikev2_session_established_count[5m])
```

身份验证延迟 **(p95)**

```
histogram_quantile(0.95,
sum(rate(epdg_auth_duration_ms_bucket[5m])) by (le))
```

HSS 延迟 (p99)

```
histogram_quantile(0.99,
sum(rate(epdg_diameter_swx_mar_latency_ms_bucket[5m])) by (le))
```

活动会话

```
epdg_sessions_active_count
```

IP 池利用率

```
epdg_ip_pool_utilization_ratio * 100
```

ESP 吞吐量

```
# 每秒字节数 (入站)
rate(epdg_esp_bytes_in_total[5m])

# 每秒数据包数 (双向)
rate(epdg_esp_packets_in_count[5m]) +
rate(epdg_esp_packets_out_count[5m])
```

按原因划分的失败

```
# 按原因划分的 EAP 失败
sum by (reason) (rate(epdg_eap_aka_failure_count[5m]))

# 按原因划分的会话终止
sum by (reason) (rate(epdg_session_terminated_count[5m]))
```

告警规则

OmniEPDG 的示例 Prometheus 告警规则:

```yaml
groups:
  - name: omniepdg
    rules:
      # 高身份验证失败率
      - alert: OmniEPDGHighAuthFailureRate
        expr: |
          sum(rate(epdg_eap_aka_failure_count[5m]))
          /
          (sum(rate(epdg_eap_aka_success_count[5m])) +
sum(rate(epdg_eap_aka_failure_count[5m])))
          > 0.1
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "高 EAP-AKA 身份验证失败率"
          description: "过去 5 分钟身份验证失败率为 {{ $value |
humanizePercentage }}"

      # IP 池接近耗尽
      - alert: OmniEPDGIPPoolLow
        expr: epdg_ip_pool_utilization_ratio > 0.9
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "IP 池利用率超过 90%"
          description: "IP 池利用率为 {{ $value | humanizePercentage
}}"

      # IP 池耗尽
      - alert: OmniEPDGIPPoolExhausted
        expr: epdg_ip_pool_available_count == 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "IP 池耗尽"
          description: "没有可用于新会话的 IP 地址"

      # HSS 延迟高
      - alert: OmniEPDGHSSLatencyHigh
        expr: |
```

```
          histogram_quantile(0.95,
sum(rate(epdg_diameter_swx_mar_latency_ms_bucket[5m])) by (le))
          > 1000
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "高 HSS (SWx) 延迟"
          description: "95 百分位 MAR 延迟为 {{ $value }}ms"

      # 待处理 SWx 请求积压
      - alert: OmniEPDGSWxBacklog
        expr: epdg_diameter_swx_pending_count > 100
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: "SWx 请求积压"
          description: "{{ $value }} 个待处理的 SWx 请求"

      # VM 内存高
      - alert: OmniEPDGMemoryHigh
        expr: vm_memory_total > 2147483648
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "OmniEPDG 内存使用高"
          description: "VM 内存使用为 {{ $value | humanize1024 }}"

      # 调度程序过载
      - alert: OmniEPDGSchedulerOverload
        expr: vm_statistics_run_queue > 10
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Erlang 调度程序运行队列高"
          description: "运行队列长度为 {{ $value }}，表示 CPU 饱和"

      # 没有会话（潜在服务问题）
      - alert: OmniEPDGNoSessions
        expr: epdg_sessions_active_count == 0 and
epdg_ikev2_session_initiated_count > 0
```

```
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "尽管有连接尝试，但没有活动会话"
          description: "会话正在启动，但没有活动"

      # 高速率限制活动 (潜在攻击)
      - alert: OmniEPDGHighRateLimiting
        expr: rate(epdg_auth_rate_limited_count[5m]) > 10
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "被阻止的身份验证尝试□率高"
          description: "{{ $value | printf \"%.1f\" }} 每秒被阻止的身份验证尝试"

      # GeoIP 阻止激增 (特定区域的潜在攻击)
      - alert: OmniEPDGGeoIPBlockingSpike
        expr: |
          rate(epdg_auth_geoip_blocked_count[5m]) > 5
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "GeoIP 阻止活动升高"
          description: "{{ $value | printf \"%.1f\" }} 每秒被 GeoIP 阻止的连接"

      # 检测到 ESP 重放攻击
      - alert: OmniEPDGReplayAttack
        expr: rate(epdg_esp_replay_detected_count[5m]) > 0
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: "检测到 ESP 重放攻击"
          description: "{{ $value | printf \"%.1f\" }} 每秒重放尝试"

      # AUTH 验证失败 (潜在 MITM)
      - alert: OmniEPDGAuthVerificationFailures
        expr: rate(epdg_auth_verification_failed_count[5m]) > 0
        for: 2m
```

```
        labels:
          severity: critical
        annotations:
          summary: "检测到 AUTH 有效负载验证失败"
          description: "潜在的中间人攻击或实现错误"
```

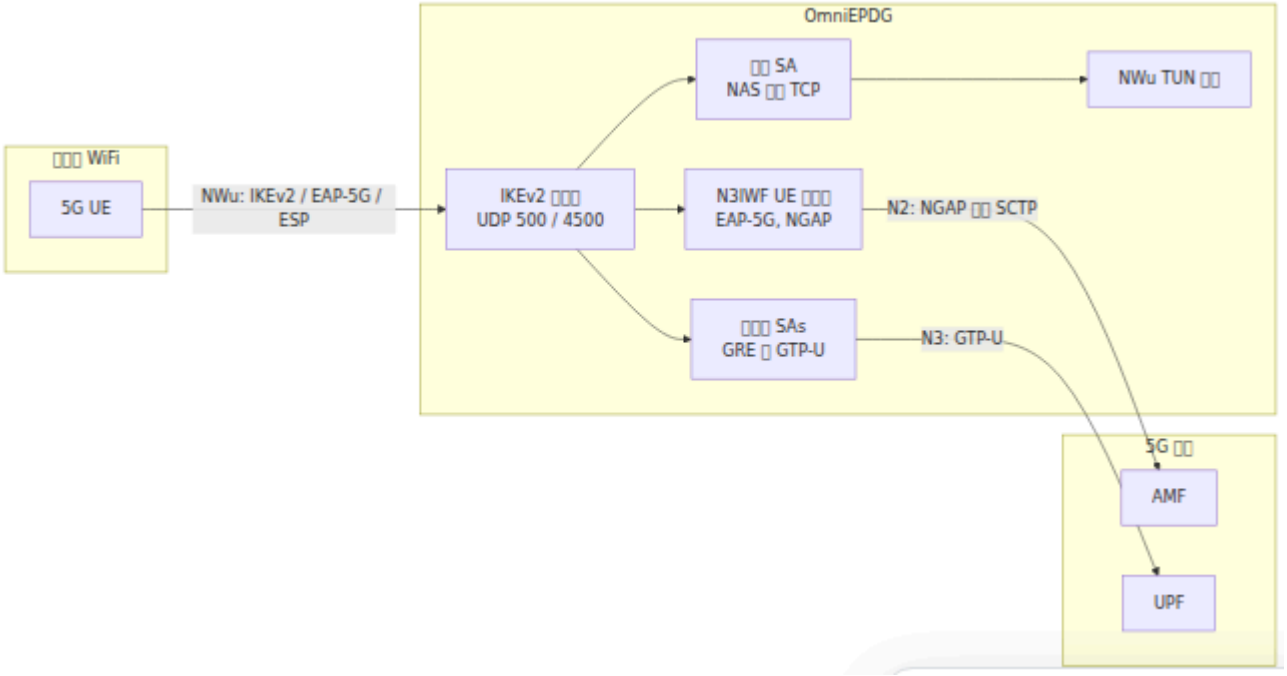
N3IWF 模式（5G 不可信非 3GPP 接入）

OmniEPDG 可以作为 非 **3GPP** 互通功能 **（N3IWF）** 运行。在此模式下，连接到不可信 WiFi 网络的 5G UE 通过 OmniEPDG 连接到 5G 核心 (5GC)。N3IWF 和 ePDG 使用相同的 IKEv2 监听器、相同的 ESP 数据路径和相同的 GTP-U 套接字。

返回 操作指南。

目录

- 架构概述
- OmniEPDG 如何选择 N3IWF 模式
- 接口
- AMF 选择
- 注册流程
- PDU 会话流程
- 释放流程
- 内部 NWu 地址分配
- 配置
- 主机要求
- 指标
- 限制
- 故障排除
- 3GPP 规范



架构概述

N3IWF 在 UE 和 AMF 之间中继 NAS 消息。N3IWF 不读取 NAS 内容。用户数据在 IPsec 子 SA 内的 GRE 数据包中传输。N3IWF 将每个 GRE 数据包转换为 UPF 的 GTP-U 数据包，反向操作用于下行数据。

OmniEPDG 如何选择 N3IWF 模式

OmniEPDG 检查每个 IKE SA 的第一个 IKE_AUTH 请求。

UE 发送	OmniEPDG 行为
类型为 `ID_KEY_ID` 的 IDi 且没有 AUTH 有效负载	会话作为 N3IWF 会话运行 (5G UE)。
类型为 `ID_RFC822_ADDR` (NAI) 的 IDi	会话作为 ePDG 会话运行 (EAP-AKA, SWx, S2b)。

5G UE 必须在 IDi 中发送 `ID_KEY_ID`，如 3GPP TS 33.501 §7.2.1 和 3GPP TS 24.502 §7.3.2.1 所规定。启用 N3IWF 模式时，ePDG 会话不会改变。

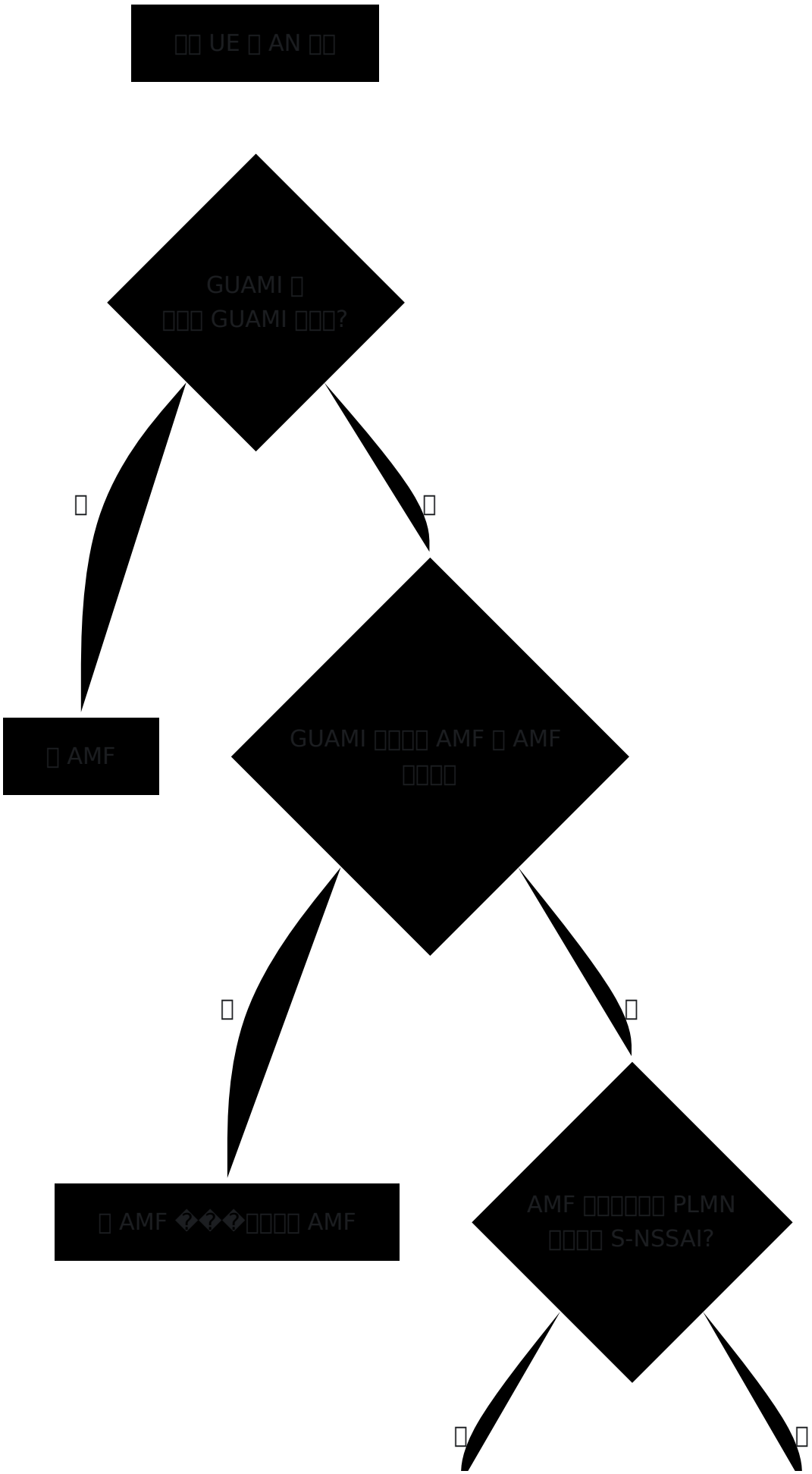
接口

接口	对端	传输	协议	参考
NWu	5G UE	UDP 500 / 4500	IKEv2, EAP-5G, ESP (NAT-T)	TS 24.502, RFC 7296
NWu (内部)	5G UE	TCP 在信令 SA 内	NAS 消息封装	TS 24.502 §8.2
NWu (内部)	5G UE	GRE 在用户面 SAs 内	GRE 带有 QFI 的密钥	TS 24.502 §8.3
N2	AMF	SCTP, PPID 60	NGAP	TS 38.412, TS 38.413
N3	UPF	UDP 2152	GTP-U 带 PDU 会话容器	TS 29.281, TS 38.415

在启动时，OmniEPDG 连接到 `amfs` 中的每个 AMF，并向每个 AMF 发送带有 `GlobalN3IWF-ID` 的 NG 设置请求。当至少一个 AMF 发送 NG 设置响应时，OmniEPDG 接受 5G UE。如果 SCTP 关联失败，OmniEPDG 每 5 秒尝试该 AMF。如果关联断开，OmniEPDG 仅释放该 AMF 上的 N3IWF 会话。

AMF 选择

N3IWF 在第一个 EAP-5G NAS 消息到达时为每个 UE 选择一个 AMF。N3IWF 使用 UE 的 AN 参数 (TS 24.502 §9.3.2.2.2) 和每个 AMF 在其 NG 设置响应中发送的数据。UE 在该 AMF 上保持直到释放。请参见 TS 23.501 §6.3.5。


□□ UE □ AN □□
GUAMI □
□□□ GUAMI □□□?
□
□
□ AMF
GUAMI □□□□ AMF □ AMF
□□□□
□
□
□ AMF ���□□□□ AMF
AMF □□□□□□ PLMN
□□□□ S-NSSAI?
□
□



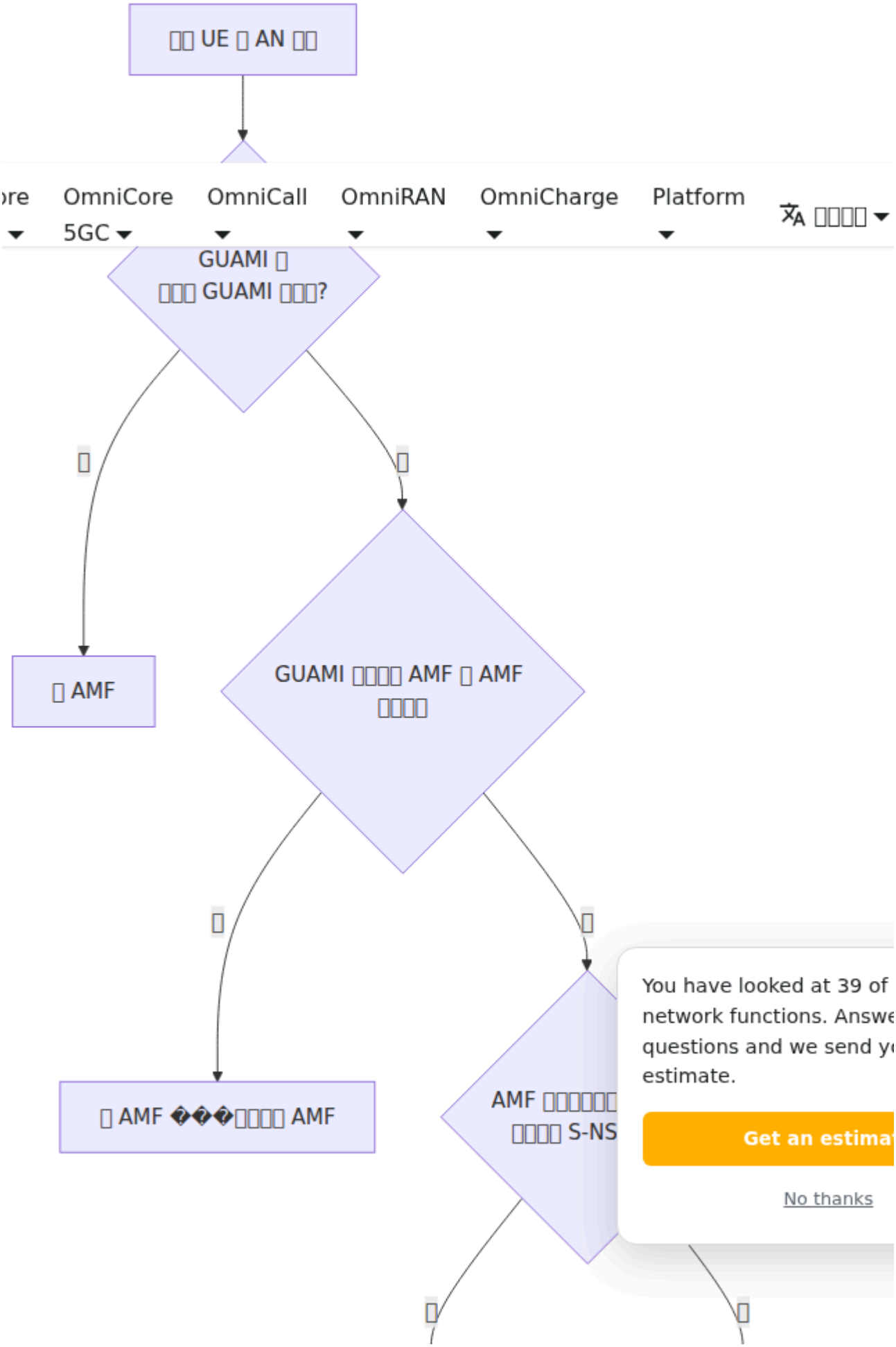
步骤	**AN** 参数	**AMF** 数据来自 **NG** 设置响应	结果
1	GUAMI	服务的 GUAMI 列表	提供此确切 GUAMI 的 AMF。已注册的 UE 返回其 AMF。
2	GUAMI	服务的 GUAMI 列表	具有相同 PLMN、AMF 区域 ID 和 AMF 集 ID 的 AMF。
3	选定的 PLMN ID，请求的 NSSAI	PLMN 支持列表	支持所选 PLMN 和至少一个请求的 S-NSSAI 的 AMF。如果 UE 不发送 PLMN 或 NSSAI，则该检查不适用。

当多个 AMF 匹配某个步骤时，N3IWF 随机选择一个。选择□根据每个 AMF 的相对 AMF 容量加权的。容量为 0 的 AMF 仅在 UE 的确切 GUAMI 匹配时获得 UE，或者当每个匹配的 AMF 的容量为 0 时。

如果没有连接的 AMF 匹配，N3IWF 向 UE 发送 EAP-Failure，并记录 `no AMF serves the UE's GUAMI, PLMN or slices`。

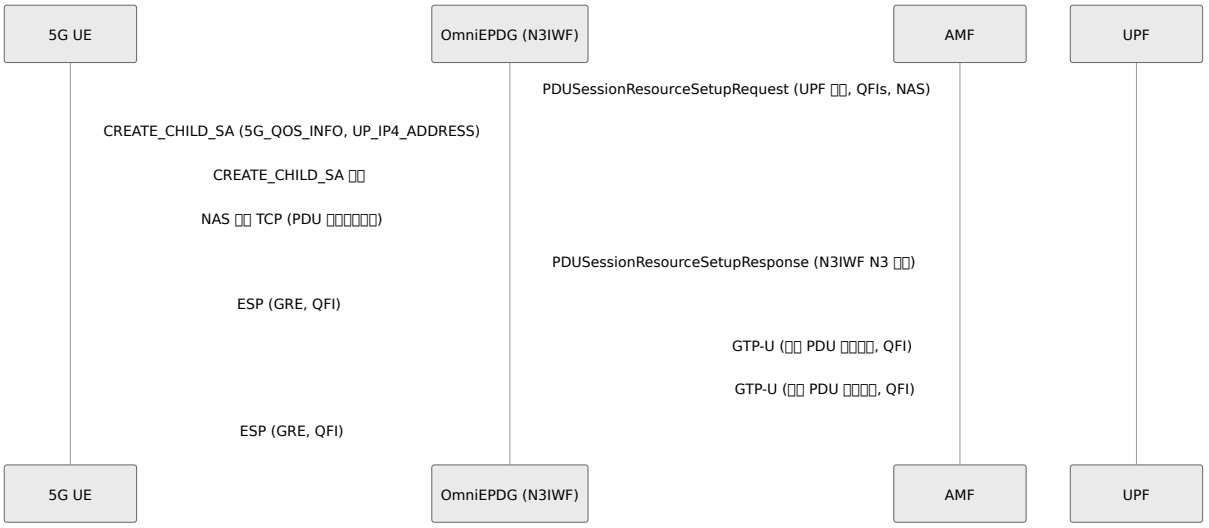
用例：为不同 5G 核心的用户使用一个 N3IWF。将每个核心的 AMF 放入 `amfs` 中。之前注册的 UE 通过 GUAMI 返回到其 AMF。新 UE 前往服务其 PLMN 和切片的 AMF。

□□□□



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks





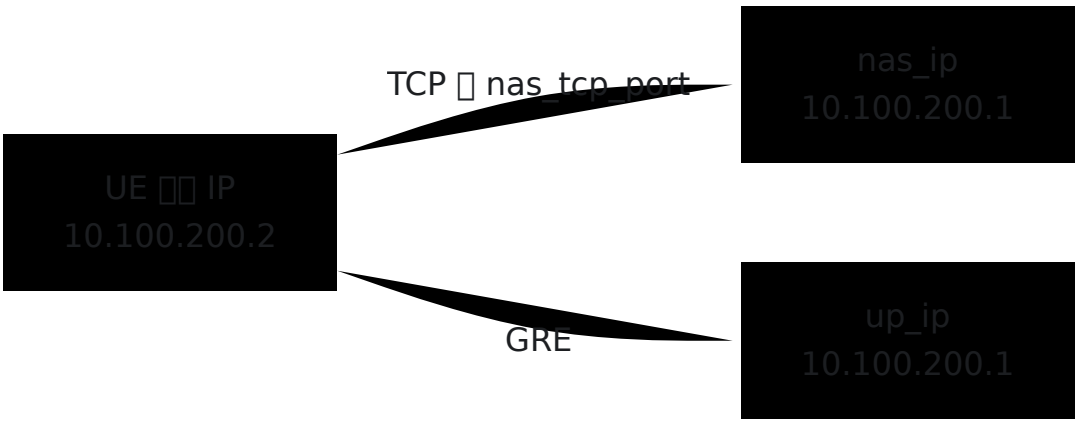
这些要点适用于该流程：

- N3IWF 使用 `key_file` 中的密钥对其第一个 IKE_AUTH 响应进行签名。只有当 UE 发送 CERTREQ 时，N3IWF 才包括其证书。
- N3IWF 不会在 AMF 发送下行 NAS 消息或 N3IWF 密钥之前回应来自 UE 的 EAP-5G 请求。
- UE 和 N3IWF 使用 **Kn3iwf** 从 InitialContextSetupRequest 作为 EAP MSK。两个对等方从此密钥计算其最终 AUTH 有效负载。
- 如果 AMF 在 UE 打开 TCP 连接之前发送下行 NAS，N3IWF 会保留 NAS 消息，并在连接打开时发送。
- 如果 UE 发送 EAP-响应/5G-停止，N3IWF 发送 EAP-Failure 并移除 IKE SA。

PDU 会话流程

N3IWF 为每个 PDU 会话创建一个用户面子 SA。此子 SA □默认子 SA，承载 PDU 会话的所有 QoS 流。5G_QOS_INFO 通知有效负载告诉 UE PDU 会话 ID 和 QFIs。

如果 UE 拒绝子 SA，或在 5 次传输后没有响应，N3IWF 会在 PDUSessionResourceSetupResponse 中报告 PDU 会话失败。



释放流程

触发	**N3IWF** 行动
来自 AMF 的 UEContextReleaseCommand	N3IWF 删除 IKE SA，关闭 NAS 连接并发送 UEContextReleaseComplete。
UE 删除 IKE SA	N3IWF 向 AMF 发送 UEContextReleaseRequest。
来自 AMF 的 PDUSessionResourceReleaseCommand	N3IWF 删除每个 PDU 会话的子 SA 并发送 PDUSessionResourceReleaseRespons
来自 AMF 的 NG 重置	N3IWF 释放指定的 UE（或所有 UE）并发送 NGResetAcknowledge。
N2 关联丢失	N3IWF 释放所有 N3IWF 会话。
会话不活动	N3IWF 在 `session_inactivity_timeout_ms` 内 IKE 或 ESP 流量后释放会话。请参见 配置参考。

内部 **NWu** 地址分配

N3IWF 为每个 UE 提供一个内部 **IP** 地址，来自 `ue_ip_pool`。UE 在其 IPsec 隧道内部使用此地址。

- 信令 **SA**：UE 打开到 `nas_ip` 和 `nas_tcp_port` 的 TCP 连接。N3IWF 将这些数据包写入 NWu TUN 接口，因此 Linux TCP 栈终止连接。
- 用户面 **SAs**：UE 向 `up_ip` 发送 GRE 数据包。OmniEPDG 在用户空间解封 GRE。内核无法看到这些数据包。

该池不会将 `nas_ip` 或 `up_ip` 分配给 UE。`nas_ip` 和 `up_ip` 可以相同的地址。

配置

N3IWF 模式默认关闭。在 `runtime.exs` 中设置参数。

```
config :omniepdg, :n3iwf,
  enabled: true,
  # NG 设置的 GlobalN3IWF-ID 和服务 TA
  n3iwf_id: 1,
  name: "OmniN3IWF",
  plmn: {"001", "01"},
  tac: 1,
  slices: [%{sst: 1, sd: nil}],
  # N2 (SCTP) 到每个 AMF
  amfs: [
    %{name: "amf1", ip: {10, 74, 0, 40}, port: 38412},
    %{name: "amf2", ip: {10, 74, 0, 41}, port: 38412}
  ],
  ngap_local_ip: {10, 74, 0, 13},
  # 发送给 UE 的 IDr 及签署 IKE_AUTH 的证书/密钥
  identity: "n3iwf.5gc.mnc001.mcc001.3gppnetwork.org",
  cert_file: "/etc/omniepdg/certs/n3iwf.crt",
  key_file: "/etc/omniepdg/certs/n3iwf.key",
  # 广告给 UPF 的 N3 GTP-U 地址
  n3_ip: {10, 74, 0, 13},
  # 内部 NWu 地址分配
  ue_ip_pool: "10.100.200.0/24",
  nas_ip: {10, 100, 200, 1},
  up_ip: {10, 100, 200, 1},
  nas_tcp_port: 20000,
  tun_name: "omnin3iwf0"
```

当 `enabled` 为 `true` 时，所有参数都必需的。如果缺少参数或参数无效，OmniEPDG 不会启动。启动错误会命名参数。

参数

参数	类型	必需	默认	描述
`enabled`	布尔值	□	`false`	开启或关闭 N3IWF 模式。当为 `false` 时，OmniEPDG 忽略所有其他 N3IWF 参数。
`n3iwf_id`	整数	□	-	NG 设置请求中的 `GlobalN3IWF-ID` 的 N3IWF ID。范围 0 到 65535 (16 位)。请参见 TS 38.413 §9.3.1.57。
`name`	字符串	□	-	NG 设置请求中的 RAN 节点名称。AMF 为 N3IWF 显示此名称。
`plmn`	元组	□	-	N3IWF 的 PLMN，格式为 `{mcc, mnc}` 数字字符串。MNC 具有 2 或 3 位数字。AMF 必须服务此 PLMN。
`tac`	整数	□	-	NG 设置请求的支持 TA 列表中的跟踪区域代码 (24 位)。AMF 必须服务此 TAC。
`slices`	列表	□	-	N3IWF 支持的 S-NSSAIs。每个条目为 `%{sst: 0..255, sd: nil}` 或 `%{sst: 0..255, sd: <<b1, b2, b3>>}`。
`amfs`	列表	□	-	通过 N2 连接的 AMF。每个条目为 `%{name: 字符串, ip: IPv4 元组, port: 整数}`。`name` □日志的唯一标签。`port` □ AMF N2 接口的 SCTP 端口。标准端口为 38412 (TS 38.412)。请参见 AMF 选择。
`ngap_local_ip`	元组	□	-	SCTP 关联到 AMF 的本地 IPv4 地址。
`identity`	字符	□	-	N3IWF 身份 (FQDN) 在 IDr 有效负载中。UE 将此值与其选择的 N3IWF 进行比较。使用证书中的 FQDN。

参数	类型	必需	默认	描述
	串			
`cert_file`	字符串	□	-	N3IWF 证书 (PEM) 的路径。N3IWF 在 UE 发送 CERTREQ 时发送此证书。
`key_file`	字符串	□	-	`cert_file` 的私钥路径 (PEM，未加密)。RSA 密钥使用 AUTH 方法 1 签名。ECDSA P-256 密钥使用 AUTH 方法 9 签名。
`n3_ip`	元组	□	-	N3 上 N3IWF 提供给 AMF 的 IPv4 地址，用于下行 GTP-U。UPF 将下行数据发送到此地址的 UDP 2152。
`ue_ip_pool`	字符串	□	-	UE 内部地址的 IPv4 CIDR。前缀长度必须为 8 到 30。
`nas_ip`	元组	□	-	NAS 通过 TCP 的内部 IPv4 地址。N3IWF 在 NAS_IP4_ADDRESS 通知有效负载中发送它，并将其分配给 NWu TUN 接口。使用 `ue_ip_pool` 内的地址。
`up_ip`	元组	□	-	GRE 用户数据的内部 IPv4 地址。N3IWF 在 UP_IP4_ADDRESS 通知有效负载中发送它。
`nas_tcp_port`	整数	□	-	NAS 传输的 TCP 端口。N3IWF 在 NAS_TCP_PORT 通知有效负载中发送它。
`tun_name`	字符串	□	-	NWu TUN 接口的名称。使用没有其他进程使用的名称。

N3IWF 使用 IKEv2 监听器设置 (`listen_ip`, `port_500`, `port_4500`) 和主配置中的 `session_inactivity_timeout_ms`。请参见 配置参考。

配置示例：**ePDG** 和 **N3IWF** 在同一主机上

```
config :omniepdg,
  listen_ip: {0, 0, 0, 0},
  port_500: 500,
  port_4500: 4500,
  vpn_mode: :gtp

config :omniepdg, :n3iwf,
  enabled: true,
  n3iwf_id: 1,
  name: "OmniN3IWF",
  plmn: {"001", "01"},
  tac: 1,
  slices: [%{sst: 1, sd: nil}],
  amfs: [%{name: "amf1", ip: {10, 74, 0, 40}, port: 38412}],
  ngap_local_ip: {10, 74, 0, 13},
  identity: "n3iwf.5gc.mnc001.mcc001.3gppnetwork.org",
  cert_file: "/etc/omniepdg/certs/n3iwf.crt",
  key_file: "/etc/omniepdg/certs/n3iwf.key",
  n3_ip: {10, 74, 0, 13},
  ue_ip_pool: "10.100.200.0/24",
  nas_ip: {10, 100, 200, 1},
  up_ip: {10, 100, 200, 1},
  nas_tcp_port: 20000,
  tun_name: "omnin3iwf0"
```

工作原理：4G UE (NAI 身份) 使用 ePDG 流和通过 S2b 的 PGW。5G UE (`ID_KEY_ID` 身份) 使用 N3IWF 流和通过 N2 的 AMF。两个流共享 UDP 500、UDP 4500 和 UDP 2152 上的 GTP-U 套接字。

用例：一个 WiFi 呼叫网关为 4G 和 5G 用户提供服务。

第一个块中的参数在 配置参考 中。第二个块中的参数在 参数 中。

主机要求

- OmniEPDG 必须有权限创建 NWu TUN 接口，并运行 `ip addr` 和 `ip link`（`CAP_NET_ADMIN`）。如果 OmniEPDG 无法创建接口，则不会启动。
- 内核必须支持 SCTP（`sctp` 模块）。

- 允许从 `ngap_local_ip` 到 `amfs` 中每个 AMF 的 SCTP 连接。
- 允许 UDP 2152 在 `n3_ip` 和 UPF 之间。
- UPF 必须将下行流量路由到 `n3_ip`。

指标

N3IWF 会话使用与 ePDG 会话相同的会话指标。请参见 指标参考。

指标：`epdg_session_created_count`
类型：计数器
描述：创建的总会话数。N3IWF 会话的标签值为 `vpn_mode="n3iwf"`。
标签：

- `vpn_mode` - `simple`、`gtp` 或 `n3iwf`

示例查询：

```
# N3IWF 每秒注册数
rate(epdg_session_created_count{vpn_mode="n3iwf"}[5m])
```

控制面板的会话页面显示带有标签 `n3iwf-<RAN-UE-NGAP-ID>` 的 N3IWF 会话。N3IWF 不知道 SUPI，因为 NAS 消息在 UE 和 AMF 之间加密的。

限制

- 内部地址、NAS 传输和 N3 仅支持 IPv4。
- N3IWF 在 NG 设置请求中向每个 AMF 发送相同的 `plmn`、`tac` 和 `slices`。`amfs` 中的每个 AMF 必须服务该 PLMN 和 TAC。
- N3IWF 仅从 NG 设置响应中读取服务的 GUAMI 列表、PLMN 支持列表和相对 AMF 容量。它不应用 AMF 配置更新。
- 每个 PDU 会话一个用户面子 SA。N3IWF 不为单独的 QoS 流创建额外的子 SA。
- N3IWF 不接受来自 UE 的 CREATE_CHILD_SA 请求。它以 NO_ADDITIONAL_SAS 回应，因此不支持 UE 发起的重新密钥。
- N3IWF 记录 PDUSessionResourceModifyRequest，但不对此采取行动。

故障排除

5G UE 无法连接

症状：UE 发送 IKE_AUTH 并收到 AUTHENTICATION_FAILED。日志显示 `n2_not_ready`。

可能原因：

- 没有与 AMF 的 SCTP 关联处于活动状态。
- AMF 拒绝了 NG 设置请求。

解决方案：

1. 在日志中查找 `N2 SCTP connect to AMF failed` 或 `NG Setup rejected by AMF`。`amf` 日志字段命名 `amfs` 中的条目。
2. 确保 `amfs` 中每个条目的 `ip` 和 `port` 正确。
3. 确保每个 AMF 服务 `plmn` 和 `tac` 值。

UE 在发送第一个 EAP-5G 消息后收到 EAP-Failure

症状：UE 在发送注册请求后收到 EAP-Failure。日志显示 `no AMF serves the UE's GUAMI, PLMN or slices`。

可能原因：

- 没有连接的 AMF 服务于 UE 的 GUAMI、所选 PLMN 或请求的 S-NSSAI。

解决方案：

1. 将日志字段中的 `guami`、`selected_plmn` 和 `requested_nssai` 与每个 AMF 的 PLMN 支持列表进行比较。
2. 确保该核心的 AMF 在 `amfs` 中，并且其关联处于活动状态。

UE 拒绝 N3IWF 证书

症状：UE 在第一个 IKE_AUTH 响应后停止。

可能原因：

- UE 不信任 `cert_file` 中的证书。
- 证书 FQDN 与 `identity` 不同。

解决方案：

1. 在 UE 上安装 `cert_file` 的根 CA。
2. 确保 `identity` 证书中的 FQDN。

UE AUTH 验证在 EAP-Success 后失败

症状：日志显示 `N3IWF: UE AUTH verification failed`。

可能原因：

- AMF 使用错误的接入类型区分符派生 Kn3iwf。对于非 3GPP 接入，区分符为 `0x02` (TS 33.501 附录 A.9)。
- AMF 使用的 NAS 上行 COUNT 与 UE 值不同。

解决方案：

1. 确保 AMF 为 Kn3iwf 使用非 3GPP 接入区分符。
2. 确保 AMF 使用非 3GPP NAS 连接的 NAS COUNT。

注册接受未到达 UE

症状：IPsec SA 已建立。AMF 发送 DownlinkNASTransport，但 UE 未注册。

可能原因：

- UE 无法打开到 `nas_ip` 和 `nas_tcp_port` 的 TCP 连接。
- NWu TUN 接口已关闭。

解决方案：

1. 确保 TUN 接口 `tun_name` 已启动并具有 `nas_ip`。
2. 确保没有防火墙规则阻止 TUN 接口上的 TCP 到 `nas_tcp_port`。

PDU 会话设置后没有用户数据

症状：PDU 会话已建立，但 UE 没有数据。

可能原因：

- UPF 无法通过 UDP 2152 到达 `n3_ip`。
- 防火墙阻止 N3IWF 和 UPF 之间的 GTP-U。

解决方案：

1. 确保 UPF 将下行 GTP-U 发送到 `n3_ip`。
2. 确保 UDP 2152 在两个方向上都□开放的。

3GPP 规范

规范	标题
TS 23.501	5G 系统���系统架构
TS 23.502	5G 系统的程序 (§4.12: 不可信非 3GPP 接入)
TS 24.502	通过非 3GPP 接入网络访问 3GPP 5GC (EAP-5G, NWu, GRE)
TS 33.501	5G 系统的安全架构和程序 (§7.2.1, 附录 A.9)
TS 38.412	NG 信令传输 (SCTP)
TS 38.413	NG 应用协议 (NGAP)
TS 38.415	PDU 会话用户面协议
TS 29.281	GTP-U
RFC 7296	IKEv2

网络要求

本文档涵盖了部署 OmniEPDG 进行 WiFi 通话所需的防火墙端口和 DNS 条目。

防火墙端口

面向互联网的端口 (UE ↔ ePDG)

这些端口必须对互联网开放，以便移动设备建立 WiFi 通话连接。

端口	协议	方向	目的
500	UDP	入站	IKEv2 初始交换 (IKE_SA_INIT, IKE_AUTH)
4500	UDP	入站	IKEv2 NAT 穿越和 ESP-in-UDP 封装

端口 **500/UDP** 处理初始 IKEv2 协商。如果在 UE 和 ePDG 之间检测到 NAT（这几乎总□ WiFi 通话的情况），连接会自动迁移到端口 4500。

端口 **4500/UDP** 在 NAT-T 活动时承载 IKEv2 信令和 ESP 加密的用户数据。这□所有 WiFi 通话流量的主要数据路径。

内部网络端口 (ePDG ↔ 核心)

这些端口用于 OmniEPDG 和移动核心网络之间的通信。它们应该可以从 ePDG 访问，但不应暴露于互联网。

端口	协议	方向	目的	对端
3868	TCP	双向	Diameter SWx (认证)	HSS / DRA
3868	TCP	双向	Diameter S6b (会话授权)	PGW / AAA
2123	UDP	双向	GTPv2-C 控制平面 (S2b)	PGW
2152	UDP	双向	GTP-U 用户平面 (S2b-U)	PGW

管理端口

这些端口用于操作监控，应限制在管理网络中。

端口	协议	目的
4000	TCP	控制面板网页界面 (HTTP)
443	TCP	控制面板网页界面 (HTTPS, 生产)
9568	TCP	Prometheus 指标端点

DNS 要求

ePDG 发现 DNS 记录

移动设备使用 3GPP TS 23.003 中定义的标准化 DNS 命名约定发现 ePDG。FQDN 格式为：

```
epdg.epc.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org
```

其中：

- `<MCC>` — 3 位数的移动国家代码（例如，澳大利亚为 `505`）
- `<MNC>` — 2 或 3 位数的移动网络代码，零填充到 3 位（例如，`001`）

DNS 记录应为 **A** 记录（或 IPv6 的 AAAA），指向 OmniEPDG 的公共 IP 地址。

```
epdg.epc.mnc999.mcc999.pub.3gppnetwork.org.  IN  A  203.0.113.10
```

证书通用名称

ePDG 的 TLS 证书必须具有与设备将连接的 ePDG FQDN 匹配的主题备用名称 (SAN)。在 IKEv2 认证期间，UE 会验证这一点。

证书要求：

- SAN 必须包含 ePDG 发现 FQDN（例如，`epdg.epc.mnc001.mcc001.pub.3gppnetwork.org`）
- 证书必须由受信任的 CA 签名（设备验证链）
- 最小 RSA 2048 位或 ECDSA P-256

Diameter 域 DNS

为了正确的 Diameter 路由，域应通过 DNS NAPTR/SRV 记录解析，符合 RFC 6408。域格式为：

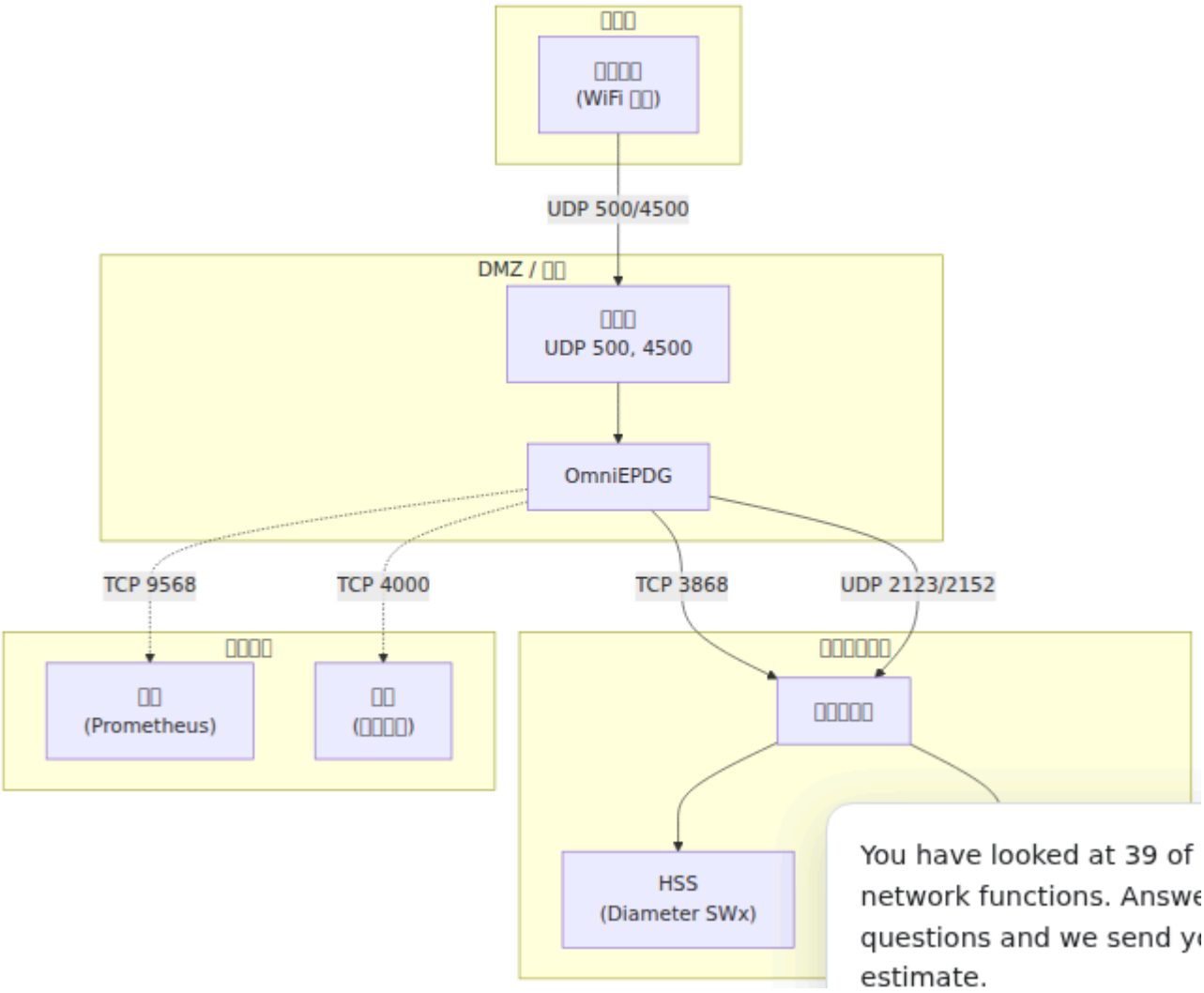
```
epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
```

示例：

```
epc.mnc001.mcc001.3gppnetwork.org
```

该域用于：

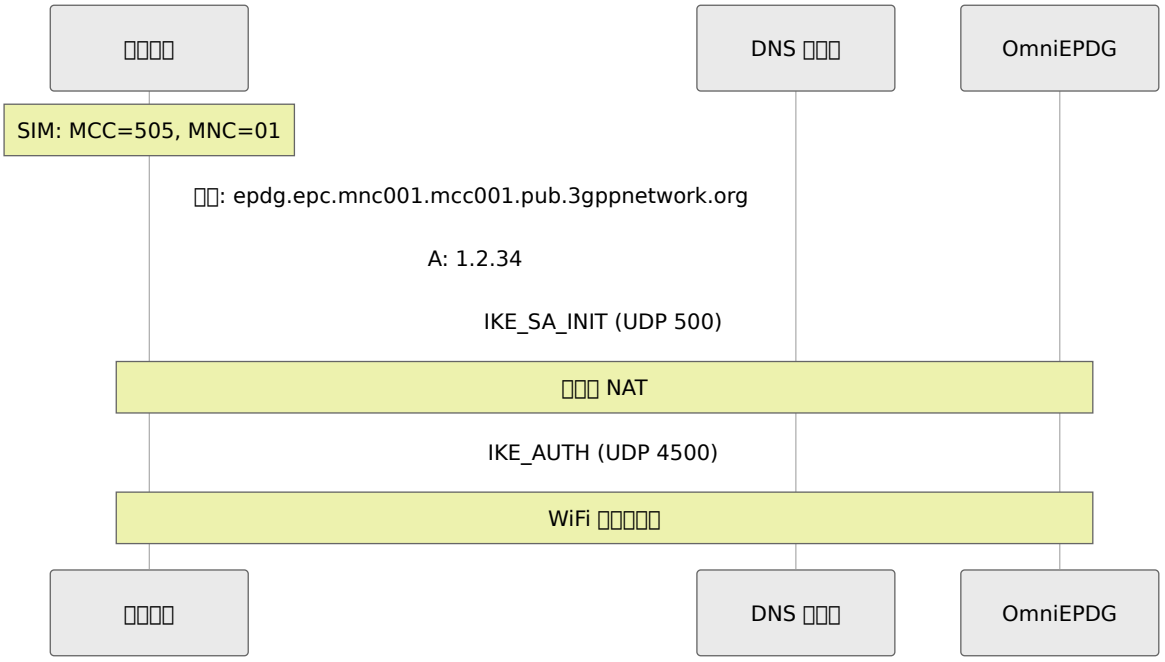
- UE 在认证期间发送的 NAI（网络接入标识符）
- Diameter Origin-Realm 和 Destination-Realm AVP
- Diameter 路由决策



网络拓扑

DNS 查找流程

当移动设备发起 WiFi 通话时，发生以下 DNS 解析：



移动设备
DNS 服务器
OmniEPDG
SIM: MCC=505, MNC=01
查询: epdg.epc.mnc001.mcc001.pub.3gppnetwork.org
A: 1.2.34
IKE_SA_INIT (UDP 500)
检测到 NAT
IKE_AUTH (UDP 4500)
WiFi 通话已建立
移动设备
DNS 服务器
OmniEPDG

清单

面向互联网的要求

□ UDP 500 入站开放至 OmniEPDG

□ UDP 4500 入站开放至 OmniEPDG

□ DNS A 记录: `epdg.epc.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org` → ePDG 公共 IP

□ 安装具有匹配 SAN 的 TLS 证书

核心网络要求

□ TCP 3868 在 OmniEPDG 和 HSS/DRA 之间开放

□ TCP 3868 在 OmniEPDG 和 PGW/AAA 之间开放（仅限 GTP 模式）

□ UDP 2123 在 OmniEPDG 和 PGW 之间开放（仅限 GTP 模式）

□ UDP 2152 在 OmniEPDG 和 PGW 之间开放（仅限 GTP 模式）

管理要求

□ TCP 4000/443 可从操作网络访问

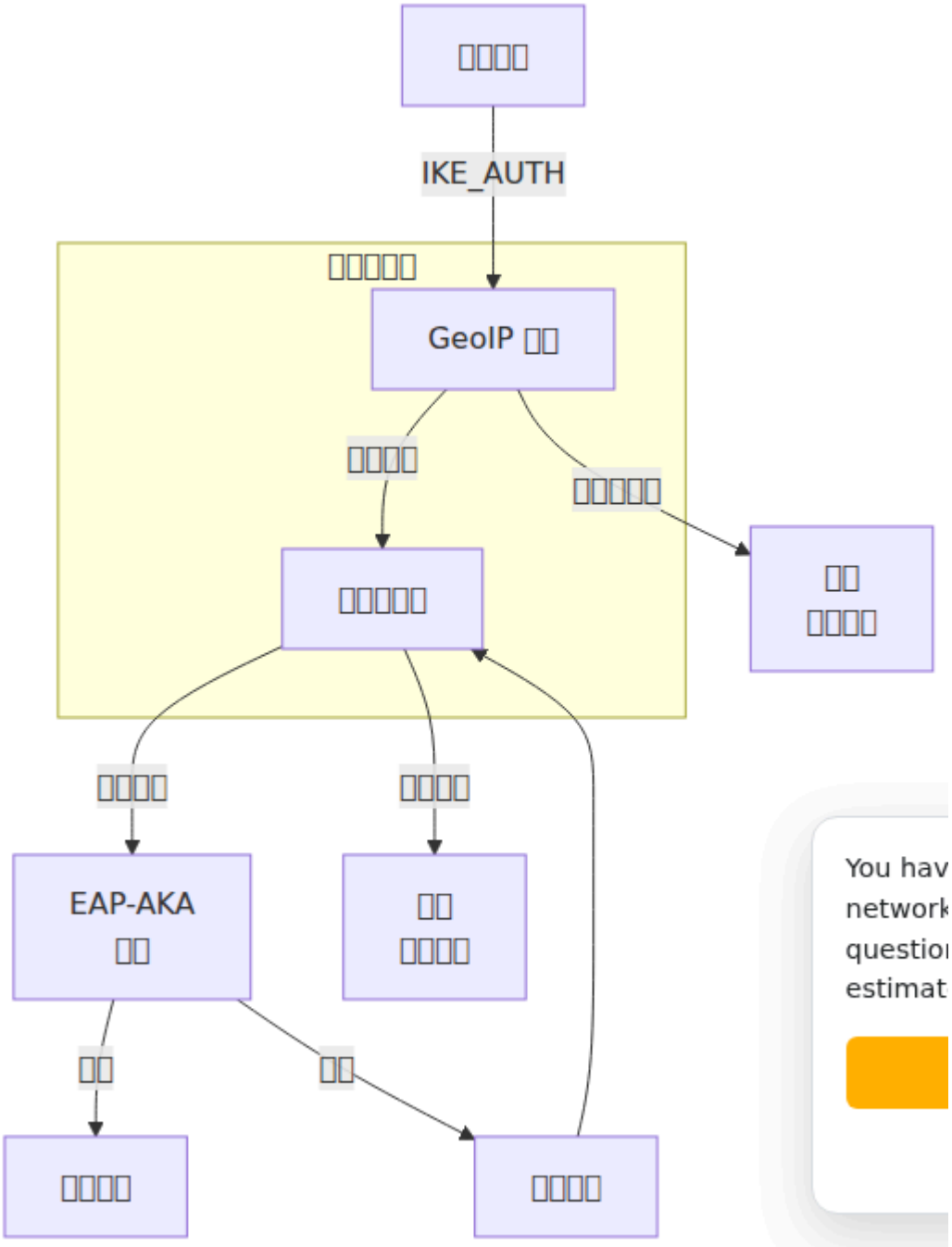
- [ ] TCP 9568 可从监控基础设施访问

参考文献

- 3GPP TS 23.003 - 编号、地址和标识 (ePDG FQDN 格式)
- 3GPP TS 23.402 - 非 3GPP 接入的架构增强
- RFC 7296 - IKEv2 协议
- RFC 3948 - IPsec ESP 数据包的 UDP 封装 (NAT-T)
- RFC 6733 - Diameter 基础协议

OmniEPDG □□□□

OmniEPDG □□□□



概述

OmniEPDG 在 IKE_AUTH 交换开始时执行安全检查，在昂贵的加密操作之前：

1. **GeoIP** 检查（可选） - 验证源 IP 是否来自允许的国家
2. 速率限制检查 - 确保 IP/IMSI 未超过失败阈值
3. **EAP-AKA** 认证 - 如果检查通过，则进行标准 3GPP 认证

认证速率限制

速率限制通过跟踪每个源 IP 和每个 IMSI 的失败认证尝试来防止暴力攻击。当超过阈值时，进一步的尝试会被暂时阻止。

工作原理

速率限制器使用 滑动窗口算法：

- 每个失败的尝试都会记录时间戳
- 超过配置窗口的尝试会自动过期
- 当窗口中的失败超过阈值时，源会被阻止
- 阻止在配置的冷却期后过期

双重跟踪

同时强制执行两个独立的限制：

跟踪类型	目的	默认阈值	默认阻止
每个 **IP**	捕获端口扫描器和来自单一源的分布式攻击	10 次失败 / 分钟	5 分钟
每个 **IMSI**	捕获针对特定用户的攻击	5 次失败 / 分钟	10 分钟

两个检查都必须通过，认证尝试才能继续。如果任一阈值被超出，尝试将被拒绝。

配置

```
config :omniepdg,
  # 每个 IP 的□率限制
  auth_rate_limit_per_ip: 10,              # 阻止前的最大失败次数
  auth_rate_limit_ip_window_ms: 60_000,  # 窗口大小（1 分钟）
  auth_rate_limit_ip_block_ms: 300_000,  # 阻止持续时间（5 分钟）

  # 每个 IMSI 的□率限制
  auth_rate_limit_per_imsi: 5,               # 阻止前的最大失败次数
  auth_rate_limit_imsi_window_ms: 60_000,  # 窗口大小（1 分钟）
  auth_rate_limit_imsi_block_ms: 600_000   # 阻止持续时间（10 分钟）
```

每个 IP 参数

参数	类型	必需	默认	描述
`auth_rate_limit_per_ip`	整数	否	`10`	在阻止之前，单个 IP 地址在窗口期内允许的最大失败认证尝试次数。
`auth_rate_limit_ip_window_ms`	整数	否	`60000`	用于计数 IP 失败的滑动窗口大小（以毫秒为单位）。超过此时间的失败不计入。
`auth_rate_limit_ip_block_ms`	整数	否	`300000`	超过阈��后阻止 IP 的持续时间（以毫秒为单位）。默认□ 5 分钟。

每个 **IMSI** 参数

参数	类型	必需	默认	描述
`auth_rate_limit_per_imsi`	整数	否	`5`	在阻止之前，单个 IMSI 在窗口期内允许的最大失败认证尝试次数。低于每个 IP 以捕获针对性攻击。
`auth_rate_limit_imsi_window_ms`	整数	否	`60000`	用于计数 IMSI 失败的滑动窗口大小（以毫秒为单位）。
`auth_rate_limit_imsi_block_ms`	整数	否	`600000`	超过阈值后阻止 IMSI 的持续时间（以毫秒为单位）。默认 10 分钟（比 IP 阻止时间长，以保护特定用户）。

成功时的行为

当认证成功时，速率限制器会清除该 IP/IMSI 对的所有失败历史。这允许经历瞬时失败（例如网络问题）的合法用户在不被永久惩罚的情况下恢复。

示例配置

高安全性环境

对失败尝试容忍度低的环境实施严格限制：

```
config :omniepdg,
  auth_rate_limit_per_ip: 5,
  auth_rate_limit_ip_window_ms: 120_000,    # 2 分钟窗口
  auth_rate_limit_ip_block_ms: 900_000,     # 15 分钟阻止

  auth_rate_limit_per_imsi: 3,
  auth_rate_limit_imsi_window_ms: 120_000,
  auth_rate_limit_imsi_block_ms: 1_800_000  # 30 分钟阻止
```

工作原理：每个 IP 允许 5 次失败或每个 IMSI 允许 3 次失败，时间限制为 2 分钟。阻止持续时间分别为 15-30 分钟。

使用案例：企业部署、高价值用户群或处于主动攻击中的网络。

放宽环境

对开发或测试实施更宽松的限制：

```
config :omniepdg,
  auth_rate_limit_per_ip: 50,
  auth_rate_limit_ip_window_ms: 60_000,
  auth_rate_limit_ip_block_ms: 60_000,     # 1 分钟阻止

  auth_rate_limit_per_imsi: 20,
  auth_rate_limit_imsi_window_ms: 60_000,
  auth_rate_limit_imsi_block_ms: 120_000   # 2 分钟阻止
```

工作原理：更高的阈值和更短的阻止时间允许更多的测试灵活性。

使用案例：开发环境、集成测试。

GeoIP 国家阻止

GeoIP 阻止根据连接 IP 地址的地理位置限制 WiFi 通话访问。这对于需要根据法规或商业原因将服务限制在特定国家的运营商非常有用。

概述

IP 地理位置数据库

OmniEPDG 使用 **DB-IP IP-to-City Lite** 数据库（CC BY 4.0。参见 `priv/geoip/ATTRIBUTION.md`）。它未提交到 git（约 125 MB）。相反，它在构建时下载。`mix release` 在发布步骤中运行 `geoip.fetch` 任务，因此每个包都带有当前数据库。（开发/测试时可通过 `mix test` 别名按需获取。）因此：

- 每个会话都自动注释国家、城市和大致坐标。Web UI（“位置”列）和 `/api/sessions` 响应显示这些：`country_code`、`country_name`、`continent_code`、`geo_city`、`geo_latitude`、`geo_longitude`。
- `geoip_enabled` 仅控制□否按国家 □□ 连接。注释无论此标志如何都会发生。

使用您自己的数据库（可选）。将 `geoip_database_path` 设置为 MaxMind GeoLite2（`https://www.maxmind.com/en/geolite2/signup`）或其他 DB-IP/MaxMind MMDB。如果配置的文件丢失，则使用捆绑的（下载的）数据库。注意：MaxMind GeoLite2 的最终用户许可协议不允许在产品中重新分发，因此使用 DB-IP Lite（CC BY 4.0）。

要��动刷新数据库（DB-IP 每月发布）：

```
mix geoip.fetch
```

配置

```
config :omniepdg,
  # 按国家阻止连接（地理位置注释始终开启）
  geoip_enabled: true,

  # 可选：用您自己的 MMDB（国家或城市）覆盖捆绑的数据库
  # geoip_database_path: "/etc/omniepdg/GeoLite2-Country.mmdb",

  # 访问控制模式
  geoip_mode: :whitelist,

  # 国家列表（ISO 3166-1 alpha-2 代码）
  geoip_countries: ["AU", "NZ"],

  # 处理未知 IP
  geoip_allow_unknown: false,

  # 数据库不可用时的行为
  geoip_fail_open: true
```

参数

参数	类型	必需	默认	描述
`geoip_enabled`	布尔	否	`false`	启用或禁用基于 GeoIP 的访问 □□（阻止）。当 `false` 时，所有 IP 都被允许，但会话仍然会进行地理注释。
`geoip_database_path`	字符串	否	捆绑的 `dbip-country-lite.mmdb`	MMDB 格式数据库（国家或城市）的路径。如果未设置或缺失，则回退到捆绑的 DB-IP 数据库。
`geoip_mode`	原子	否	`:whitelist`	访问控制模式：`:whitelist`（仅允许列出的国家）或 `:blacklist`（阻止列出的国家，允许其他国家）。
`geoip_countries`	列表	否	`[]`	ISO 3166-1 alpha-2 国家代码的��表。解释取决于 `geoip_mode`。
`geoip_allow_unknown`	布尔	否	见下文	允许来自无法进行地理定位的 IP（私有 IP、未知范围）的连接。白名单模式下默认为 `false`，黑名单模式下默认为 `true`。
`geoip_fail_open`	布尔	否	`true`	GeoIP 数据库加载失败时的行为。当 `true` 时，允许所有连接（失败开放）。当 `false` 时，阻止所有连接（失败关闭）。

访问控制模式

白名单模式（推荐用于 **WiFi** 通话）

仅允许来自指定国家的连接。所有其他国家均被阻止。

```
config :omniepdg,
  geoip_enabled: true,
  geoip_mode: :whitelist,
  geoip_countries: ["AU", "NZ", "FJ"]  # 澳大利亚、新西兰、斐济
```

工作原理：仅允许来自澳大利亚、新西兰或斐济 IP 地址的用户设备进行认证。所有其他国家均被拒绝。

使用案例：希望将 WiFi 通话限制在其许可服务区域的运营商。

黑名单模式

阻止来自指定国家的连接。所有其他国家均被允许。

```
config :omniepdg,
  geoip_enabled: true,
  geoip_mode: :blacklist,
  geoip_countries: ["CN", "RU", "KP", "IR"]  # 中国、俄罗斯、朝鲜、伊朗
```

工作原理：来自列出国家的用户设备被拒绝。所有其他国家均可进行认证。

使用案例：阻止高风险地区，同时允许全球漫游。

处理未知国家

某些 IP 地址无法进行地理定位：

- 私有 IP 范围（10.x.x.x、192.168.x.x 等）
- 数据库中尚未分配的新 IP 块
- Tor 出口节点和某些 VPN

`geoip_allow_unknown` 参数控制行为：

模式	**`geoip_allow_unknown`** 默认值	行为
白名单	`false`	未知 = 不在白名单中 = 被阻止
黑名单	`true`	未知 = 不在黑名单中 = 被允许

要覆盖默认值:

```
config :omniepdg,
  geoip_mode: :whitelist,
  geoip_allow_unknown: true  # 即使在白名单模式下也允许未知 IP
```

数据库更新

MaxMind 每周更新 GeoLite2 数据库。要更新:

1. 下载新的 `GeoLite2-Country.mmdb` 文件
2. 替换配置路径下的现有文件
3. 数据库将在下次查找时自动重新加载（无需重启）

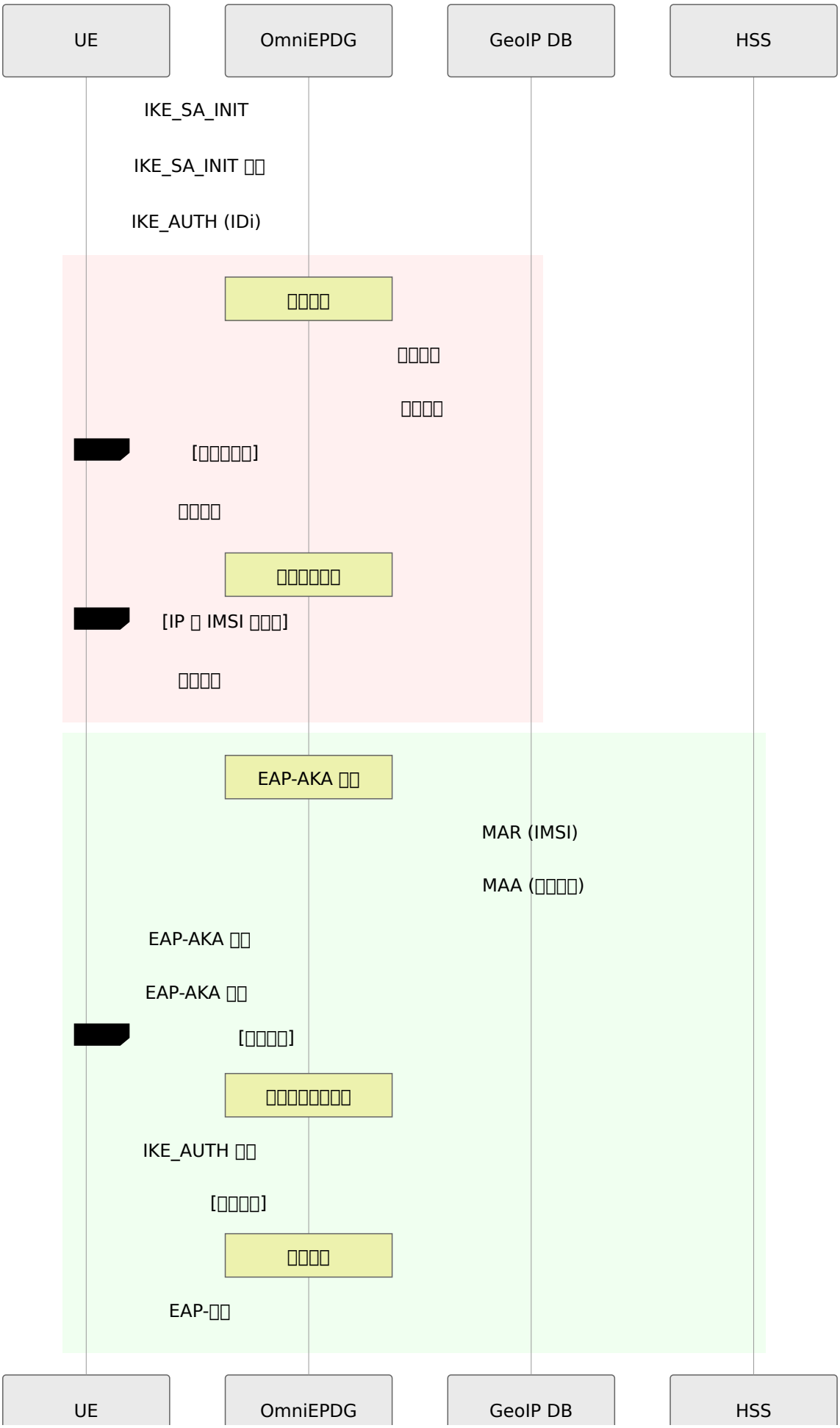
常见国家代码

代码	国家	代码	国家
AU	澳大利亚	US	美国
NZ	新西兰	GB	英国
CA	加拿大	DE	德国
FR	法国	JP	日本
SG	新加坡	HK	香港
IN	印度	CN	中国

完整列表: ISO 3166-1 alpha-2

安全流程

完整的认证安全流程:



UE
OmniEPDG
GeoIP DB
HSS
IKE_SA_INIT
IKE_SA_INIT 响应
IKE_AUTH (IDi)
安全检查
查找国家
国家代码
[国家被阻止]
认证失败
检查速率限制
[IP 或 IMSI 被阻止]
认证失败
EAP-AKA 认证
MAR (IMSI)
MAA (认证向量)
EAP-AKA 挑战
EAP-AKA 响应
[认证成功]
清除速率限制历史
IKE_AUTH 成功
[认证失败]
记录失败
EAP-失败
UE
OmniEPDG
GeoIP DB
HSS

指标

速率限制指标

指标：`epdg_auth_rate_limited_count`

类型：计数器

描述：被速率限制阻止的认证尝试次数

标签：

- `type` - 阻止原因`???``ip`（IP 阈值超出）或 `imsi`（IMSI 阈值超出）

示例查询：

```
# 每分钟被□率限制的尝试
rate(epdg_auth_rate_limited_count[1m])

# 按类型被□率限制
sum by (type) (rate(epdg_auth_rate_limited_count[5m]))

# 警报：高□率限制活动
rate(epdg_auth_rate_limited_count[5m]) > 10
```

GeoIP 指标

指标：`epdg_auth_geoip_blocked_count`

类型：计数器

描述：被 GeoIP 阻止的认证尝试次数

标签：

- `country` - ISO 3166-1 alpha-2 国家代码，或 `UNKNOWN` 表示无法解析的 IP

示例查询：

```
# 每分钟的 GeoIP 阻止
rate(epdg_auth_geoip_blocked_count[1m])

# 被阻止的国家前十名
topk(10, sum by (country) (epdg_auth_geoip_blocked_count))

# 警报：异常国家尝试访问
increase(epdg_auth_geoip_blocked_count{country="XX"}[1h]) > 100
```

故障排除

速率限制问题

合法用户被阻止

症状：用户报告在多次失败尝试后无法连接

可能原因：

- 用户多次输入错误凭证
- 网络问题导致认证超时被计为失败
- 环境的阈值设置过低

解决方案：

1. 检查受影响的 IP/IMSI 的指标
2. 如果误报常见，考虑提高阈值
3. 修复根本原因后，阻止将自动过期

高速率的被阻止尝试

症状：`epdg_auth_rate_limited_count` 快速增加

可能原因：

- 暴力攻击正在进行
- 配置错误的用户设备重复失败认证

- 凭证填充攻击

解决方案：

1. 检查日志中的源 IP 以寻找模式
2. 考虑为持续攻击者实施 IP 级防火墙规则
3. 如果合法用户受到影响，请验证 HSS 连接

GeoIP 问题

所有连接被阻止

症状：启用 GeoIP 后没有用户设备可以连接

可能原因：

- 数据库文件未找到或损坏
- 配置中的国家代码错误
- `geoip_allow_unknown: false` 阻止实验环境中的私有 IP

解决方案：

1. 验证数据库文件□否存在于配置路径
2. 检查国家代码□否正确（大写，2 个字母）
3. 对于实验/开发环境，设置 `geoip_allow_unknown: true`
4. 检查日志以获取与 GeoIP 相关的警告

GeoIP 数据库未加载

症状：日志中警告：“未找到 GeoIP 数据库”

可能原因：

- 文件路径不正确
- 文件权限阻止读取
- 文件不□有效的 MMDB 格式

解决方案：

1. 验证文件□否存在: `ls -la /etc/omniepdg/GeoLite2-Country.mmdb`
2. 检查权限: `chmod 644 /etc/omniepdg/GeoLite2-Country.mmdb`
3. 通过从 MaxMind 下载新副本来验证文件完整性

意外的国家阻止

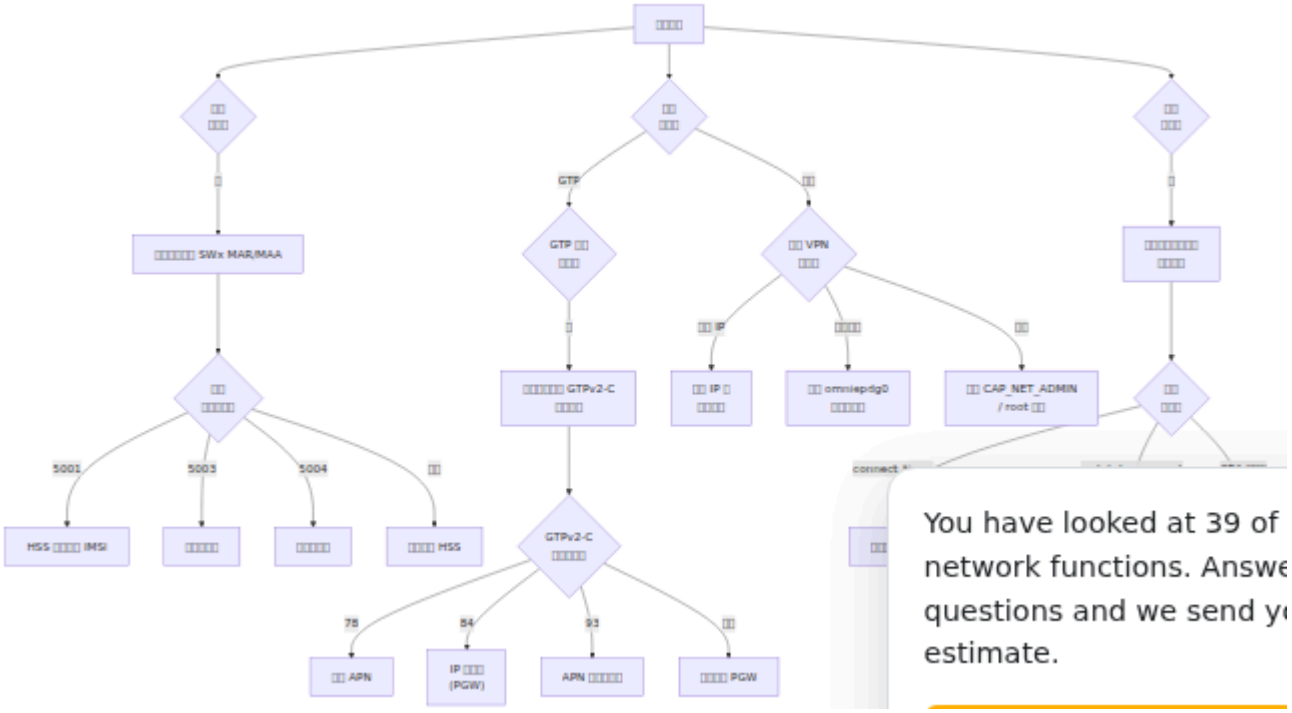
症状：来自允许国家的用户被阻止

可能原因：

- VPN/代理使 IP 看起来来自不同国家
- 过时的 GeoIP 数据库
- 企业网络出口在意外位置

解决方案：

1. 更新 GeoIP 数据库到最新版本
2. 检查用户的实际出口 IP 与预期国家
3. 如果用户通过企业网络漫游，考虑添加其他国家



OmniEPDG 故障排除

本指南涵盖了 OmniEPDG 的常见操作问题、诊断程序和解决步骤。

诊断概述

日志文件

OmniEPDG 将日志写入相对于应用程序工作目录的 `log/` 目录。有关日志配置详细信息，请参见 配置参考。

文件	目的	何时检查
`log/console.log`	所有应用程序消息，调试级别	任何问题的首要调查点
`log/error.log`	仅错误	快速扫描活动问题
`log/crash.log`	OTP 进程崩溃	当进程意外重启时
`log/erlang.log`	Erlang 内核日志	低级 Erlang/OTP 问题

关键日志模式

直径对等连接事件：

- `peer_up` - 直径对等方已连接并交换能力
- `peer_down` - 直径对等方已断开连接

UE FSM 状态转换：

- `ue_fsm state_<name> event=<event>` - FSM 在给定状态下处理事件
- `ue_fsm init(<IMSI>)` - 为订阅者创建新的 FSM 实例
- `terminating epdg_ue_fsm with reason <reason>` - FSM 正在关闭

超时事件：

- `Timeout swm_der_timeout` - SWm DER 响应超时
- `Timeout create_session_timeout` - GTPv2-C 创建会话响应超时
- `Timeout s2b_delete_session_timeout` - GTPv2-C 删除会话响应超时
- `Timeout cancel_location_timeout` - 取消位置响应超时

直径连接问题

HSS 连接失败（SWx）

症状：没有订阅者可以进行认证。日志显示重复尝试连接 HSS。

可能原因：

- 防火墙阻止 OmniEPDG 和 HSS 之间的 SCTP 端口 3868
- 配置中的 `dia_swx_remote_ip` 或 `dia_swx_remote_port` 不正确
- HSS 未运行或不接受直径连接
- 网络路径上未启用 SCTP（某些防火墙默认阻止 SCTP）
- Origin-Host 或 Origin-Realm 不匹配导致 CEA 拒绝

解决方案：

1. 验证与 HSS IP 和端口的网络连接

2. 确认 `dia_swx_remote_ip` 和 `dia_swx_remote_port` 与 HSS 配置匹配
3. 检查所有防火墙□否允许 SCTP 流量。如果 SCTP 被阻止，将 `dia_swx_proto` 设置为 `tcp` 作为后备
4. 验证 `dia_swx_origin_host` □可解析的 FQDN，并且与 HSS 期望的匹配
5. 检查 HSS 日志以获取直径 CER/CEA 协商失败的信息

PGW ���接失败 (S6b)

症状：认证成功但 GTP 隧道创建失败或 PGW 从未收到 S6b AAR。日志显示没有 S6b peer_up 事件。

可能原因：

- PGW 未配置为连接 OmniEPDG 的 S6b 监听器
- 防火墙阻止 OmniEPDG 的 S6b 绑定地址上的 SCTP 端口 3868
- PGW 无法访问 `dia_s6b_local_ip`
- Origin-Host 或 Origin-Realm 不匹配

解决方案：

1. 确认 PGW 配置为连接 OmniEPDG 的 `dia_s6b_local_ip`:`dia_s6b_local_port`
2. 验证 S6b 绑定地址可以从 PGW 网络访问
3. 检查防火墙规则允许在 S6b 地址的端口 3868 上的入站 SCTP
4. 验证 `dia_s6b_origin_host` 和 `dia_s6b_origin_realm` 与 PGW 期望的匹配

直径看门狗故障

症状：已建立的直径连接间歇性掉线。日志显示看门狗转换为 SUSPECT 或 DOWN 状态。

可能原因：

- 网络路径不稳定或丢包
- 对等方过载，未在 `dia_swx_watchdog_timer` 内响应 DWR
- 激进的看门狗配置（在声明可疑之前重试次数过少）

解决方案：

1. 检查 OmniEPDG 与对等方之间的网络路径质量（丢包、延迟）

2. 如果预期会有丢包，增加 `dia_swx_watchdog_config` / `dia_s6b_watchdog_config` 阈值（例如，`[{okay, 5}, {suspect, 3}]`）
3. 检查对等系统的健康状况（CPU、内存、连接数）

IKEv2 协商故障

未选择提案

症状：UE 的 IKE_SA_INIT 被拒绝，错误为 `NO_PROPOSAL_CHOSEN`；日志显示 `exchange error::no_proposal_chosen`。UE 从未通过 IKE_SA_INIT。

可能原因：

- UE 仅提供 OmniEPDG 不支持的加密变换。
- 扫描器或配置错误的客户端探测公共端口。

背景：OmniEPDG 接受 AES-CBC 和 AES-GCM 加密；HMAC-SHA2、HMAC-SHA1 和 **AES-XCBC-96** 完整性；HMAC-SHA2、HMAC-SHA1 和 **PRF-AES128-XCBC** PRF；以及 MODP/ECP/Curve25519/448 DH 组。有关完整列表，请参见 加密算法。某些手机（尤其某些三星 VoWiFi 客户端）仅提供 `AES-CBC-256 / AUTH_AES_XCBC_96 / PRF_AES128_XCBC / MODP-2048`，均被支持。

解决方案：

1. 启用 `:debug` 日志并检查 UE 的 SA 负载中的 `IKEv2 packet` 事件。`payloads[].proposals[]` 字段列出提供的确切变换。
2. 确认提供的加密/完整性/PRF/DH 变换在算法表中显示为支持。
3. 如果确实需要不支持的变换，请与工程部门提出。

连接建立后立即掉线（子 **SA** 被删除）

症状：IKE_AUTH 完成，会话短暂建立，然后 UE 在 ~100 毫秒内发送信息 `DELETE`。没媒体流。每次尝试都重复。

可能原因：

- 响应中的子 SA（ESP）变换与 UE 提供的变换不匹配，因此 UE 关闭 SA。
- 数据平面无法处理的 ESP 完整性/加密算法。

背景：OmniEPDG 从 UE 提供的提案中协商 ESP 子 SA（包括 AES-XCBC-96 和 AES-CBC-128/192/256），而不强制使用固定套件。选择 UE 未提供的变换的响应根据 RFC 7296 被拒绝。

解决方案：

1. 在 `:debug` 中检查 `child SA proposal selected` 日志行。`esp_offered`、`esp_encr` 和 `esp_integ` 字段显示提供和选择的内容。
2. 验证所选 ESP 加密密钥长度与密码匹配（不匹配会显示为 `Bad key size` 和丢弃的数据包）。
3. 确认 UE 提供的 ESP 完整性 OmniEPDG 支持的（请参见 ESP 子 SA 算法）。

认证故障

IMSI 未知（直径 5001）

症状：特定订阅者无法通过 EAP-AKA 认证。日志显示 SWx MAA 结果代码 5001（DIAMETER_ERROR_USER_UNKNOWN）。

可能原因：

- 订阅者未在 HSS 中配置
- UE SIM 和 HSS 数据库之间的 IMSI 不匹配
- NAI 格式不正确，导致 IMSI 提取失败

解决方案：

1. 验证订阅者 IMSI 否存在于 HSS 数据库中
2. 检查日志中的 NAI 格式否符合预期模式：`0<IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org`
3. 确认 SIM 卡 IMSI 与 HSS 配置的值匹配

授权被拒绝（直径 5003）

症状：订阅者通过认证，但在服务器分配期间被拒绝。日志显示 SWx SAA 结果代码 5003。

可能原因：

- 订阅者未被授权使用 WiFi 通话服务

- 此订阅者不允许使用 APN
- 订阅配置文件限制

解决方案：

1. 检查 HSS 中订阅者的服务配置文件
2. 验证□否为订阅者启用 WiFi 通话 / ePDG 访问
3. 确认请求的 APN 在订阅者的允许 APN 列表中

不允许漫游（直径 **5004**）

症状：漫游订阅者无法通过认证。日志显示 SWx MAA 或 SAA 结果代码 5004。

可能原因：

- HSS 漫游策略拒绝订阅者的当前位置
- 不允许漫游订阅者使用 WiFi 通话

解决方案：

1. 审查 HSS 漫游政策以获取订阅者的 HPLMN/VPLMN 组合
2. 检查漫游协议下□否允许 WiFi 通话

认证超时

症状：认证挂起，然后在 10 秒后失败。日志显示 `Timeout swm_der_timeout` 在 `state_wait_auth_resp` 中。

可能原因：

- HSS 在 10 秒内未响应 SWx MAR
- SWx 直径连接在请求期间断开
- HSS 过载

解决方案：

1. 检查 HSS 的响应能力和负载
2. 验证 SWx 直径对等方处于 OKAY 状态（而不□ SUSPECT 或 DOWN）

3. 检查 `dia_swx_transmit_timer` 是否足够适应到 HSS 的网络延迟

EAP-AKA 类型不匹配

症状：认证失败，日志中显示 "type_mismatch" 错误。UE 身份前缀与使用的 EAP 方法不匹配。

可能原因：

- UE 发送带前缀 `0`（EAP-AKA）的身份，但网络期望 EAP-AKA'，反之亦然
- HSS 返回错误 EAP 类型的认证向量

背景：根据 3GPP TS 23.003，NAI 身份前缀指示预期的认证类型：

- 前缀 `0` 表示 EAP-AKA
- 前缀 `6` 表示 EAP-AKA'

OmniEPDG 根据 UE 的身份前缀自动选择认证方法。大多数 WiFi 通话 UE 使用前缀 `0`（EAP-AKA）。

解决方案：

1. 检查日志中的 UE NAI 身份以验证前缀
2. 确保 HSS 配置为返回适当的认证向量
3. 验证 SIM 卡是否正确配置以匹配预期的认证类型

EAP-AKA RES 不匹配

症状：在挑战/响应后认证失败。日志显示 "RES mismatch" 或 "res_mismatch" 错误。

可能原因：

- SIM 卡认证失败
- UE 和网络之间的密钥派生不匹配
- HSS 返回的认证向量损坏

解决方案：

1. 验证 SIM 卡是否有效且未损坏
2. 检查 HSS 返回的认证向量是否有效（RAND、AUTN、XRES、CK、IK）

3. 启用调试日志以比较预期的 XRES 和接收到的 RES
4. 如果使用测试 SIM，请验证 SIM 和 HSS 之间的 Ki 和 OP/OPc 值□否匹配

GTP 隧道故障（仅 GTP 模式）

PGW 拒绝创建会话

症状：认证成功，但隧道创建失败。日志显示 GTPv2-C 创建会话响应带有错误原因。

常见原因代码和□作：

原因代码	名称	□作
78	缺少或未知 APN	验证 PGW 上□否配置 APN 并与订阅者配置文件匹配
82	在 RAT 中被拒绝	检查 PGW 策略□否允许 WiFi（非 3GPP）接入类型
84	所有动态地址已占用	PGW IP 池耗尽；扩展池或调查泄漏
92	用户认证失败	PGW 侧认证失败；检查 S6b 会话授权
93	APN 访问被拒绝	订阅者未被授权在 PGW 上使用 APN
96	IMSI/IMEI 未知	订阅者在 PGW 中未知；验证 S6b 会话□否已授权
113	APN 拥塞	APN 过载；重试或调查 PGW 容量
120	GTP-C 实体拥塞	PGW 控制平面过载

创建会话超时

症状：隧道创建挂起 10 秒后失败。日志显示 `Timeout create_session_timeout` 在 `state_wait_create_session_resp` 中。

可能原因：



- PGW 在 `gtpc_remote_ip`:`gtpc_remote_port` 上无法访问
- 防火墙阻止 OmniEPDG 和 PGW 之间的 UDP 端口 2123
- PGW 过载，未响应 GTPv2-C 请求

解决方案：

1. 验证与 PGW 在 UDP 端口 2123 上的网络连接
2. 检查防火墙规则允许 OmniEPDG 和 PGW 之间的 UDP 2123
3. 检查 PGW 健康状况和 GTPv2-C 处理能力

GTP-U 隧道未传递流量

症状：隧道已建立（创建会话成功），但订阅者流量未流动。

可能原因：

- GTP-U 内核模块未加载
- `gtp_u_kmod` 套接字 IP 与信号传递给 PGW 的 GTP-U 隧道端点地址不匹配
- 未为 GTP 隧道设备配置路由
- 防火墙阻止 UDP 端口 2152（GTP-U）

解决方案：

1. 验证 Linux 内核 GTP 模块是否已加载（`lsmod | grep gtp`）
2. 确认 GTP 隧道设备存在（`ip link show gtp0`）
3. 验证 `gtp_u_kmod` 的 `ip` 与 `gtpc_local_ip` 或在创建会话请求中信号传递的地址匹配
4. 检查路由表是否包含通过 GTP 隧道设备的路由
5. 验证防火墙允许 OmniEPDG 和 PGW 之间的 UDP 2152

简单 VPN 故障（仅简单 VPN 模式）

TUN 接口未创建

症状：OmniEPDG 启动但没有出现 `omniepdg0` 接口。会话在隧道设置时失败。日志可能显示 `simple_vpn_route` 在启动时的错误。

可能原因：

- OmniEPDG 进程缺少 `CAP_NET_ADMIN` 能力或未以 root 身份运行
- TUN/TAP 内核模块未加载
- 另一个进程已经创建了名为 `omniepdg0` 的接口

解决方案：

1. 验证 TUN 内核模块□否可用（`lsmod | grep tun`）
2. 确认 OmniEPDG 以足够的权限运行以创建 TUN 接口
3. 检查 `omniepdg0` □否已经存在于之前的实例中（`ip link show omniepdg0`）
4. 检查 `log/crash.log` 中路由管理器进程的错误

IP 池耗尽

症状：认证成功，但隧道设置失败。日志显示来自 `simple_vpn_pool` 的 IP 分配失败。

可能原因：

- 配置的 CIDR 池中的所有地址都已分配给活动会话
- 会话拆除后未释放 IP 地址（泄漏）
- 池大小对于并发订阅者数量来说太小

解决方案：

1. 检查活动的 `epdg_ue_fsm` 进程数量与池大小的关系
2. 验证会话□否正确拆除（检查 `terminating` 日志消息）
3. 如果池确实已满，通过在 `simple_vpn_pool_ipv4` 中使用更大的 CIDR 前缀扩展池（需要重启）
4. 检查 `log/crash.log` 中的 FSM 崩溃信息，这可能阻止了 IP 的释放

订阅者流量未流动

症状：会话已建立，UE 收到 IP 地址，但流量未通过 TUN 接口流动。

可能原因：

- 未在 `omniepdg0` 上为订阅者 IP 添加主机路由
- OmniEPDG 主机上未启用 IP 转发

- 防火墙规则阻止在 `omniepdg0` 接口上的流量
- 缺少针对订阅者 IP 范围的 NAT/伪装规则

解决方案：

1. 验证主机路由是否存在（`ip route show` 并查看订阅者的 /32 路由通过 `omniepdg0`）
2. 确认已启用 IP 转发（`sysctl net.ipv4.ip_forward`）
3. 检查 iptables/nftables 规则是否允许通过 `omniepdg0` 转发
4. 如果订阅者需要访问互联网，请验证是否为订阅者 IP 范围配置了 NAT/伪装（例如，`iptables -t nat -A POSTROUTING -s 10.45.0.0/16 -o <wan-interface> -j MASQUERADE`）

崩溃后的陈旧路由

症状：在 OmniEPDG 重启后或会话异常终止后，订阅者 IP 的主机路由仍保留在路由表中。

可能原因：

- FSM 在路由被删除之前崩溃
- OmniEPDG 进程在未正常关闭的情况下被终止

解决方案：

1. 检查 `log/crash.log` 中的进程崩溃信息
2. 手动删除陈旧路由（`ip route del <subscriber-ip>/32 dev omniepdg0`）
3. 重启 OmniEPDG 将重新创建 `omniepdg0` 接口，从而删除所有相关路由

会话拆除问题

拆除在注销期间挂起

症状：会话拆除未完成。UE FSM 卡在 `dereg_*` 或 `wait_*` 状态。

可能原因：

- PGW 未响应删除会话请求
- 直径对等方未响应 STR 或 ASR

- 超时级联未完成，导致多个超时堆叠

解决方案：

1. 检查相关状态中的日志以获取超时消息
2. 验证 PGW 和 HSS 的连接
3. 在 10 秒后，FSM 应超时并继续进行下一个拆除步骤或终止。如果没有，请检查是否有意外事件记录为 `Unexpected call event`

孤立的 GTP-U PDP 上下文

症状：会话终止后，GTP-U 隧道条目仍保留在内核中。`ip link show gtp0` 显示设备仍有活动 PDP 上下文。

可能原因：

- FSM 在删除 PDP 上下文之前异常终止
- 拆除序列期间崩溃

解决方案：

1. 检查 `log/crash.log` 中的进程崩溃信息
2. FSM 的 `terminate/3` 回调尝试清理 PDP 上下文。如果 FSM 被杀死（例如，监视器重启），则可能跳过了清理
3. 重启 OmniEPDG 将重新创建 GTP-U 套接字并清除陈旧的上下文

进程和系统问题

监视器重启循环

症状：OmniEPDG 进程反复重启。日志显示监视器重启消息和崩溃报告。

可能原因：

- 持续的配置错误导致处理程序在启动时崩溃
- 外部依赖不可用（例如，未找到 gen_socket 库）
- 直径对等方发送格式错误的消息导致处理程序崩溃

解决方案：

1. 检查 `log/crash.log` 以获取崩溃的根本原因
2. 验证 `gen_socket` 的 `libdir` 路径□否正确，并且库文件存在
3. 检查 `config/runtime.exs` 中□否存在所有必需的配置参数
4. 查找崩溃报告中的格式错误的直径消息

高内存使用

症状：Erlang VM 内存消耗随时间增长。

可能原因：

- 会话拆除后未清理 UE FSM 进程
- 邮箱中日志消息积累
- 大量并发会话

解决方案：

1. 检查正在运行的 `epdg_ue_fsm` 和 `aaa_ue_fsm` 进程的数量（这些应与活动订阅者数量匹配）
2. 验证 FSM 在会话拆除后□否正确终止（检查 `terminating` 日志消息）
3. 审查日志轮换设置以确保日志文件正在轮换

OmniEPDG Webhooks

OmniEPDG 可以发送出站 webhook 以处理订阅者会话生命周期事件。当订阅者连接或断开时，OmniEPDG 会向每个配置的端点发送带有 JSON 主体的 HTTP `POST` 请求。使用 webhook 可以实时为计费系统、在线状态服务或分析管道提供数据。

默认情况下，webhook 禁用的。请参见 Webhooks Configuration 以获取配置参数。

目录

- 概述
- 事件
- 交付模型
- 有效负载模式
- 示例有效负载
- 接收者要求
- 故障排除

概述

OmniEPDG 使用状态机跟踪每个订阅者会话。两个转换会生成 webhook:

- 会话在成功连接后达到 `active` 状态。
- 会话在断开后达到 `terminating` 状态。

下图显示了在会话生命周期中每个 webhook 发送的位置。

事件

事件	触发	描述
`session_established`	成功连接	当订阅者认证并且 PDN 连接建立时，OmniEPDG 发送此事件。会话准备好承载流量。
`session_terminated`	断开	当一个已建立的会话被拆除时，OmniEPDG 发送此事件。原因可能包括 UE 断开、非活动超时或核心发起的拆除。

OmniEPDG 仅对达到 `active` 状态的会话发送 `session_terminated`。在 PDN 连接建立之前失败的连接不会生成 webhook。

设置 `events` 配置参数以控制 OmniEPDG 发送哪些事件。请参见 Webhooks Configuration。

交付模型

OmniEPDG 采用火并忘模式交付 webhook。每次交付在受监控的后台任务中运行。交付不会阻塞或延迟会话状态机，慢或不可用的端点不会影响呼叫处理。

请注意以下交付属性：

- OmniEPDG 每个配置的端点、每个事件发送一个 `POST`。
- OmniEPDG 不会重试失败的交付。失败的交付会被记录并丢弃。
- OmniEPDG 不保证两个事件之间的事件顺序。设计接收者以通过其 `timestamp` 字段处理事件。
- OmniEPDG 接受任何 `2x` HTTP 响应作为成功。对于连接错误或非 `2xx` 响应，它会记录警告。
- OmniEPDG 不验证 `https` 端点的 TLS 证书。使用受信任的网络路径连接到接收者。

有效负载模式

每个 webhook 主体□一个 JSON 对象，包含以下字段。

字段	类型	描述
`event`	字符串	事件名称: `session_established` 或 `session_terminated`。
`timestamp`	字符串	事件时间，采用 ISO 8601 格式（UTC）。
`session_id`	字符串	OmniEPDG 内部会话标识符。
`imsi`	字符串	订阅者 IMSI。
`apn`	字符串或 null	PDN 连接的接入点名称。
`pdn_type`	字符串	PDN 类型: `ipv4`、`ipv6` 或 `ipv4v6`。
`ue_ipv4`	字符串或 null	分配给 UE 的 IPv4 地址。
`ue_ipv6`	字符串或 null	分配给 UE 的 IPv6 地址。
`peer_ip`	字符串或 null	UE 的公共 IP 地址（WiFi 接入点）。
`geoip`	对象或 null	`peer_ip` 的地理位置。请参见 GeoIP Object。当无法定位地址时，它为 `null`。
`handover`	布尔值	如果会话□作为 LTE 到 WiFi 的切换创建，则为 `true`。
`bytes_in`	整数	事件发生时会话的上行字节计数。
`bytes_out`	整数	事件发生时会话的下行字节计数。
`metadata`	对象	包含额外事件上下文的对象。对于大多数事件，它□空的。

`bytes_in` 和 `bytes_out` 字段在 `session_established` 事件中为 `0`，因为尚未传输流量。

GeoIP 对象

OmniEPDG 在连接时使用 UE 公共 IP 的地理位置注释每个会话。注释使用捆绑的 DB-IP 数据库，无论□否启用 GeoIP 访问控制，都会发生。`geoip` 字段承载该注释。`city`、`latitude` 和 `longitude` 字段仅在使用城市级数据库时存在。

字段	类型	描述
`country_code`	字符串或 null	ISO 3166-1 alpha-2 国家代码（例如 `AU`）。
`country_name`	字符串或 null	国家名称。
`continent_code`	字符串或 null	两个字母的大陆代码（例如 `OC`）。
`city`	字符串或 null	城市名称。仅在使��城市级数据库时存在。
`latitude`	数字或 null	近似纬度。仅在使用城市级数据库时存在。
`longitude`	数字或 null	近似经度。仅在使用城市级数据库时存在。

□□□□□□

session_established


```
{
  "event": "session_established",
  "timestamp": "2026-08-30T02:15:44.123456Z",
  "session_id": "a1b2c3d4-...",
  "imsi": "001010000000001",
  "apn": "ims",
  "pdn_type": "ipv4v6",
  "ue_ipv4": "100.64.32.5",
  "ue_ipv6": null,
  "peer_ip": "203.0.113.9",
  "geoip": {
    "country_code": "AU",
    "country_name": "Australia",
    "continent_code": "OC",
    "city": "Sydney",
    "latitude": -33.8688,
    "longitude": 151.2093
  },
  "handover": false,
  "bytes_in": 0,
  "bytes_out": 0,
  "metadata": {}
}
```

session_terminated


```
{
  "event": "session_terminated",
  "timestamp": "2026-08-30T03:42:10.998877Z",
  "session_id": "a1b2c3d4-...",
  "imsi": "001010000000001",
  "apn": "ims",
  "pdn_type": "ipv4v6",
  "ue_ipv4": "100.64.32.5",
  "ue_ipv6": null,
  "peer_ip": "203.0.113.9",
  "geoip": {
    "country_code": "AU",
    "country_name": "Australia",
    "continent_code": "OC",
    "city": "Sydney",
    "latitude": -33.8688,
    "longitude": 151.2093
  },
  "handover": false,
  "bytes_in": 154238,
  "bytes_out": 892011,
  "metadata": {}
}
```


接收者要求

构建 webhook 接收者以满足以下要求：

- 接受带有 `Content-Type: application/json` 主体的 HTTP `POST` 请求。
- 一旦接受事件，立即返回 `2xx` 响应。不要为下游处理保持连接打开。
- 使用 `session_id` 字段将事件与会话匹配。
- 处理重复事件。网络条件可能导致 OmniEPDG 多次发送相同事件。
- 按 `timestamp` 字段对事件进行排序，而不□按到达顺序。

故障排除

没有 **webhook** 到达

症状：接收者在连接或断开后没有收到请求。

可能的原因：

- webhook 被禁用（`enabled: false`）。
- 事件不在 `events` 列表中。
- 端点 URL 错误或无法从 OmniEPDG 主机访问。
- 防火墙阻止了到端点的出站连接。

解决方案：

1. 确认 webhook 配置中的 `enabled: true`。
2. 确认事件名称在 `events` 列表中。
3. 确认端点 URL 正确并且可以从 OmniEPDG 主机访问。
4. 检查 OmniEPDG 日志以获取 webhook 交付警告。

仅收到 **session_established**，而没有 **session_terminated**

症状：接收者收到连接事件，但没有断开事件。

���能的原因：

- `session_terminated` 不在 `events` 列表中。
- 会话从未达到 `active` 状态，因此没有断开事件。

解决方案：

1. 将 `:session_terminated` 添加到 `events` 列表中。
2. 确认会话在结束之前完成成功连接。

交付被记录为失败

症状: OmniEPDG 日志显示 webhook 交付警告。

可能的原因:

- 接收者返回非 `2xx` 响应。
- 接收者响应缓慢或不可用。
- 端点 URL 使用的主机无法解析。

解决方案:

1. 确认接收者返回 `2xx` 响应。
2. 确认接收者响应迅速，因为 OmniEPDG 不会重试。
3. 确认端点主机可以从 OmniEPDG 主机解析。