← □述

OmniNGAP □作

OmniNGAP 是 Omnitouch 5G 核心的 **NGAP** 协议编解码器。NGAP（NG 应用协议，3GPP TS 38.413）是通过 **N2** 接口在 5G NR 基站（gNB）和接入与移动管理功能（AMF）之间传输的应用层信令协议。OmniNGAP 编译了 TS 38.413 中完整的 NGAP ASN.1 语法，并使用对齐数据包编码规则（**APER**）提供 NGAP 协议数据单元（PDU）的编码和解码，这是规范要求的传输格式。

OmniNGAP 作为构建依赖项被使用：

- **OmniAMF** 将 OmniNGAP 链接到编码和解码它与 gNB 通过 N2 交换的每个消息（NG 设置、初始上下文设置、NAS 传输、寻呼、PDU 会话资源管理、UE 上下文释放、切换等）。
- **OmniSMF** 生成 AMF 在编码之前嵌入 NGAP 消息中的 **N2 SM** 信息容器（例如 *PDU □□□□□□□□□□*）。SMF 从不直接与 N2 通信；它将这些不透明的传输容器交给 AMF，AMF 是 gNB 唯一的 NGAP 对等方。

编解码器的构建□□□式

NGAP 语法表示为 `asn1/` 下的一组 ASN.1 模块。在构建时，这些模块按照依赖顺序编译一次，生成执行 APER 编码和解码的低级编解码器。此编译仅是构建问题 - 在运行时没有任何配置或调整。构成 NGAP 语法的模块包括：

ASN.1 模块	角色
NGAP-CommonDataTypes	跨语法共享的基本类型（Criticality、Presence、ProcedureCode、ProtocolIE-ID）
NGAP-Constants	程序代码、IE 标识符和基数常量分配
NGAP-Containers	通用协议 IE / 协议扩展容器模板
NGAP-IEs	所有 NGAP 信息元素定义，包括 N2 SM 传输容器
NGAP-PDU-Contents	每个程序消息体（InitiatingMessage / SuccessfulOutcome / UnsuccessfulOutcome 载荷）
NGAP-PDU-Descriptions	顶级 NGAP-PDU 类型和基本程序调度表

APER（对齐 PER）是故意使用的：NGAP 根据 TS 38.413 需要对齐编码，因此编解码器以对齐-PER 模式生成，而不是非对齐 PER 或 BER。

PDU 形状：三种消息变体

每个 NGAP-PDU 是三种变体中的一种的选择（TS 38.413 条款 9.3.5）。因此，解码总是产生其中之一，AMF 根据变体进行模式匹配以决定如何继续：

变体	出现时机	意义
`initiatingMessage`	打开一个基本程序	程序的第一条消息（请求、报告或一次性指示）
`successfulOutcome`	关闭一个成功的程序	对发起消息的积极响应（对于期望响应的程序）
`unsuccessfulOutcome`	关闭一个失败的程序	负响应，携带 **Cause**（见下方 Cause 参考）

每个变体是三个字段的 SEQUENCE: **procedureCode**（这是哪个基本程序 - 见下方代码表）、消息的 **criticality** 和 **value**（程序特定的消息体）。类 2 程序（如寻呼、下行 NAS 传输、错误指示）没有成功/失败结果 - 它们仅由

`initiatingMessage` 组成。类 1 程序（如 NGSetup 或 InitialContextSetup）有一个或两个结果。

Criticality 和 Criticality Diagnostics

每个 IE 和每条消息都携带一个 **criticality**，其值来自 `Criticality ::= ENUMERATED { reject, ignore, notify }`（TS 38.413 条款 9.3.1）。Criticality 告诉接收方在遇到其 ID 或类型不理解或无法在当前状态下处理的 IE（或整个消息）时该怎么做：

Criticality	对于不理解/未处理项目的接收方行为
`reject`	拒绝整个程序 - 该项目是必需的，因此无法处理消息
`ignore`	静默跳过该项目并处理消息的其余部分
`notify`	在没有该项目的情况下处理消息，但将该项目报告回发��方的 **Criticality Diagnostics**

Criticality Diagnostics（ProtocolIE-ID 19）是接收方添加到其响应或错误指示中的 IE，以告诉发送方哪些 IEs 被拒绝、忽略或通知，以及原因。它是两个端点在不丢弃关联的情况下调和版本或能力不匹配的方式。

OmniNGAP 像处理 PDU 的其他部分一样编码和解码 criticality 字段和 Criticality Diagnostics IE。它并不直接应用拒绝/忽略/通知政策 - 决定如何处理未知或处理不当的 IE，并构建结果的 Criticality Diagnostics，是 AMF 中的程序逻辑。

3GPP 角色和规范参考

规范	相关性
TS 38.413	NGAP 协议 - 消息集、信息元素、程序代码、原因值、APER 编码
TS 38.412	N2 信令传输 - NGAP 通过 SCTP 的承载
TS 23.502	通过 N2 运行的 5G 系统程序（注册、PDU 会话建立/修改/释放、寻呼、切换）
TS 23.501	5G 系统架构 - gNB 和 AMF 之间 N2 参考点的位置
ITU-T X.691	数据包编码规则（PER），NGAP 在对齐形式中使用的编码系列

编译的 ASN.1 对应于 TS 38.413 的 **Rel-17**（**h** 系列） NGAP 语法，因此完整的 Rel-17 消息集 - 包括 UE 上下文恢复/挂起、RAN 早期状态传输和 MBS（多播/广播）程序 - 在编译的编解码器中存在。

支持的 **NGAP** 程序

OmniNGAP 编译了 TS 38.413 定义的 整个 **NGAP** 基本程序集。任何这些程序的任何格式良好的 NGAP PDU 都可以被解码，任何结构有效的术语都可以被编码 - 编解码器并不限制自己到一个子集。实际中执行的内容由消费 NF（主要是 AMF）驱动。

下表列出了与 AMF/SMF 操作最相关的程序及其在编译的 `NGAP-Constants` 模块中分配的 实际 程序代码。方向是 *initiating* 消息的方向。

程序	代码	方向	目的
NGSetup	21	gNB → AMF	建立 NG（N2）关联；交换 gNB/AMF 身份、PLMN 和切片支持
RANConfigurationUpdate	35	gNB → AMF	gNB 更新已建立关联上的配置
AMFConfigurationUpdate	0	AMF → gNB	AMF 更新其配置 / TNL 关联信息
NGReset	20	任意	重置部分或全部 NG 关联
InitialUEMessage	15	gNB → AMF	UE 的第一条 NAS 消息，打开与 UE 关联的信令连接
DownlinkNASTransport	4	AMF → gNB	向 UE 传输 NAS PDU
UplinkNASTransport	46	gNB → AMF	从 UE 传输 NAS PDU
InitialContextSetup	14	AMF → gNB	在 gNB 上建立 UE 上下文、安全性和（可选）PDU 会话资源
UEContextModification	40	AMF → gNB	修改已建立的 UE 上下文
UEContextReleaseRequest	42	gNB → AMF	gNB 请求释放与 UE 关联的连接
UEContextRelease	41	AMF → gNB	AMF 命令释放 UE 上下文（命令 / 完成）
Paging	24	AMF → gNB	在跟踪区域中寻呼处于空闲模式的 UE

程序	代码	方向	目的
PDUSessionResourceSetup	29	AMF → gNB	在 gNB 上设置 PDU 会话资源（携带 SMF N2 SM 容器）
PDUSessionResourceModify	26	AMF → gNB	修改现有 PDU 会话资源
PDUSessionResourceModifyIndication	27	gNB → AMF	gNB 发起的 PDU 会话资源修改
PDUSessionResourceRelease	28	AMF → gNB	在 gNB 上释放 PDU 会话资源
PDUSessionResourceNotify	30	gNB → AMF	通知 QoS 流/会话状态变化
HandoverPreparation	12	源 gNB → AMF	基于 NG 的切换：源 gNB 请求切换准备
HandoverResourceAllocation	13	AMF → 目标 gNB	基于 NG 的切换：请求目标 gNB 的资源
HandoverNotification	11	目标 gNB → AMF	目标 gNB 确认 UE 到达
HandoverCancel	10	源 gNB → AMF	取消正在进行的切换
PathSwitchRequest	25	gNB → AMF	基于 Xn 的切换：将 N3 用户平面路径切换到目标 gNB
ErrorIndication	9	任意	报告未与特定程序相关的协议错误
UEContextResume	58	gNB → AMF	恢复挂起的（RRC-INACTIVE）UE 上下文

程序	代码	方向	目的
UEContextSuspend	59	gNB → AMF	挂起 UE 上下文

> 其他 TS 38.413 程序都存在于编译的语法中，并且可以被编码/解码，但不属于 AMF 的核心连接管理和会话管理流程。这些包括（此列表是示例性的，而非详尽的）NRPPa 传输、跟踪管理、警告消息广播（PWS）、位置报告、RIM 信息传输、次要 RAT 使用报告、MBS 广播/多播程序、过载开始/停止、AMF 状态指示、NAS 重定向和非交付指示、RRC 非活动过渡报告、RAN 和 RAN 早期状态传输以及私有消息。编译的语法定义了该版本的完整基本程序集，因此 TS 38.413 Rel-17 中的任何程序都可以编码，即使它未在此列出。

关键 NGAP 流

以下序列显示了 OmniNGAP 执行编码/解码的位置。编解码器是 AMF 的 N2 边缘的翻译层；程序逻辑存在于 AMF 中（对于 SM 容器，存在于 SMF 中）。

NG 设置

关联建立，在每个 gNB–AMF 关联上运行一次，在任何 UE 流量之前。

如果 AMF 无法为 gNB 提供服务（例如不支持的 PLMN 或切片），它会编码一个 **NGSetupFailure**，携带一个 **Cause** 值（见下方 Cause 参考）。

N2 上的 PDU 会话资源设置

在 PDU 会话建立期间运行。SMF 构建 N2 SM 信息容器；AMF 将其包装在 NGAP 消息中，OmniNGAP 编码整个 PDU。

编解码器将每个 SM 信息容器视为 PDU 会话资源消息内部的嵌套传输结构。AMF 不解释 SM 容器的内容 - 它在 gNB 和 SMF 之间不加修改地转发它们。

不透明载荷：NAS PDUs 和 SM 容器

两种不同的载荷家族通过 N2 接口交叉，而 NGAP 不查看它们的内部，且两者在 ASN.1 中建模为 `OCTET STRING`：

- **NAS PDUs** - NAS 消息（注册、认证、PDU 会话建立请求等）作为不透明的 `NAS-PDU` OCTET STRING 在 NGAP 中传输。它出现在 NAS 传输程序（InitialUEMessage、DownlinkNASTransport、UplinkNASTransport）中，也可以在 InitialContextSetup 和 PDU 会话消息中传输。NGAP 将其解码为原始字节；AMF 将这些字节传递给其 NAS 层（或将其转发给 UE） - NGAP 从不解析 NAS。
- **SM** 信息容器 - SMF 构建的 PDU 会话传输结构，如上所述，同样作为 OCTET STRING 在 gNB 和 SMF 之间传输。

实际后果对两者是相同的：OmniNGAP 确保 □□ PDU 格式良好，并逐字返回（或传输）内部字节。无论这些内部字节是有效的 NAS 消息还是有效的 SM 容器，都是 NAS 层和 SMF 各自的关注，而与 NGAP 编解码器无关。这就是为什么 NAS PDU 或 SM 容器内部的故障从不表现为 NGAP 编码/解码错误。

NGAP 原因值

每个 NGAP 失败或释放消息都携带一个 **Cause**，这是一个 CHOICE，选择 TS 38.413 中定义的五个原因组之一（条款 9.3.1.2）。OmniNGAP 编码和解码所有五个组及其枚举值。`Cause` CHOICE 是：

原因组	意义	典型用途
radioNetwork	无线网络层原因	切换结果、无线链路丢失、用户不活动、资源不可用、切片不支持
transport	传输网络层原因	传输资源不可用
nas	NAS 层原因	正常释放、认证失败、注销
protocol	NGAP 协议错误	抽象/传输语法错误、消息与接收方状态不兼容
misc	杂项原因	过载、硬件故障、运维干预、未知 PLMN/SNPN

每个组内的代表性枚举值（均按 TS 38.413 条款 9.3.1.2）：

组	值	意义
radioNetwork	successful-handover	切换成功完成
radioNetwork	handover-cancelled	切换被源侧中止
radioNetwork	user-inactivity	由于 UE 不活动而释放连接
radioNetwork	radio-connection-with-ue-lost	与 UE 的无线链路丢失
radioNetwork	radio-resources-not-available	gNB 没有无线资源满足请求
radioNetwork	unknown-PDU-session-ID	引用的 PDU 会话在对等方未知
radioNetwork	slice-not-supported	请求的 S-NSSAI 不被支持
radioNetwork	ims-voice-eps-fallback-or-rat-fallback-triggered	为 IMS 语音回退到 EPS/其他 RAT
radioNetwork	release-due-to-ngran-generated-reason	NG-RAN 发起的释放
radioNetwork	release-due-to-5gc-generated-reason	5GC 发起的释放
transport	transport-resource-unavailable	没有可用的传输（N3/TNL）资源
transport	unspecified	未指定的传输层故障
nas	normal-release	正常的 NAS 触发释放
nas	authentication-failure	UE 认证失败
nas	deregister	UE 注销
protocol	transfer-syntax-error	PDU 无法解码（传输语法错误）

组	值	意义
protocol	abstract-syntax-error-reject	语义上无效的 PDU，被拒绝
protocol	message-not-compatible-with-receiver-state	消息对对等方当前状态无效
protocol	semantic-error	在其他可解码 PDU 中的语义错误
misc	control-processing-overload	对等方过载
misc	hardware-failure	对等方硬件故障
misc	om-intervention	运维干预
misc	unknown-PLMN-or-SNPN	对等方未识别的 PLMN/SNPN
misc	unspecified	未指定的杂项故障

radioNetwork 组是最大的，包含完整的 Rel-17 集（切换计时器到期、NPN/CAG 拒绝访问、RedCap 不支持、与 MBS 相关的原��等）。上述值是在 AMF 连接和会话管理流程中最常见的。

集成

AMF 如何使用 OmniNGAP

- 入站：在 N2 关联上接收到的 SCTP 载荷被传递到编解码器的解码入口点，该入口点返回结构化的 NGAP PDU（三种 消息变体 之一 - `initiatingMessage`、`successfulOutcome` 或 `unsuccessfulOutcome`）或标记错误。格式错误或截断的 PDU 返回错误，而不是使 AMF 崩溃，并记录下来，以便坏的对等方消息不会静默。
- 出站： AMF 为其正在运行的程序构建 PDU 术语，然后调用编解码器的编码入口点，以获取写入 SCTP 关联的对齐-PER 编码。编码器将结果作为 iodata（一个 iolist）返回，调用者在传输之前将其扁平化为二进制。
- AMF 拥有所有程序语义 - IE 填充、可选性和 criticality 处理。OmniNGAP 仅保证结构有效的术语编码为符合规范的二进制，并且符合规范的二进制解码为相应的术语。

SMF 如何使用 OmniNGAP

SMF 不直接调用编解码器进行 N2 传输。它构建 **N2 SM** 信息传输容器（例如 *PDU □□□□□□□□□□，... □□□□□□，... □□□□□□*），AMF 将这些容器嵌入到相应的 PDU 会话资源 NGAP 消息中。这些容器□□□□□□不透明的 OCTET STRING 在 NGAP 层传输，并通过 AMF 使用 OmniNGAP 作为封装 PDU 的一部分进行编码/解码。

← 概述

OmniNGAP 故障排除

由于 OmniNGAP 是一个编译时编解码库，没有运行时界面，因此没有任何配置、重启或抓取的内容。它的可观察行为仅限于它如何处理由消费 NF（AMF）提供的二进制文件。N2 的信令级可观察性来自 AMF，而不是来自这个库。

解码格式错误或截断的 PDU

格式错误或截断的 PDU 返回标记错误，而不是使消费 NF 崩溃。当 gNB 发送编解码器无法解码的消息时：

- 解码入口点返回标记错误而不是抛出异常，因此错误的对等消息永远不会使 AMF 崩溃。
- AMF 记录失败，因此错误的对等消息永远不会是静默的。
- 对等方的相应 NGAP 层信号是 `protocol` 组的 **Cause** - 例如 `transfer-syntax-error`（PDU 无法解码）或 `abstract-syntax-error-reject`（PDU 可解码但语义无效）。有关完整集，请参见 NGAP 原因值 参考。

解码锚定在缓冲区中的第一个 PDU：一旦读取了完整且格式良好的 PDU，后面的任何字节都会被忽略而不是拒绝。NGAP 通过 SCTP 每条消息传递一个 PDU，因此在正常操作中不期望有尾随数据；在对等方认为它发送了额外内容时成功的解码指向对等方或 SCTP 层中的帧故障，而不是编解码器中的故障。

编码失败

OmniNGAP 仅保证 ASN.1 的术语编码为符合规范的二进制。IE 的填充、可选性和 关键性 是 AMF 的责任。因此，编码错误指向 AMF 构建的术语（缺少强制 IE、值超出其约束范围或格式错误的嵌套 SM 容器），而不是编解码器本身。

不透明负载问题（NAS PDU 和 SM 容器）

NGAP 以 不透明 OCTET STRING 的形式携带两种类型的负载，而不进行检查：NAS PDU 和 SMF 生成的 N2 SM 信息容器。OmniNGAP 不解释这两者。

- **SM** 容器 - 如果会话管理传输在端到端中看起来损坏，但封闭的 NGAP PDU 编码和解码正常，则故障在于 SMF 构建容器的方式或 AMF 放置它的方式，而不是 NGAP 编解码器。

- **NAS PDU** - 同样，如果 NAS 消息格式错误，但封闭的 NGAP PDU（InitialUEMessage、DownlinkNASTransport、UplinkNASTransport 等）格式良好，则故障在于 NAS 层，而不是 NGAP。

一般规则：封闭 PDU 的干净编码/解码意味着 NGAP 层完成了它的工作 - 在不透明负载的所有者（NAS 层或 SMF）中查找损坏。