← 述

OmniSMF 置参考

操作员可写配置通过 `config/runtime.exs` 提供，该配置主要由操作系统环境变量驱动。下面的大多数值都有内置默认值，并且在 SMF 启动之前可以通过环境变量覆盖。少数键仅通过应用程序配置设置，并且没有环境变量（`chf_uri`、`charging` 块、`pgw_c_addr` 和 `pco` 块）；这些将在出现时进行说明。编译的应用程序环境键为 `:omnismf`。

```
config :omnismf,
  sbi_scheme: "http",                       # SBI_SCHEME
  sbi_addr:   "127.0.0.4",                  # SBI_ADDR
  sbi_port:   7777,                         # SBI_PORT
  nrf_uri:    "http://127.0.0.1:7777",      # NRF_URI
  udm_uri:    "http://127.0.0.12:7777",     # UDM_URI
  pcf_uri:    "http://127.0.0.13:7777",     # PCF_URI
  amf_uri:    "http://127.0.0.5:7777",      # AMF_URI
  upf_addr:   "10.179.1.24",                # UPF_ADDR
  upf_pfcp_port: 8805,                      # UPF_PFCP_PORT
  pfcp_addr:  "0.0.0.0",                    # PFCP_ADDR
  mcc: "999",                               # MCC
  mnc: "70",                                # MNC
  dns:  ["8.8.8.8", "8.8.4.4"],             # DNS
  dns6: ["2001:4860:4860::8888"],           # DNS6
  mtu:  1400,                               # MTU
  prometheus_metrics_port: 9568,            # PROMETHEUS_PORT
  heartbeat_interval: 10_000,               # HEARTBEAT_INTERVAL
  ue_ip_pool: [subnet_map: %{default: ["10.45.0.0/16"]}]  #
UE_IP_SUBNET_MAP / UE_IP_POOL
```

SBI□□□□□ PLMN □□

□□	□□	□□	□□	□□□□
`sbi_scheme`	□□□	□	`"http"`	`SBI_SCHEME`
`sbi_addr`	□□□	□	`"127.0.0.4"`	`SBI_ADDR`
`sbi_port`	□□	□	`7777`	`SBI_PORT`
`nrf_uri`	□□□	□	`"http://127.0.0.1:7777"`	`NRF_URI`
`udm_uri`	□□□	□	`"http://127.0.0.12:7777"`	`UDM_URI`
`pcf_uri`	□□□	□	`"http://127.0.0.13:7777"`	`PCF_URI`
`amf_uri`	□□□	□	`"http://127.0.0.5:7777"`	`AMF_URI`
`upf_addr`	□□□	□	`"10.179.1.24"`	`UPF_ADDR`

□□	□□	□□	□□	□□□□
`upf_pfcp_port`	□□	□	`8805`	`UPF_PFCP_POR`
`pfcp_addr`	□□□	□	`"0.0.0.0"`	`PFCP_ADDR`
`mcc`	□□□	□	`"999"`	`MCC`
`mnc`	□□□	□	`"70"`	`MNC`
`heartbeat_interval`	□□□□□	□	`10000`	`HEARTBEAT_IN`

N26 互通

参数	类型	必需	默认	环境变量	描述
`pgw_c_addr`	字符串（IPv4）	是，针对 N26 互通	无	（仅应用程序配置）	在 N26 互通期间，在 EPS PDN 连接视图中返回的 PGW-C S5/S8-C（GTP-C）地址。AMF 将其放入转发重定位请求中，因此这��� SGW 发送 S5/S8-C 请求以进行互通会话的地址。没有默认值：当未设置或不是 IPv4 地址时，N26 互通 API（`/api/eps_handover`）将失败并显示命名 `pgw_c_addr` 的错误。OmniSMF 本身不终止 S5/S8-C（没有 GTP-C 监听器），因此此地址必须是能够处理 SGW 会话请求的 GTP-C 端点。通过 `:omnismf` 应用程序环境设置；此键没有环境变量。

N40 统一计费（Nchf，TS 32.290 / TS 32.291）

SMF 可以作为 N40（Nchf_ConvergedCharging）下的统一计费消费者。当启用时，它在每个 PDU 会话建立时打开计费数据请求，在会话期间根据 N4 URR 使用报告进行更新，并在释放时关闭。CHF 首先通过 NRF 发现进行解析，当发现未解析时回退到配置的 `chf_uri`。默认情况下禁用统一计费；在禁用状态下，计费路径为无操作，不会进行任何 CHF 调用。

顶级 `chf_uri` 键和 `charging` 块仅通过 `:omnismf` 应用程序配置设置；两者都没有环境变量。

```
config :omnismf, :charging,
  enabled: false,
  chf_uri: "http://127.0.0.20:7777"
```

参数	类型	必需	默认	环境变量	描述
`chf_uri`	字符串	否	`"http://127.0.0.20:7777"`	无（仅应用程序配置）	在 NRF 发现未解析时用于统一计费的 CHF 的静态回退基本 URI。作为顶级 `:omnismf` 键 `chf_uri` 设置。
`charging enabled`	布尔值	否	`false`	无（仅应用程序配置）	N40 统一计费的主开关。当为 `false` 时，计费路径为无操作，不会进行任何 CHF 调用。
`charging chf_uri`	字符串	否	（回退到顶级 `chf_uri`）	无（仅应用程序配置）	每个块的 CHF 基本 URI 覆盖。当在 `charging` 块下设置时，它优先于顶级 `chf_uri` 回退。

会话-AMBR 到 PCF

在创建 SM 策略关联时，订阅的会话-AMBR（从 UDM DNN 配置 `sessionAmbr` 字段读取）在 两个 规范字段名称 `subsSessAmbr`（TS 29.512 `SmPolicyContextData`）和遗留键 `sessionAmbr` 下发送到 PCF。仅发送 `sessionAmbr` 会使 `subsSessAmbr` 未设置，导致 PCF 回退到其 ~1000 Kbps 默认值并限制每个会话。没有操作员配置管理此项；在此处记录是为了让操作员了解 PCF 必须遵循的字段。

面向 UE 的 PCO 默认值（平面键）

这些平面键填充返回给 UE 的 PDU 会话建立接受中的协议配置选项。它们在存在结构化的 `:pco` 块时被覆盖。

参数	类型	必需	默认	环境变量	描述
`dns`	字符串列表	否	`["8.8.8.8", "8.8.4.4"]`	`DNS`	广告给 UE 的 IPv4 DNS 服务器。环境变量中以逗号分隔。
`dns6`	字符串列表	否	`["2001:4860:4860::8888"]`	`DNS6`	广告给 UE 的 IPv6 DNS 服务器。环境变量中以逗号分隔。
`mtu`	整数	否	`1400`	`MTU`	广告给 UE 的 IPv4 链路 MTU。

UE IP 池

参数	类型	必需	默认	环境变量
`ue_ip_pool`	关键字（`subnet_map`）	否	`%{default: ["10.45.0.0/16"]}`	`UE_IP_SUBNET_MAP` / `UE_IP_POOL`

`UE_IP_SUBNET_MAP` JSON 以确切的 DNN 名称、`^regex` 模式或 `"default"` 为键；每个值是 IPv4 和/或 IPv6 CIDR 的列表。匹配优先级为 确切 **DNN** → 正则表达式 → **`default`**。示例：

```
{
  "internet":    ["10.45.0.0/16", "2001:db8:45::/64"],
  "ims":         ["10.47.0.0/24"],
  "^enterprise": ["10.99.0.0/16"],
  "default":     ["10.45.0.0/16"]
}
```

如果既未设置 `UE_IP_SUBNET_MAP` 也未设置 `UE_IP_POOL`，则默认池为 `10.45.0.0/16`。有关分配和静态 IP 保留行为，请参见 UE IP 地址分配。

P-CSCF 和□□议□置选项（`:pco`）

对于 VoNR/IMS，SMF 在 PDU 会话建立接受 ePCO 中返回 P-CSCF 地址（以及其他 PCO 项：DNS、NBNS、链路 MTU）。此块使用与 **4G PGW-C** 相同的布局和选择逻辑，因此操作员以相同的方式配置两个节点。当存在时，`:pco` 值优先于上面的平面 `dns`/`dns6`/`mtu` 键。此块仅通过 `:omnismf` 应用程序配置设置，并且没有环境变量。

```
config :omnismf, :pco,
  p_cscf_ipv4_address_list: ["10.179.4.165"],
  p_cscf_ipv6_address_list: [],
  p_cscf_discovery_enabled: true,
  p_cscf_discovery_fqdn:
"pcscf.ims.mnc070.mcc999.3gppnetwork.org",
  p_cscf_discovery_dns_server: "10.179.2.10",
  p_cscf_discovery_timeout_ms: 5000,
  primary_dns_server_address: "8.8.8.8",
  secondary_dns_server_address: "8.8.4.4",
  primary_nbns_server_address: nil,
  secondary_nbns_server_address: nil,
  ipv4_link_mtu_size: 1400
```

□□	□□	□□	□□
`p_cscf_ipv4_address_list`	□□□□□	□	`[]`
`p_cscf_ipv6_address_list`	□□□□□	□	`[]`
`p_cscf_discovery_enabled`	□□□	□	`true`
`p_cscf_discovery_fqdn`	□□□	□	`pcscf.mnc{MNC}.mcc{MCC}.3gpp`
`p_cscf_discovery_dns_server`	□□□	□	□□□□□
`p_cscf_discovery_timeout_ms`	□□□□	□	`5000`

参数	类型	必需	默认
	秒）		
`primary_dns_server_address`	字符串	否	（平面 `dns`）
`secondary_dns_server_address`	字符串	否	（平面 `dns`）
`primary_nbns_server_address`	字符串 | nil	否	`nil`
`secondary_nbns_server_address`	字符串 | nil	否	`nil`
`ipv4_link_mtu_size`	整数	否	（平面 `mtu`）

遗留平面 **P-CSCF** 回退键

平面键 `pcscf` 和 `pcscf6` 是为与旧 PGW-C 配置保持一致而保留的遗留回退。仅在未设置结构化 `:pco` P-CSCF 地址列表（`p_cscf_ipv4_address_list` / `p_cscf_ipv6_address_list`）时才会查询它们。对于新部署，优先使用结构化 `:pco` 形式。每个键接受单个地址字符串或地址字符串列表。这些键仅通过 `:omnismf` 应用程序配置设置，并且没有环境变量。

参数	类型	必需	默认	环境变量	描述
`pcscf`	字符串或字符串列表	否	`[]`	无（仅应用程序配置）	遗留静态 P-CSCF IPv4 回退，仅在未设置 `p_cscf_ipv4_address_list` 时使用。优先使用结构化 `:pco` 形式。
`pcscf6`	字符串或字符串列表	否	`[]`	无（仅应用程序配置）	遗留静态 P-CSCF IPv6 回退，仅在未设置 `p_cscf_ipv6_address_list` 时使用。优先使用结构化 `:pco` 形式。

指标和管理端口

参数	类型	必需	默认	环境变量	描述
`prometheus_metrics_port`	整数	否	`9568`	`PROMETHEUS_PORT`	Prometh 指标导出器的 端口（`GET /metric`
管理 API 端口（`:api_ex`）	整数	否	`8443`	`API_PORT`	管理/OAM REST A TCP 端口 （TLS）。 盖，以便 SM 以在主机网络 行。
控制面板 HTTPS 端口	整数	否	`7443`	`CONTROL_PANEL_PORT`	控制面板 W UI 的 TC ﷺ （HTTPS

管理 API 和控制面板通过使用位于 `priv/cert/omnitouch.crt` 的证书和位于 `priv/cert/omnitouch.pem` 的密钥通过 TLS 提供。只有监听端口可以通过上述环境变量在运行时进行调整。

管理 / OAM API

管理 API（TLS，默认端口 `8443`，环境变量 `API_PORT`）公开以下操作和诊断的端点。所有管理端点都以 `/api` 前缀挂载。

方法	路径	目的
GET	`/api/status/api`	API 服务状态。
GET	`/api/status/nrf`	NRF 注册状态。
GET	`/api/status/nf`	NF（SMF）状态摘要。
GET	`/api/pdu_session`	列出活动的 PDU 会话。
POST	`/api/pdu_session`	创建 PDU 会话（操作测试）。
GET	`/api/pdu_session/{id}`	显示单个 PDU 会话。
DELETE	`/api/pdu_session/{id}`	删除单个 PDU 会话。
GET	`/api/upf`	UPF / PFCP 关联视图。
POST	`/api/eps_handover`	驱动 N26 EPS 互通（采用 / 切换 / 查找）。
GET	`/api/ip_pool`	整体 IPv4 容量以及每个 DNN 池摘要。
GET	`/api/ip_pool/detail`	每个 DNN 池及其范围的 CIDR、总数和分配的地址。
GET	`/api/statistics`	会话和操作统计信息。
GET	`/api/oam/config`	读取运行时配置。
PATCH/PUT	`/api/oam/config/{id}`	��新运行时配置值。
GET	`/api/oam/resources`	资源利用快照。
GET	`/api/oam/diagnostics`	诊断快照。

方法	路径	目的
GET	`/api/oam/peers`	SBI 对等体（UDM/PCF/AMF/NRF）状态。
GET	`/api/oam/pfcp_peers`	PFCP（UPF）对等体状态。
GET	`/api/oam/pfcp_session/{id}`	检查 PFCP 会话。
POST	`/api/oam/upf`	管理 UPF 关联。
POST	`/api/oam/pfcp/reassociate`	重新触发与 UPF 的 PFCP 关联设置。
POST	`/api/oam/cache/clear`	清除内部缓存。
POST	`/api/oam/log_level`	在不重启的情况下更改运行时日志级别。
POST	`/api/oam/nrf/reregister`	强制 NRF 重新注册。
DELETE	`/api/oam/session/{id}`	清空单个会话。
DELETE	`/api/oam/session/by_dnn/{id}`	按 DNN 清空会话。
DELETE	`/api/oam/session/by_supi/{id}`	按 SUPI 清空会话。
DELETE	`/api/oam/session/by_snssai/{id}`	按 S-NSSAI 清空会话。
DELETE	`/api/oam/session/by_upf/{id}`	按 UPF 清空会话。
POST	`/api/oam/session/modify`	修改会话（操作测试）。

日志记录

SMF 发出结构化 JSON 日志。默认日志格式化程序在 `runtime.exs` 中配置，且不打算在每个部署中进行编辑：

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

参数	类型	必需	默认	描述
`format`	格式化程序	否	`OmniLogger.JsonFormatter`	将每条日志行作为 JSON 对象发出，以便被日志管道摄取。
`metadata`	原子/列表	否	`:all`	在每条 JSON 行中包含所有 Logger 元数据（SUPI、PDU 会话 ID、SM 上下文引用、DNN、过程）。

可以通过管理 API 端点 `POST /api/oam/log_level` 在不重启的情况下更改运行时日志级别。通过 SUPI 过滤日志以跟踪订阅者的完整会话生命周期。

← 概述

OmniSMF 指标

指标在 `GET /metrics` 上暴露，使用 `prometheus_metrics_port`（默认 `9568`，环境变量 `PROMETHEUS_PORT`）。以下所有指标名称均以导出器所示的前缀开头。

PDU 会话指标

指标	类型	标签	描述
`omni_smf_pdu_session_create_count`	counter	`result`	PDU 会话创建操作。
`omni_smf_pdu_session_modify_count`	counter	`result`	PDU 会话修改操作。
`omni_smf_pdu_session_release_count`	counter	`result`	PDU 会话释放操作。
`omni_smf_pdu_session_creates_total`	counter	`result`, `dnn`	按结果和 DNN 统计的 PDU 会话创建总数。
`omni_smf_pdu_session_releases_total`	counter	`result`	按结果统计的 PDU 会话释放总数。
`omni_smf_active_pdu_sessions_count`	gauge	`dnn`	按 DNN 统计的活跃 PDU 会话数量。在会话建立和从 SMF 上下文存储释放时发出（活跃会话的真实来源）。
`omni_smf_session_duration_seconds`	distribution	none	PDU 会话持续时间（秒），在释

指标	类型	标签	描述
			放时发出并从会话建立开始测量（桶：1, 5, 15, 30, 60, 300, 900, 3600, 86400 s）。

示例（建立成功率）：

```
sum(rate(omni_smf_pdu_session_creates_total{result="success"}
[5m]))
  / sum(rate(omni_smf_pdu_session_creates_total[5m]))
```

示例（每个 DNN 的活跃会话）：

```
sum by (dnn) (omni_smf_active_pdu_sessions_count)
```

PFCP / UPF 指标

指标	类型	标签	
`omni_smf_pfcp_sessions_total`	counter	`operation`	P [/
`omni_smf_pfcp_associations_total`	counter	`result`	[[2
`omni_smf_pfcp_heartbeats_total`	counter	`result`	[[2
`omni_smf_upf_health`	gauge	none	U [[
`omni_smf_n4_usage_reports_total`	counter	none	[U 2
`omni_smf_n4_usage_report_volume_bytes_total`	counter	none	[[[

这两个 `omni_smf_n4_*` 指标是 SMF 唯一的用户面流量指标：它们来自 UPF 通过 N4 返回的 URR 使用报告，因此仅在 UPF 安装了使用报告规则并提供报告时才会���加。

示例（当 UPF 关联下降时警报）：

```
omni_smf_upf_health == 0
```

示例（所有会话的用户面流量吞吐量）：

```
sum(rate(omni_smf_n4_usage_report_volume_bytes_total[5m]))
```

示例（从 UPF 接收到的使用报告速率）：

```
sum(rate(omni_smf_n4_usage_reports_total[5m]))
```

示例（针对 UPF 的 PFCP 心跳失败）：

```
sum(rate(omni_smf_pfcp_heartbeats_total{result="failure"}[5m]))
```

N40 计费指标

指标	��型	标签	描述
`omni_smf_n40_charging_total`	counter	`operation`	针对 CHF 的 N40 Nchf 融合计费操作总数（TS 32.290）。

示例（按事件的融合计费操作速率）：

```
sum by (operation) (rate(omni_smf_n40_charging_total[5m]))
```

NRF 指标

指标	类型	标签	描述
`omni_smf_nrf_registration_status`	gauge	`nf_type`	NRF 注册状态（1 = 已注册，0 = 未注册）。

示例：

```
omni_smf_nrf_registration_status == 0
```

BEAM VM 指标

指标	类型	描述
`beam_memory_total`	gauge	总 BEAM 内存（字节）。
`beam_memory_processes`	gauge	分配给 Erlang 进程的内存。
`beam_memory_processes_used`	gauge	进程实际使用的内存。
`beam_memory_system`	gauge	系统内存。
`beam_memory_atom`	gauge	总原子内存。
`beam_memory_atom_used`	gauge	使用的原子内存。
`beam_memory_binary`	gauge	二进制内存。
`beam_memory_code`	gauge	代码内存。
`beam_memory_ets`	gauge	ETS 表内存。
`beam_processes_count`	gauge	Erlang 进程数量。
`beam_ports_count`	gauge	Erlang 端口数量。
`beam_atom_count`	gauge	原子数量。
`beam_vm_uptime`	gauge	VM 运行时间秒）。

活跃进程数量也作为会话规模信号，因为每个 PDU 会话都是一个专用工作进程。`beam_processes_count` 和 `omni_smf_active_pdu_sessions_count` 之间的持续差异可能表示孤立的会话工作进程（请参见 故障排除）。

← 概述

OmniSMF 操作指南

OmniSMF 是 5G 核心的会话管理功能 (SMF)。它负责 PDU 会话的完整生命周期：地址分配、策略关联、用户平面编程和拆除。本指南涵盖 3GPP 的角色、它所暴露的 SBI 和 N4 接口，以及它驱动的关键程序，每个程序都有一个序列图。

3GPP 角色和规范参考

规范	相关性
TS 23.501	系统架构：SMF 角色、PDU 会话概念、QoS 模型
TS 23.502	程序：PDU 会话建立 (§4.3.2)、修改 (§4.3.3)、释放 (§4.3.4)、服务请求 (§4.2.3)
TS 29.502	Nsmf_PDUSession 服务 API
TS 29.244	N4 接口：SMF 和 UPF 之间的 PFCP 协议
TS 29.503	Nudm_SubscriberDataManagement (sm-data) 和 Nudm_UEContextManagement (smf-registrations)
TS 29.512	Npcf_SMPolicyControl：向 PCF 的 SM 策略关联和通知
TS 29.518	Namf_Communication：向 AMF 的 N1N2MessageTransfer
TS 32.290 / TS 32.291	Nchf_ConvergedCharging (N40)：向 CHF 的融合计费
TS 24.501	5GS NAS (5GSM)：PDU 会话建立/修改、会话类型和 5GSM 原因处理 (§6.4.1.3)
TS 29.571	通用数据类型：`IpAddress`、S-NSSAI

接口

接口	对等方	协议	目的
N11	AMF	Nsmf_PDUSession (SBI, HTTP/2)	SM 上下文创建/检索/更新/释放，N1N2MessageTransfer
N7	PCF	Npcf_SMPolicyControl (SBI)	SM 策略关联创建/删除 + 策略通知回调
N10	UDM	Nudm_SDM + Nudm_UECM (SBI)	会话订阅数据，SMF 注册
N4	UPF	PFCP (UDP)	PFCP 关联 + 会话建立/修改/删除
N40	CHF	Nchf_ConvergedCharging (SBI)	融合计费：在 PDU 会话生命周期内打开/更新/关闭计费数据请求（CHF 通过 NRF 发现解析，使用静态回退 URI；默认禁用）
N26	PGW-C	内部互通	采用/切换/查找 4G/5G 移动性
n/a	NRF	Nnrf_NFManagement (SBI)	NF 注册 + 心跳（无外部发现）

SBI 端点

所有 SBI 端点都在基础 URL `{SBI_SCHEME}://{SBI_ADDR}:{SBI_PORT}` 下提供。

方法	路径	服务	[
POST	`/nsmf-pdusession/v1/sm-contexts`	Nsmf_PDUSession	创建 SM 上下 建立）
POST	`/nsmf-pdusession/v1/sm-contexts/{smContextRef}/retrieve`	Nsmf_PDUSession	检索 SM 上下 POST，而不[
POST	`/nsmf-pdusession/v1/sm-contexts/{smContextRef}/modify`	Nsmf_PDUSession	更新 SM 上下 UP 状态变化、 修改、释放标志[
POST	`/nsmf-pdusession/v1/sm-contexts/{smContextRef}/release`	Nsmf_PDUSession	释放 SM 上下
POST	`/nsmf-callback/sm-policy-notify/{smContextRef}`	Npcf 回调	接收来自 PCF `SmPolicy` 更新（专用 Qo

创建 **SM** 上下文

必填请求字段：`supi`、`sNssai`、`servingNetwork`、`dnn`、`smContextStatusUri`、`pduSessionId`、`anType`、`servingNfId`。返回 `201 Created` 和 `Location: {base}/nsmf-pdusession/v1/sm-contexts/{ref}`。错误映射：`400 MANDATORY_IE_MISSING`、`404 DNN_DENIED`、`500 SYSTEM_FAILURE`、`503 NF_DISCOVERY_FAILURE`（UDM 无法访问）。

检索 **SM** 上下文

AMF 通过引用当前 SM 上下文来读取它（TS 29.502 §5.2.2.6 使用 `POST .../sm-contexts/{smContextRef}/retrieve`，而不是 `GET`）。SMF 返回 `200 OK`，并为已知引用提供当前 SM 上下文视图，或为未知引用返回 `404 CONTEXT_NOT_FOUND`。

更新 SM 上下文

修改处理程序根据主体中的第一个识别的键进行分发：

主体键	场景
`n2SmInfo` + `n2SmInfoType`	来自 gNB 的 N2 SM 信息（`PDU_RES_SETUP_RSP` 激活隧道；`PDU_RES_REL_RSP` 使其失效）
`upCnxState: "DEACTIVATED"`	AN 释放 / UE 空闲进入（下行 FAR 到 BUFFER）
`upCnxState: "ACTIVATING"`	服务请求；响应携带 N2 `PDU_RES_SETUP_REQ`
`hoState`	切换状态转换：`PREPARING`、`PREPARED`、`COMPLETED`（重新激活隧道）、`CANCELLED`
`qosFlows`	建立后的 QoS 流添加/修改（返回 `204 No Content`；通过 N2 SM 信息传达给 gNB）
`release: true`	AMF 发起的释放（返回 `204 No Content`）
`servingNfId`	移动过程中的 AMF 实例更改（返回 `204 No Content`）

释放 SM 上下文

主体可以包含 `ueLocation` 以记录最终位置。返回 `204 No Content`。



关键程序

PDU 会话建立 (TS 23.502 §4.3.2)

N1N2MessageTransfer 是异步发送的，因此 `201 Created` 首先返回给 AMF。N1 SM 消息是 NAS PDU 会话建立接受，N2 SM 信息是 NGAP `PDUSessionResourceSetupRequestTransfer`，作为 `multipart/related` 主体传递（TS 29.518），其中二进制 NAS/NGAP 部分由 `contentId` 引用。

DNN 配置来源

DNN 的行为在两个地方定义，操作员必须一致地设置这两个地方，以便 DNN 正确启动：

DNN 属性	来源	如何□置	
允许的 S-NSSAI	AMF/UDM 订阅	AMF 在创建 SM 上下文时提供 `sNssai`；DNN 必须在订阅者的 UDM SM 数据中存在于匹配的 S-NSSAI 下	SM NS `dn` 中查[
会话-AMBR	UDM `dnnConfiguration.sessionAmbr`	在 HSS → UDM/UDR 中配置	作为 `se` PC PC
默认 QoS (5QI + ARP)	UDM `dnnConfiguration.5gQosProfile`	在 HSS → UDM/UDR 中配置	缺失[优先[(QI
PDU 会话类型	UDM `dnnConfiguration.pduSessionTypes`	`defaultSessionType` + `allowedSessionTypes`	UE 足，[IP
SSC 模式	UDM `dnnConfiguration.sscModes`	`defaultSscMode` + `allowedSscModes` (TS 23.501 §5.6.9.2)	UE 足，[SS
静态 UE IP	UDM `dnnConfiguration.staticIpAddress`	在 HSS → UDM/UDR 中配置	见 □
UE IP 池 (动态)	**SMF** `ue_ip_pool` `subnet_map`	SMF 配置	按 D 池
DNS / MTU / P-CSCF (ePCO)	**SMF** `dns` / `dns6` / `mtu` / `:pco`	SMF 配置	在建[UE CS 开头[

经验法则：每个用户和每个 **DNN** 策略（**S-NSSAI**、**AMBR**、默认 **QoS**、会话类型、**SSC** 模式、静态 **IP**）存在于 **UDM** 订阅中；交给 **UE** 的网络资源（动态 **IP** 池、**DNS**、**MTU**、**P-CSCF**）存在于 **SMF** 置中。在 SMF `subnet_map` 中存在但在订阅者的 UDM SM 数据中缺失的 DNN 被拒绝，返回 `404 DNN_DENIED`；在 UDM 中存在但没有匹配的 `subnet_map` 条目且没有 `default` 池的 DNN 在 IP 分配时失败，返回 `:no_pool_for_dnn`。

UE IP 地址分

每个 PDU 会话在与 UPF 建立 PFCP 会话之前分配一个 IP 地址（IPv4、IPv6 或两者）。IP 池镜像 4G PGW-C 分配模型。来源按优先顺序尝试：

1. **N26** 采用的地址。对于通过 N26 从 4G 起源会话迁移到 5G（在创建请求中为 `adoptedIpv4`），独立 PGW-C 已分配的 UE IP 被逐字重用；该地址属于 PGW-C 的池，因此在切换过程中保持不变，而不是重新分配。对于双栈会话，IPv6 部分随后动态分配。
2. 静态 **IP**（订阅者固定）。当 UDM SM 订阅数据包含请求 DNN 的 `staticIpAddress` 时，该固定地址在该 DNN 的每个会话中得到满足。镜像了 4G HSS/PGW-C 静态 IP 行为，最终在 HSS 中配置并通过 UDM/UDR 显示。对于双栈会话，IPv6 部分动态分配。
3. 动态池。否则，从 DNN 的配置池中分配地址。分配选择一个随机的空闲主机地址（排除 IPv4 的网络/广播），在冲突时重试，并在所有活动会话中去重。

UE IP 在任何工作线程退出时返回到池中（显式释放、崩溃或主管关闭），因此陷入建立/释放/重试循环的 CPE 不会慢慢泄漏池地址。

DNN 感知池 (`subnet_map`)

池通过 `subnet_map` 按 DNN 选择。键是确切的 DNN 名称、`^regex` 模式或 `:default`；值是 IPv4 和/或 IPv6 CIDR 的列表。匹配优先级为 确切 **DNN** → 正则表达式 → `:default`。对于 `IPV4V6` 会话，SMF 从匹配池的 IPv4 和 IPv6 范围中分配。见 UE IP 池 在配置参考中。

静态 **IP** 字段和保留

`staticIpAddress` 字段遵循 TS 29.503 (`DnnConfiguration`) 和 TS 29.571 (`IpAddress`)：一个对象数组，每个对象可选地携带 `ipv4Addr`（也接受单个对象）。静态地址在池中保留，因此不会动态分配：

- 置范围内 的静态地址计入该池，并排除在动态分配之外。
- 每个置范围外 的静态地址仅用于重复检测，不消耗池容量（静 IP 可以在自己的路由子网中存在）。
- 如果静态地址 已被 其他活动会话使用，则新 PDU 会话被拒绝（`static_ip_in_use`），而不是发出重复，符合 4G PGW-C 的行为。这只应在同一静态 IP 对两个订阅者错误配置时发生。

要配置静态 IP，请在订阅者的 DNN 配置上设置 `staticIpAddress`（通过 HSS → UDM/UDR）；不需要更改 SMF 配置。

PDU 会话类型协调 (TS 24.501 §6.4.1.3)

当请求双栈（`IPV4V6`）会话但匹配的池仅产生一个地址族时，SMF 将信号会话类型协调到它实际分□的地址族，然后构建 N1 建立接受和 N2 `PDUSessionResourceSetupRequestTransfer`：

- 分配了 IPv4，没有 IPv6 → 信号 `IPV4`，5GSM 原因 **#50**（仅允许 **PDU** 会话类型 **IPv4**）。
- 分配了 IPv6，没有 IPv4 → 信号 `IPV6`，5GSM 原因 **#51**（仅允许 **PDU** 会话类型 **IPv6**）。
- 两个族都存在 → 信号 `IPV4V6` 不变。

这是为了互操作性：在仅携带 IPv4 地址的情况下宣传 `IPV4V6` 会导致 RAN 拒绝 N2 资源设置。因此，从仅 IPv4 池提供双栈订阅的运营商将看到这些会话以 IPv4 启动，并带有相应的 NAS 原因，而不是失败。

监控

- `GET /api/ip_pool`：整体 IPv4 容量以及每个 DNN 池的摘要。
- `GET /api/ip_pool/detail`：每个 DNN 池及每个范围的 CIDR、总数/已分配/可用，以及当前分配地址的完整列表（动态 + 静态）。

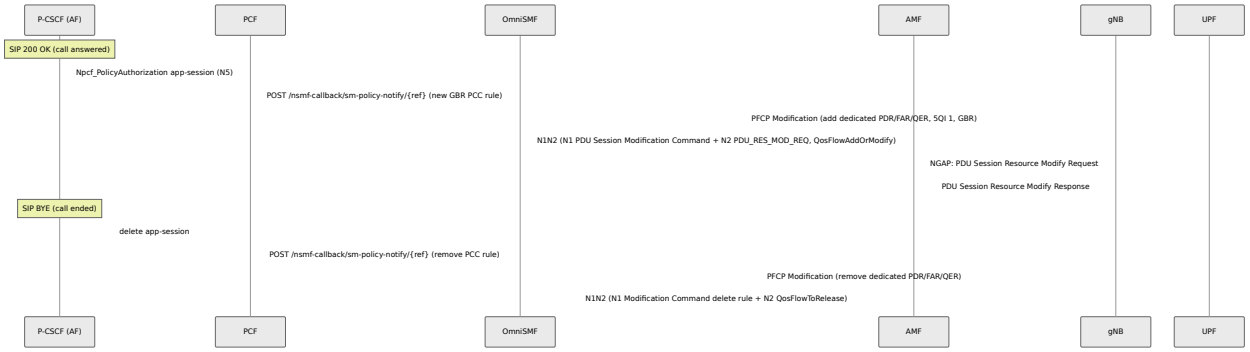
> **IPv6** 注意事项： IPv6 地址由池分配、保留、释放和报告。

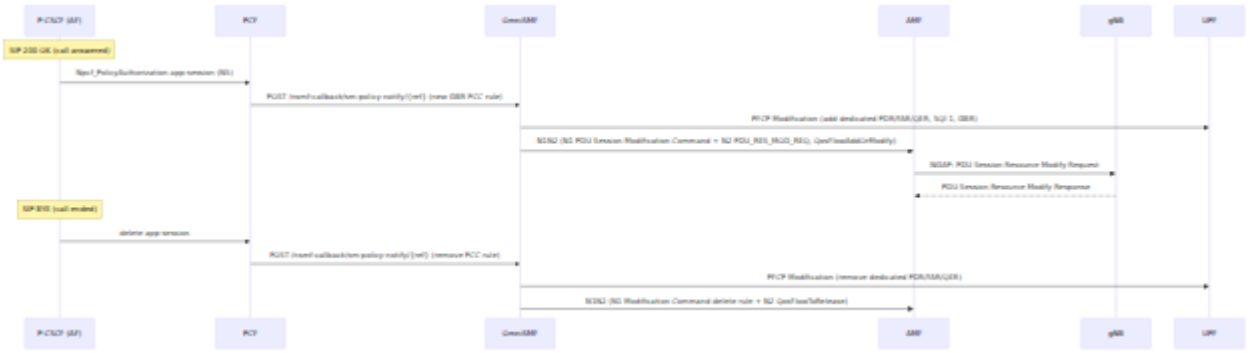
P-CSCF 发现和广告 (VoNR / IMS)

对于 IMS/VoNR PDU 会话，UE 必须学习一个 P-CSCF 以进行注册。SMF 在 PDU 会话建立接受的 **ePCO** 中返回它（P-CSCF IPv4 容器 `0x000C`，IPv6 容器 `0x0001`），每个地址一个容器，与 4G PGW-C 相同。只有在 DNN 以 `ims` 开头时才返回 P-CSCF。选择顺序：

1. **DNS** 发现 + 健康检查（首选）。当 `p_cscf_discovery_enabled` 时，SMF 解析 `_sip._tcp.<fqdn>`（SRV）到 A/AAAA，然后用 SIP `OPTIONS` 健康检查每个候选者（TCP 然后 UDP）。只有回答的服务器才会被广告，并且该集合在固定的 60 秒周期内刷新，因此停止回答 `OPTIONS` 的 P-CSCF 会在大约一分钟内从广告集合中删除，而恢复的 P-CSCF 会在下一个周期重新添加。刷新独立于会话建立运行；新会话始终读取最新的健康集合。
2. 静态回退。如果发现被禁用或返回没有健康的结果，则使用 `p_cscf_ipv4_address_list` / `p_cscf_ipv6_address_list`。

这与 PGW-C 完全相同，因此混合 4G/5G IMS 部署在两个节点上以相同方式配置 P-CSCF。见 P-CSCF 和协议配置选项。





专用 QoS 流 / GBR 承载 (VoNR, 5QI-1)

VoNR 需要一个专用的 GBR QoS 流（5QI 1）用于语音媒体，以及默认的非 GBR 流。支持 静态（预配置）和 动态（呼叫驱动）路径。

静态： GBR 规则在策略中配置（HSS PCRF 收费规则到 UDR `sm-policy-data` 到 PCF PCC 规则），并在 `sm-policies` 决策中返回。它在 PFCP 会话存在时被存储和安装，因此 UPF 从会话启动的那一刻起就用其 5QI/GBR 标记流。

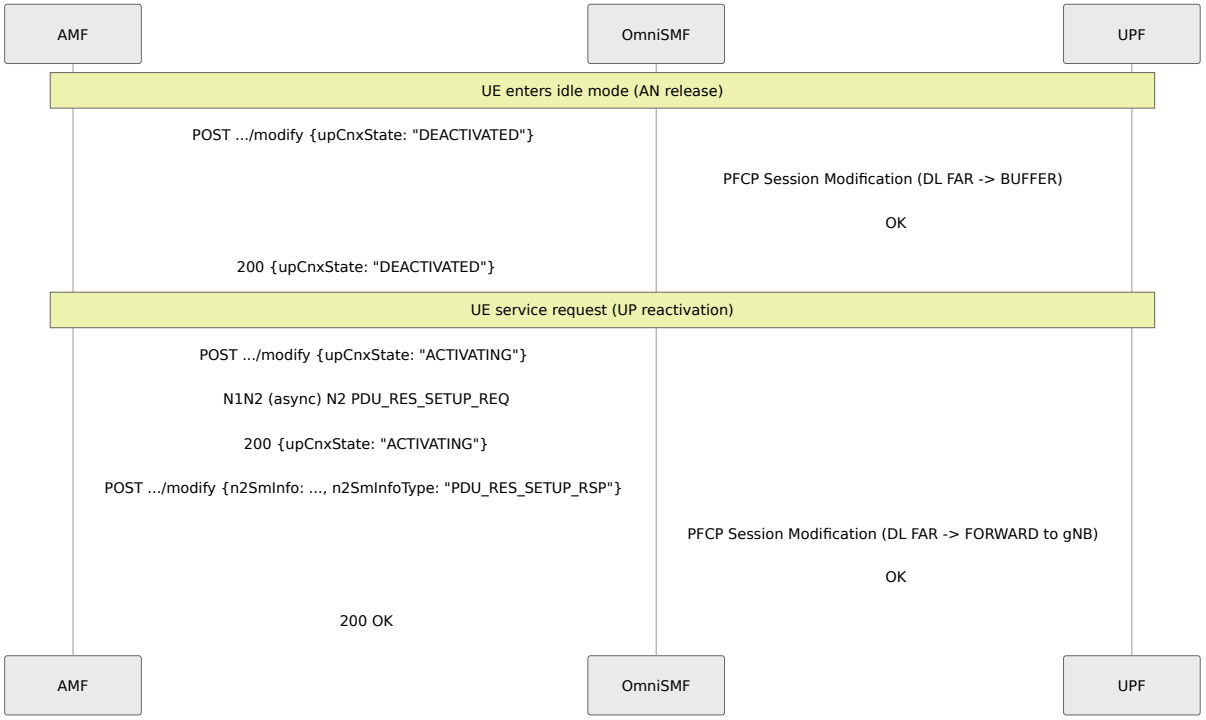
动态（呼叫驱动）： 流在呼叫接通时添加，并在 BYE 时移除，由 PCF 策略通知回调驱动：

通知发送到 `/nsmf-callback/sm-policy-notify/{smContextRef}`（SMF 从主体读取 `smPolicyDecision`）：非空的 PCC 规则安装流，空值规则移除流（TS 29.512，空值 PCC 规则是移除规则）。AF 应用会话与 SM 策略匹配（按 UE IP）在 PCF 上执行。

PDU 会话修改：QoS 流更新

AMF 还可以通过修改端点直接驱动 QoS 流修改，使用 `qosFlows` 列表，而不是通过 PCF。每个条目要么更新现有流（按 `qfi` 匹配），要么分配新的 QFI。UPF 重新编程以更改 QoS，因为 `SmContextUpdatedData` 定义没有 `qosFlows` JSON 成员，因此响应为 `204 No Content`；QoS 更改通过 N2 SM 信息传达给 gNB。

PDU 会话释放 (TS 23.502 §4.3.4)



拆除顺序为 PCF 策略删除 → PFCP 会话删除 → UDM 注销 → SM 上下文状态通知（`resourceStatus: RELEASED`）到 AMF 回调；然后工作线程将 UE IP 释放回池并停止。对于 AMF 发起的 `release: true` 修改，执行相同的拆除。

UE 空闲 / 服务请求：UP 连接状态 (TS 23.502 §4.2.3)

在 `ACTIVATING` 时，SMF 更新 UE 最新建立请求中的 PTI，并向 AMF 发送 N1N2MessageTransfer，以触发 gNB 上的 `PDUSessionResourceSetup`；`200` 响应携带 `upCnxState: ACTIVATING`。用户平面仅在 gNB F-TEID 在后续的 `PDU_RES_SETUP_RSP` 中到达时重新指向 gNB。

切换状态转换

AMF 驱动的切换通过修改端点上的 `hoState` 值推进会话：

- `PREPARING` / `PREPARED`：记录状态；用户平面未更改。
- `COMPLETED`：重新激活隧道（PFCP 修改，下行 FAR 到目标 gNB）并清除切换状态。
- `CANCELLED`：切换状态在没有用户平面更改的情况下被清除。

N4 PFCP 会话结构

每个 PDU 会话在 UPF 上安装以下 PFCP 信息元素 (IE) (TS 29.244)：

IE	方向	目的
PDR (上行)	接入 → 核心	匹配来自 gNB 的 GTP-U 流量在 UPF N3 F-TEID 上
PDR (下行)	核心 → 接入	按 UE IP 地址匹配来自 N6 的流量
FAR (上行)	核心	转发到 N6（无外部头）
FAR (下行)	接入	最初为 BUFFER；在 `PDU_RES_SETUP_RSP` 后更新为 GTP-U FORWARD
QER	双向	强制会话 AMBR（上行和下行 MBR）；每个 GBR 流添加专用 QER
URR	双向	基于时间的使用报告（阈值��自 `heartbeat_interval`）

单个 URR 使用基于持续时间的测量和时间阈值触发。阈值源自 `heartbeat_interval`（其值的六倍，例如在默认的 10 秒心跳下为 60 秒），因此无法独立调节；提高 `heartbeat_interval` 可以延长报告的频率。UPF 发送回的使用报告记录并显示为 `omni_smf_n4_usage_reports_total` 和 `omni_smf_n4_usage_report_volume_bytes_total` 指标（参见 Metrics），并且在启用 N40 收敛计费时，作为临时计费数据请求转发到 CHF。

在发送任何会话请求之前，UPF 的 PFCP 关联（带心跳）已建立。如果关联丢失，可以通过管理 API (`POST /api/oam/pfcp/reassociate`) 重新触发。

N26 EPS 互通 (4G ↔ 5G)

SMF 和独立的 PGW-C 共享一个 UPF，并通过内部 N26 互通路径进行协调，以便 UE 在切换时保持其 IP 地址：

- 采纳 **(4G → 5G)**：从 4G 发起的 EPS 会话被采纳到 5GS，重用 PGW-C 已分配的 UE IP。重新采纳现有会话将返回现有的 5G 会话，而不是创建重复会话。
- 切换：通过 PFCP 修改更新接入隧道，将现有会话的��入侧切换到 `:eps`（5G → 4G）或 `:fivegs`（4G → 5G），同时保持相同的 UE IP 和 UPF 上的相同 PFCP 会话。切换返回 EPS 视图（保留的 UE IP、DNN、会话 AMBR、默认 EPS 承载 ID 和 PGW-U F-TEID），PGW-C 使用该视图来回答 SGW。
- 查找：UE 的活动 PDU 会话可以通过 SUPI/IMSI 作为 EPS PDN 连接视图读取。

管理 API 公开 `POST /api/eps_handover` 以驱动互通。PGW-C S5/S8-C 地址通过所需的 `pgw_c_addr` 配置键进行广告（参见 Configuration Reference）。

← 概述

OmniSMF 故障排除

PDU 会话建立失败，错误代码 404 DNN_DENIED

- 症状：AMF 收到 `404`，原因是 `DNN_DENIED`。
- 原因：UDM SM 订阅数据中不包含请求的 DNN。
- 解决方案：验证 UDM 订阅者数据中的 SUPI 是否包含与请求的 DNN 匹配的 `dnnConfigurations` 条目（匹配不区分大小写）。SMF 日志显示 `dnn_not_found`。

PDU 会话建立失败，错误代码 503 NF_DISCOVERY_FAILURE

- 症状：AMF 收到 `503`，原因是 `NF_DISCOVERY_FAILURE`。
- 原因：SMF 无法连接到 `udm_uri` 的 UDM。
- 解决方案：验证 SMF 主机与 `udm_uri` 之间的连接，以及 UDM SBI 端口是否可达。SMF 日志显示底层原因。

PFCP 会话建立被拒绝

- 症状：UPF 在 PFCP 会话建立响应中返回了一个非接受的原因。
- 原因 / 解决方案：
  - `upf_addr` / `upf_pfcp_port` 不正确：验证 UPF 是否在 PFCP 端口上监听。
  - PFCP 关联未建立：SMF 必须在发送会话请求之前完成 PFCP 关联设置。检查启动日志中的关联消息；如有必要，通过 `POST /api/oam/pfcp/reassociate` 重新触发。
  - `pfcp_addr` 从 UPF 不可路由：将其设置为 UPF 可以到达的地址。

UE 未收到 IP 地址

- 症状：会话在 IP 分配期间失败。
- 原因 / 解决方案：

- `:no_pool_for_dnn`：请求的 DNN 没有匹配的 `subnet_map` 条目且没有 `default`；添加一个。
- `:ipv4_pool_exhausted` / `:ipv6_pool_exhausted`：匹配的范围已满；扩大或添加一个范围。
- `:static_ip_in_use`：相同的 `staticIpAddress` 已分配给两个订阅者；修正订阅数据。
- `:static_ip_invalid`：提供的 `staticIpAddress.ipv4Addr` 不是有效的 IPv4 地址；修正订阅数据。

- 双栈作为单栈服务：来自 IPv4 仅池的双栈请求故意以 IPv4 形式出现，带有 5GSM 原因 #50（见 PDU 会话类型协调）。

会话 AMBR 限制在 ~1 Mbps

- 症状：每个会话的吞吐量大约限制在 1 Mbps，无论订阅的 AMBR 如何。
- 原因：PCF 仅接收到一个非规范的 `sessionAmbr` 键，并且未设置 `subsSessAmbr`，因此回退到其 ~1000 Kbps 默认值。
- 解决方案：SMF 在 SM 策略关联中发送 `subsSessAmbr`（TS 29.512 字段名称）和 `sessionAmbr`，并从 UDM DNN 配置的 `sessionAmbr` 字段读取订阅的 AMBR。验证 UDM DNN 配置是否携带正确的 `sessionAmbr`，并且 PCF 是否遵循 `subsSessAmbr`。见 Session-AMBR 朝向 PCF。

N1N2MessageTransfer 到 AMF 失败

- 症状：SM 上下文和 PFCP 会话已创建，但 UE 从未收到建立接受。
- 原因：异步 N1N2MessageTransfer 到 AMF 失败；没有自动重试。
- 解决方案：SMF 将每个与 UE 关联的 Namf 消息发送到为 UE 服务的 AMF，NF 实例由 Create/Update SM Context 中的 `servingNfId` 指定，在 NRF 中查找。日志行 `Cannot resolve serving AMF <id>` 意味着该 AMF 实例未在 NRF 中注册；请注册它，或检查 NRF 可达性。否则验证 AMF 是否可达。要恢复，释放 SM 上下文并重试建立。

专用 VoNR 流未创建

- 症状：语音通话建立，但没有出现专用的 5QI-1 GBR 流。
- 原因 / 解决方案：
  - PCF 策略通知回调未到达：确认 PCF 配置了通知 URI `/nsmf-callback/sm-policy-notify/{smContextRef}`，并且 AF (P-CSCF) 应用会话已到达 PCF。
  - 没有可用的 QFI：日志显示 `No QFI available for PCC rule ...`；会话中已经存在太多专用流。

- PCC 规则未携带 5QI / GBR：仅具有可解析的 `5qi` 和 GBR 值的��则才会成为专用流��验证策略中的计费规则 / QoS 决策。

高会话计数 / 内存增长

- 症状：会话工作线程计数增长而没有匹配的释放。
- 原因：错过的 AMF/UE 发起的释放（例如，在 AMF 重启后）留下孤立的会话工作线程；没有会话超时。
- 解决方案：使用 `omni_smf_active_pdu_sessions_count` 指标（按 DNN 分类）跟踪活动会话计数，或读取 `GET /api/statistics` 和 `GET /api/pdu_session` 管理端点（见 指标）。会话可以通过 DNN、SUPI、S-NSSAI 或 UPF 通过 `/api/oam/session/by_*` 管理端点排空。当工作线程退出时，UE IP 始终返回到池中，因此排空孤立的工作线程可以回收其地址。

日志关联

日志为结构化 JSON，包含元数据（SUPI、PDU 会话 ID、SM 上下文引用、DNN、过程）。按 SUPI 过滤以追踪订阅者的会话生命周期。通过 `POST /api/oam/log_level` 调整运行时日志级别，无需重启。