

# 技 术 文 件

技术文件名称： NGN PPLAT SIP V1.01.10\_系统方案

技术文件编号：

版 本： V1.01.10

拟 制 SIP 平台项目组

审 核 丁 学 新

会 签 苏 登 军

管 建 池

标准化 黄 纯 娟

批 准 谷 布

中兴通讯股份有限公司

## 修改记录

说明:

发起的项目包括: CSCF, MGCF, PSS, PPLAT;

影响的项目选择包括: CSCF, MGCF, PSS;

修订编号	文件编号	版本号	拟制/修改人	拟制/修改日期	更改理由	主要更改内容 (写要点即可)	发起的项目	影响的项目	合入版本	合入时间
		V1.0 1.10	管建池	2008-5-26	无	无	PSS			
		V1.0 1.10	管建池	2008-6-20	增加	合入承载和支撑部分	PSS			
		V1.0 1.10	管建池	2008-6-24	增加	添加内存分配器部分	PSS			
		V1.0 1.10	管建池	2008-6-24	增加	添加 IPSec 和信令压缩相关分发部分	PSS			
		V1.0 1.10	管建池	2008-7-13	增加	增加功能描述和方案优缺点	PSS			
		V1.0 1.10	张政山	2008-7-14	修改支撑接口中的有关 ATCA 的部分	10.3	PSS			
0001		V1.0 1.10	张志海	2008-12-23	增加	5.4.2.2.11 中增加: SIP 平台采用分布式部署情况下的分发考虑	CSCF			
0002		V1.0 1.10	张志海	2008-12-24	增加	5.8.5 添加 record-route 与修改 record-route 的兼容考虑。	CSCF	PSS		
0003		V1.0 1.10	张志海	2008-12-24	增加	5.8.6 链路、IP-PORT 变更通知的兼容考虑。	CSCF	PSS		
0004		V1.0 1.10	张志海	2008-12-24	修改	10.2.2.4 PPLAT_SIP_DB_get LocalIPByDstIp 中增加一个参数 PPLAT_SIP_EXTERNAL_INFO_T *ptExternalInfo	CSCF	PSS		
0005		V1.0 1.10	张志海	2009-01-05	增加	10.2.2.5 增加 PPLAT_SIP_TP_UpdateExternalInfo 接口	CSCF	PSS		
0006		V1.0 1.10	张志海	2009-01-05	增加	10.2.2.5 增加 PPLAT_SIP_DB_GetSubAccessAddr 接口	CSCF	PSS		
0007		V1.0 1.10	张志海	2009-01-05	增加	增加 11 章接口变更说明	PPLAT			
0008		V1.0 1.10	张志海	2009-01-05	修改	5.4.2.2.9 多实例下的主备倒换考虑	PPLAT			
0009		V1.0 1.10	郑福青	2009-01-05	增加	增加 10.3.3 人机命令	PPLAT			

						接口				
0010		V1.0 1.10	管建池	2009-04-02	修改	8.1.1.5-8.1.2.3 中的异常流程中，XX 管理进程选择 XX 处理进程失败，都误写成了选择处理进程成功。 CQ : CRDCM00053024	PPLAT			
0011		V1.0 1.10	鲍雯雯	2009-4-22	修改	增加 UA 模式下对 max-forwards 处理的说明	PPLAT			
0012		V1.0 1.10	徐源远	2009-4-29	增加	8.1.1.8 PD-F-PPLAT-SIP-TUUI-0080 UA 模式下初始请求携带会话检测刷新方参数 CQ: CRDCM00094564	PPLAT			
0013		V1.0 1.10	曾丽君	2009-5-22	增加	5.8.5.2 record-route 优化 CQ: CRDCM00147935	PSS			
0014		V1.0 1.10	张政山	2010-01-23	在 10.4.4.2.2.1 中添加对 PPLAT_SIP_OSS_SendAsynMsg_Ext_wrap 的描述	611000419308	PPLAT			

## 目 录

引言 .....	7
1.1    编写目的 .....	7
1.2    预期的读者和阅读建议 .....	7
1.3    文档约定 .....	7
2 术语、定义和缩略语 .....	7
2.1    术语、定义 .....	7
2.2    缩略语 .....	7
3 设计依据 .....	8
4 概述 .....	8
5 设计原理 .....	8
5.1    系统构架的考虑 .....	8
5.2    动态特性的考虑 .....	9
5.3    内存分配器的实现原理 .....	9
5.4    对 ATCA 支持的考虑 .....	11
5.4.1    ATCA 上实现应用的几种方案 .....	11
5.4.2    SIP 平台的 ATCA 方案 .....	15
5.5    多实例情况下信令压缩相关分发流程 .....	32
5.6    多实例情况下 IPSec 相关分发流程 .....	32
5.7    进程启动考虑 .....	34
5.8    兼容性考虑 .....	35
5.8.1    SIP 平台对外接口 .....	35
5.8.2    SIP TP、SIP TR/TU 上电启动 .....	35
5.8.3    JID 与原有 PID 兼容 .....	35
5.8.4    SIP 平台向上层应用分发 .....	36
5.8.5    Record-route 相关处理 .....	36
5.8.6    MAX-FORWARDS .....	37
5.8.7    链路、IP-PORT 变更的通知 .....	37

5.8.8	缺省实现接口 .....	37
5.9	扩展性的考虑.....	37
5.10	系统性能的考虑.....	37
5.11	安全性的考虑.....	38
5.12	部署考虑.....	38
6	构架说明 .....	39
6.1	组件及接口概览.....	39
6.2	组件功能描述.....	39
6.2.1	TU 层功能描述 .....	39
6.2.2	TR 层功能描述 .....	39
6.2.3	TP 层功能描述 .....	39
6.2.4	PARSER 模块功能描述.....	39
6.2.5	SIGCOMP 模块功能描述 .....	39
6.2.6	SIPOMM 功能描述.....	39
6.2.7	SIP TP 管理进程功能描述.....	39
6.2.8	SIP TR/TU 管理进程功能描述.....	40
7	组件标准化设计 .....	40
8	协作说明 .....	40
8.1	功能协作.....	40
8.1.1	TU 层处理 .....	40
8.1.2	TP 层处理.....	49
8.1.3	分发处理.....	52
8.1.4	内存分配器 .....	58
8.1.5	OMM 处理 .....	60
8.2	非功能协作 .....	68
9	组件说明 .....	68
9.1	TU 层处理.....	68
9.1.1	功能需求.....	69
9.1.2	资源限制需求 .....	74

9.1.3	其他需求.....	74
9.2	TP 层处理.....	74
9.2.1	功能需求.....	74
9.2.2	资源限制需求 .....	75
9.2.3	其他需求.....	75
10	接口说明 .....	75
10.1	用户接口 .....	75
10.1.1	SIP 编解码接口 .....	75
10.1.2	用户接口消息 .....	75
10.2	数据库接口 .....	76
10.2.1	SD-I-PPLAT-SIP-DBA-0010 分发接口 .....	76
10.2.2	SD-I-PPLAT-SIP-DBA-0020 配置接口 .....	81
10.3	操作维护接口.....	86
10.3.1	SD-I-PPLAT-SIP-OMC-0010 SIP 平台信令跟踪接口 .....	86
10.3.2	SD-I-PPLAT-SIP-OMC-0020 SIP 平台失败观察接口 .....	87
10.3.3	SD-I-PPLAT-SIP-OMC-0030 SIP 人机命令接口.....	88
10.4	系统支撑接口.....	89
10.4.1	采用 ATCA 后与原来接口相比需要修改的支撑封装函数/接口列表	89
10.4.2	采用 ATCA 后与原来接口相比需要修改的支撑封装函数/接口和参数说明	90
10.4.3	采用 ATCA 后与原来接口相比需要修改的支撑函数/接口相关结构体封装	97
10.4.4	不需要修改的支撑封装函数/接口列表 .....	98
10.4.5	不需要修改的支撑封装函数/接口和参数说明.....	98
10.4.6	支撑函数/接口相关宏定义封装 .....	113
10.4.7	支撑函数/接口相关结构体封装 .....	115
10.5	IP 承载接口 .....	115
10.5.1	IP 承载封装函数/接口列表 .....	115
10.5.2	IP 承载封装函数/接口和参数说明 .....	116
10.5.3	IP 承载函数/接口相关宏定义封装 .....	129

10.5.4	IP 承载函数/接口相关结构体封装 .....	129
10.6	SIP 平台内部接口 .....	130
10.6.1	SLB 与 TP 层间的用户接口 (SLBUI) 定义 .....	130
10.6.2	内存分配器接口 .....	132
11	接口变更说明 .....	134
11.1	PPLAT_SIP_DB_getLocalIPByDstIp 接口 .....	134
11.2	PPLAT_SIP_TP_UpdateExternalInfo 接口 .....	135
11.3	PPLAT_SIP_DB_GetSubAccessAddr 接口 .....	135
12	可挖掘的专利点 .....	135
13	参考文献 .....	135

## 引言

### 1.1 编写目的

本文描述的对象是《NGN PPLAT SIP V1.01.10\_子系统方案》（后文简称“本系统”或“系统”），通过对设计的思路、系统的构架、组成本系统的组件间的协作、组件的具体要求和组件间接口的详细描述，来满足上游需求的要求，同时为组成本系统的每个组件的进一步设计、集成测试的设计、编写用户文档等工作提供依据。

### 1.2 预期的读者和阅读建议

本文预期的读者和阅读建议见表 1.1。

表 1.1

读者分类	阅读重点	备注
系统工程师（负责软件需求开发）	协作说明	
系统工程师（负责同层次相关子系统的设计）	设计原理	
系统工程师（负责组件的设计）	构架说明、协作说明、组件说明、接口说明	
软件工程师（负责组件的详细设计）	构架说明、协作说明、组件说明、接口说明	
测试工程师	构架说明、协作说明、接口说明	
文档工程师	接口说明	

### 1.3 文档约定

本文使用了如下的文档约定：

- 1) 插图一律使用 MS Visio 2003 中文版绘制，并一律“嵌入”于需求描述正文中，而非“浮于文字上方”。
- 2) 用同号、同体但加粗的文字来强调需要读者重视的内容。

## 2 术语、定义和缩略语

### 2.1 术语、定义

通用术语、定义见《NGN PPLAT V1.00.10\_SIP 协议缩略语术语说明.doc》。

### 2.2 缩略语

本文使用的专用缩略语见表 2.2。

表 2.2

缩略语	原文	中文含义
SIP	Session Initiation Protocol	会话发起协议
CSCF	Call Session Control Function	呼叫会话控制功能
MGCF	Media Gateway Control Function	媒体网关控制功能



缩略语	原文	中文含义
PSS	PSTN Simulation Subsystem	PSTN 模拟子系统
MSCe	Mobile Switching Center emulation	移动交换中心仿真
IMS	Ip Multimedia Subsystem	IP 多媒体子系统
TU	Transaction User Layer	事务用户层
TR	Transaction Layer	事务层
TP	Transport Layer	传输层
SIGCOMP	Signaling Compress	信令压缩
OMM	SIP Operation & Maintenance Module	SIP 操作维护模块
DBA	SIP DataBase Adapter Module	DB 适配模块
IIF	SIP Internal InterFace Module	SIP 内部接口模块

### 3 设计依据

本文涉及的设计依据见表 3.1。

表 3.1

文件编号	文件名称	版本号	说明
	NGN PPLAT V1.00.10_软件需求说明书.doc	V1.01.10	上游文档

### 4 概述

协议平台项目范围包括 SIP 协议（协议标准为 RFC3261 等），H248 协议（MGC 功能部分，表示为 H248S；协议标准为 RFC3525，ITU-T H.248.x 系列等），DIAMETER 协议（协议标准为 RFC3588 等）。

目前各协议应用的项目有固网产品 SS1B, PSS1.0/2.0, SBC/BGW 等, IMS 产品 CSCF, MGCF, PDF, HSS 等; 移动核心网产品 CDMA2000 3GCN MSCe, WCDMA/TD-SCDMA MSC Server 等; 应用于的产品众多, 项目研制及维护任务日重, 急需将此三种协议进行平台化, 以便人力共享, 经验共享, 成果共享, 准确快速的响应产品需求。

协议平台经多次研究讨论确定为分期实现, 一期 V1.00.10 规划实现 SIP 协议的平台化, 本文主要对协议平台 SIP 协议部分的软件需求情况统一进行描述。

SIP 协议（Session Initiation Protocol）为会话发起协议, 是基于文本方式的应用层协议, 用于建立, 修改, 以及终止 IP 网络的多媒体会话。

## 5 设计原理

### 5.1 系统构架的考虑

参见《NGN PPLAT V1.00.10\_系统方案》

本文中的所有信息均为中兴通讯股份有限公司内部信息, 不得向外传播。

## 5.2 动态特性的考虑

随着公司 ATCA 平台及多核系统的推出，多核系统架构下编程需要考虑的并发性问题增多。经与相关平台及产品的多次讨论，协议平台对新平台 ATCA 多核系统的支持考虑采用进程多实例的方式实现，具体原理如图 5.2.1 所示。

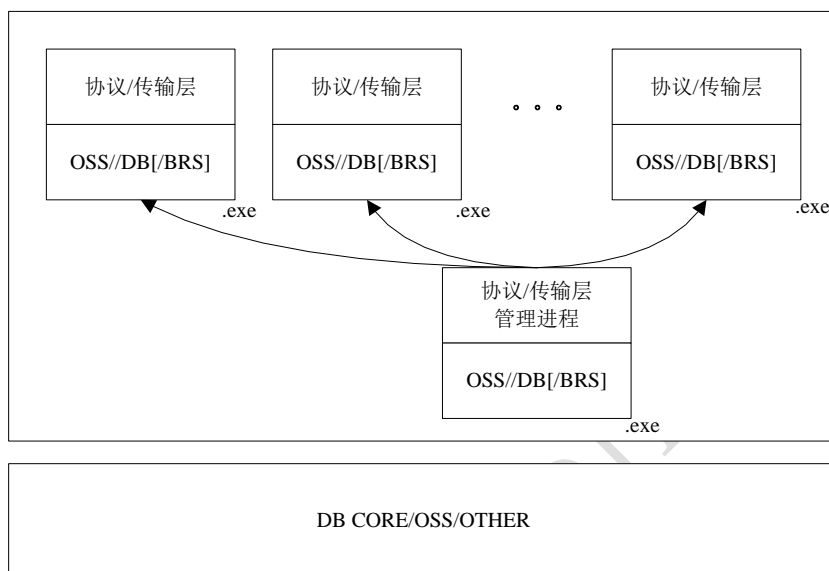


图 5.2.1 协议平台进程多实例示意图

对于 SIP 协议按其功能主要划分成协议层（TU/TR）和传输层（TP）两部分，每部分单独组织成执行进程。其中引入管理进程，实现具体功能层的内部分发部署功能，同时对外隐藏内部实现，实现类似代理的功能。

在此基础上，对于几个比较明确的部分，如全局变量，静态变量，接口函数，共用资源等进行相应考虑，减少全局变量的使用，并进行相关变量的集合定义方式，以便后续版本能够比较顺利的进行加锁操作和多线程改造。另外对于编解码模块存在多处应用进行函数调用的情况，考虑按多实例的方式进行设计和实现，以库函数的方式提供给编解码用户，避免给多核支持的启动带来障碍。

这种方案采用了进程多实例的方式，每个进程都有各自的代码空间，数据空间，对于数据区，全局变量，静态变量，临界区等共用资源的并发不再强制要求。从而对于原有公司产品的软件架构冲击较小，对研发团队的编程技术水平要求降低，能够比较好的实现原有代码的继承，能够平稳的实现多核系统的应用和推广，有利于移植平台后各产品的快速稳定，从而实现节省人力节省成本。

## 5.3 内存分配器的实现原理

SIP 平台实现内存分配器，SIP 平台各进程或者模块可以创建或者销毁内存分配器，在需要使用内存的时候向内存分配器申请内存资源。

SIP 平台可以使用内存分配器来协助实现多实例。

内存分配器的实现原理如下图：

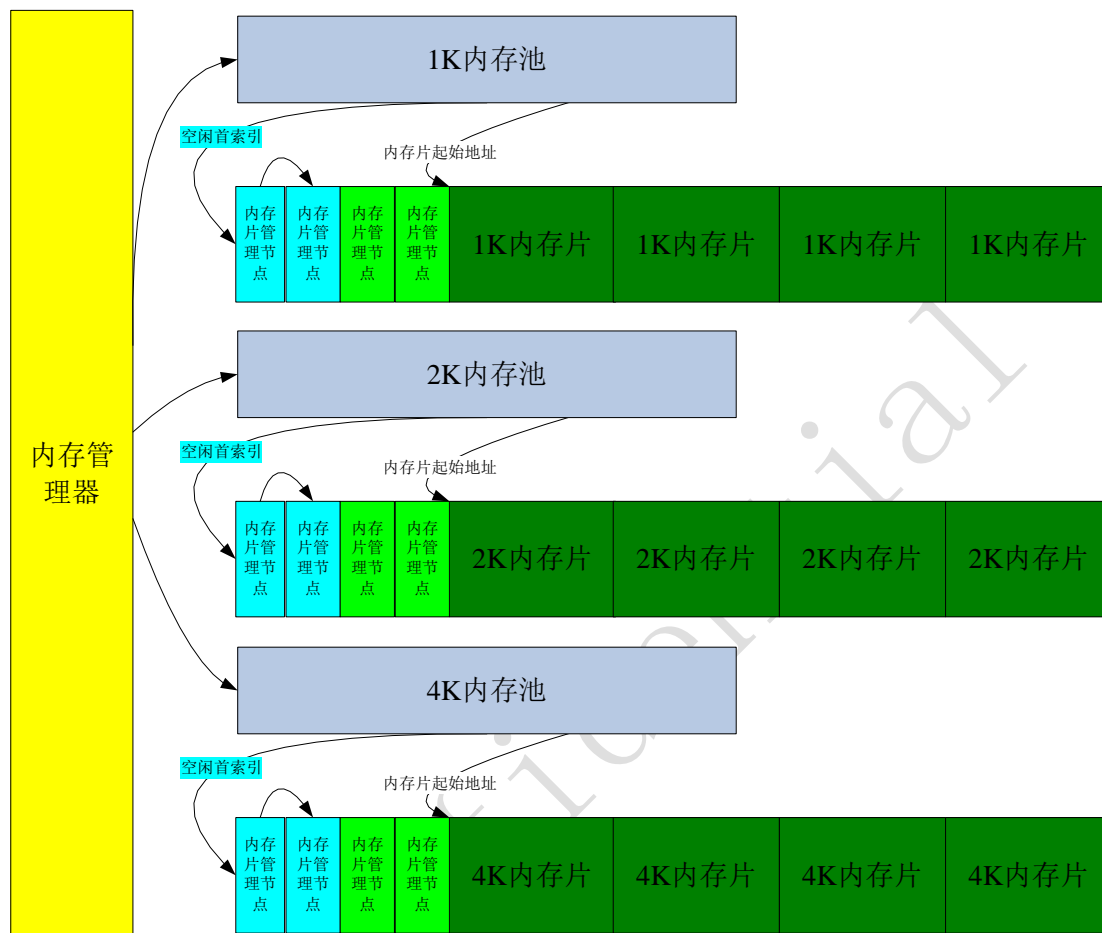


图 5.3.1

内存分配器管理的内存资源被分为两部分，一部分用于内存篇管理节点，一部分用于内存片，内存片管理节点和内存片一一对应，内存管理节点内维护着对应内存片的占用状态。空闲的内存片管理节点串连在一起形成一个单向链表。

内存分配器采用分级管理，等级初步分为 1K，2K，4K，8K，16K，32K 等六个内存池，各等级内存池内内存片的个数由调用者在创建内存分配器时指定。

内存分配器管理的内存资源可以由外部传入也可以由内存分配器自行申请，这由调用这在创建内存分配器时指定。在创建内存分配器时，如果指示内存由外部传入，而传入的内存又不足够用于内存分配器创建，则内存分配器创建失败；如果指示内存由内存分配器自行申请，而内存分配器在向系统申请内存的时候失败，则内存分配器创建失败。

内存分配器在分配内存时，根据需要分配内存的大小选择合适的内存池，将内存池内空闲内存片管理节点链表的首节点对应的内存片分配给调用者使用，将对应的管理节点占用状态置为已占用，同时将链表首节点从链表中移除，更新内存分配器内保存的链表首节点。如果小的内存池已经没有空闲的内存片，则选择更大级别的内存池，如果都没有空闲，则分配

失败。

内存分配器在释放内存时，跟据传入的内存片指针判断属于那个级别的内存池。然后根据内存指针计算出对应的内存片管理节点，将对应节点的占用状态置为空闲，并将该节点加入到空闲链表首，同时更新内存分配器内保存的空闲链表首索引；如果内存指针不在内存片的边界点上，则为非法指针，内存释放失败。

在销毁内存分配器时，如果内存分配器管理的内存由内存分配器自行申请，则销毁时需要将申请的内存释放；否则话，销毁时不进行释放，由外部负责管理内存的释放

## 5.4 对 ATCA 支持的考虑

### 5.4.1 ATCA 上实现应用的几种方案

#### 5.4.1.1 多进程多实例

基本框架如图：

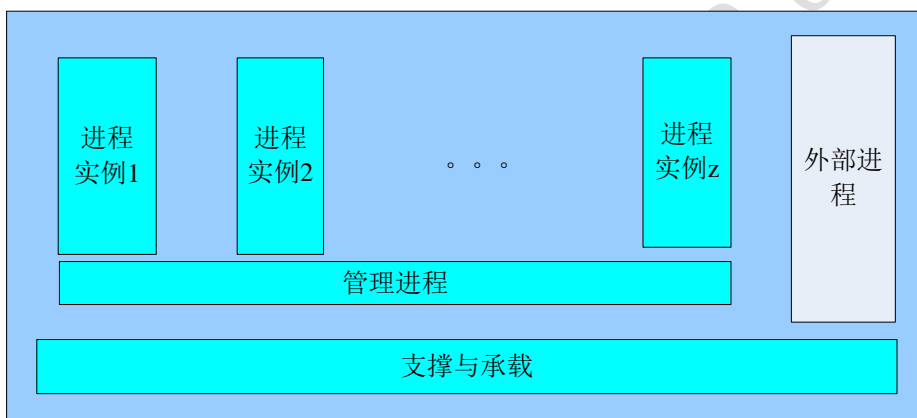


图 5.4.1 多实例方案基本框架

在多进程多实例中，对于同一个应用进程而言，可以同时存在多个实例，各个进程实例间彼此独立，都可以既可以同其他类型进程通信，接收进程间消息或者发送进程间消息，各进程实例间也可以接收和发送消息。在该种实现下，通常还会实现一个管理进程，该进程实现其所管理进程与外部进程间的消息接收和发送。

在不同的部署情况下，这一方案有不同的外观表现。

部署一：

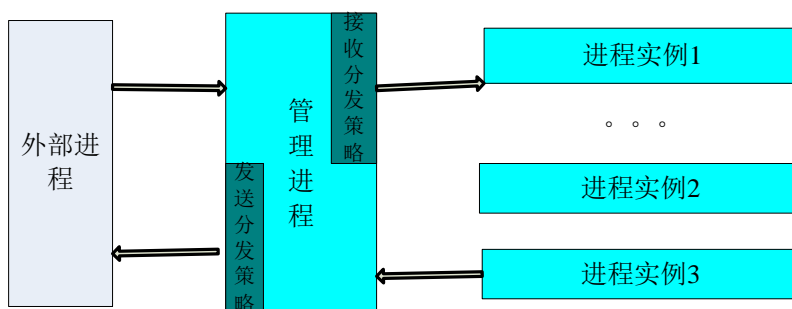


图 5.4.2 处理进程实例不对外暴露的多实例框架

该应用的多个实例进程中，仅有管理进程对外暴露，管理进程负责接收外部消息和内部消息。对于外部进程而言，应用进程仅有一个。

当管理进程收到外部消息时，其根据策略，选择一个处理进程，然后将消息发送给该处理进程，处理进程进行处理。

当应用进程需要向外部进程发送消息时，应用进程首先将该消息发送到管理进程，然后由管理进程根据策略发送给外部进程。

在该种部署情况下，由于所有的消息吞吐都是由管理进程负责，为了减轻管理进程负担，管理进程通常只负责消息的接收和发送，不进行应用处理。

部署二：

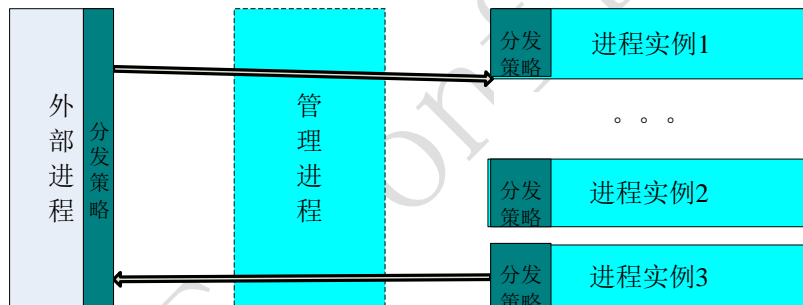


图 5.4.3 处理进程完全对外暴露的多实例框架

该应用的所有进程实例均对外部进程暴露。

外部进程在需要向应用发送消息时，都能根据本地策略确定处理消息的应用进程实例。

每个应用进程实例在需要向外部进程发送消息时，也都能够根据本地策略确定目的进程实例。在这种情况下。

应用进程与外部进程间的通讯不需要通过管理进程进行。正如图中所示，管理进程被虚化掉了。可以不进行部署。

部署三：

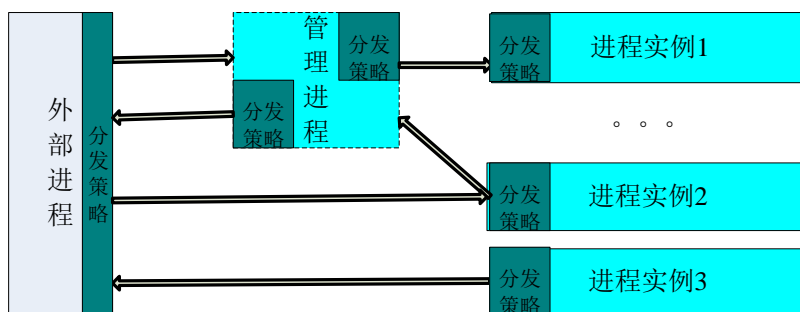


图 5.4.4 处理进程实例部分对外暴露的多实例框架

该应用的管理进程和处理进程均对外暴露。

外部进程在需要向应用发送消息时，如果能够根据本地策略确定处理消息的应用进程实例，则直接发往该实例；否则发往应用进程的管理进程，管理进程根据本地策略确定处理进程实例后，将该消息转发至处理进程实例。

该应用的处理进程在向外部进程发送消息时，如果能够根据本地策略确定外部进程实例，则直接发往外部进程实例；否则发往管理进程，管理进程根据本地策略，确定外部进程实例后，将消息转发至外部进程实例。

在该种部署情况下，如果需要通过管理进程进行消息转发的情况不多，可以在管理进程上部署处理进程，即管理进程实例和处理进程实例合一。

几种部署的优缺点：

对于部署一来说，其优点是外部应用仅能看到一个应用进程实例，看不到内部的多进程实例实现方式，接口比较清晰，处理接点；缺点是所有的消息处理，都需要通过管理进程进行转发，性能比较低。

对于部署二来说，其优点是内部外部均对外暴露，根据具体应用场景可以选择性的对外暴露管理进程和处理进程，从而取得接口易用性和性能间的平衡；缺点是内部实现相对复杂，需要根据不同的场景选择不同的策略。

对于部署三来说，其优点是消息的转发次数少，性能较高；其缺点是将内部实现对外暴露，外部进程需要选择处理实例，耦合比较紧，外部进程的处理相对负责。

对于 SIP 消息来说，通常对于初始请求各实例都可以进行处理，但是对于对话内消息的处理则要求和初始请求在同一个实例内处理，因此采用部署二来说比较合适。

#### 5.4.1.2 单进程多线程

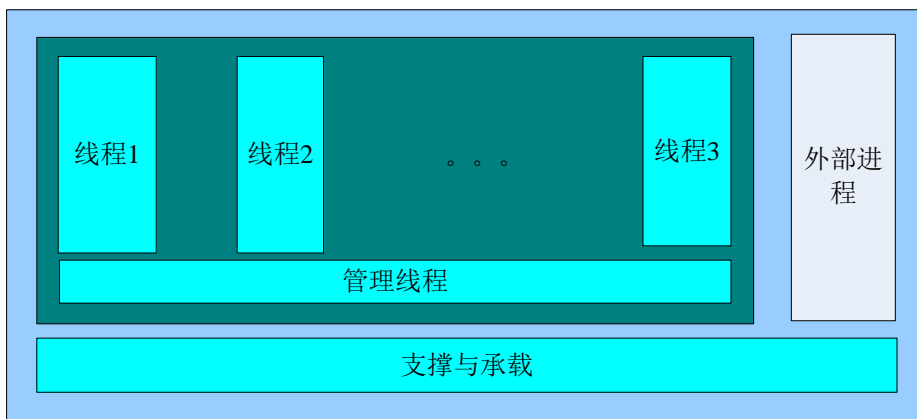


图 5.4.5 单进程多线程框架

在单进程多线程方案中，每个应用仅有一个进程实例，在每个进程内部采用多线程的方式来使用多核，以提高性能。

#### 5.4.1.3 多进程多线程

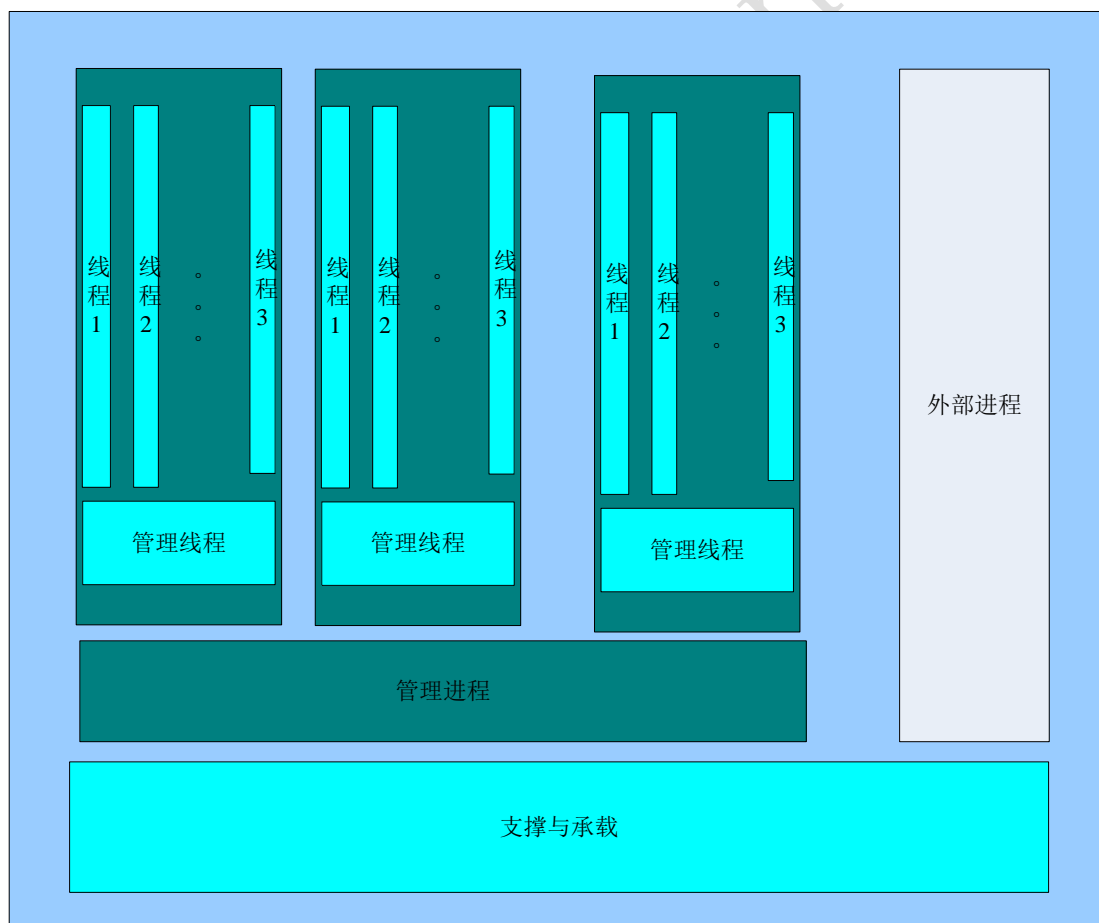


图 5.4.6 多进程多线程框架

该方案是方案一和方案二的融合，在该方案中，如果仅部署一个进程，则退化为单进程多线程方案，如果部署多个进程，但是每个进程仅由一个线程，则退化为多进程多实例方案。

#### 5.4.1.4 几种方案的优缺点

对于多线程方案，其优点是对于外部应用始终只暴露一个进程，与外部接口简单易用，并且进程间消息转发的次数较少，性能较高；其缺点是，在同一个代码、数据空间内，需解决全局变量、临界区等问题，对支撑层、应用层都要求有较大的改造工作量。

对于多进程方案来说，其优点是其单个进程实例的实现和原有实现相同，对于现有代码冲击较小，需要较小的改造工作量；其缺点是可能暴露多个进程，配置和使用相对复杂，并且由于转发消息的存在，其性能相对较低。

对于多进程多线程来说，其优点的是通过配置可以退化成多进程或者多线程方案，灵活性较大，可以适应不同项目的需求；其缺点是实现复杂，需要投入大量的改造工作量。

SIP 平台 V1.01.10 版本的多核设计采用多进程方案中的部署二框架。

### 5.4.2 SIP 平台的 ATCA 方案

#### 5.4.2.1 SIP 平台的多线程考虑

对于 SIP 平台的 V1.01 暂不实现多线程。

注：在 SSS2.0 中，SIP TP、SIP TR/TU、上层应用在同一个二次调度进程内。

#### 5.4.2.2 SIP 平台的多进程多实例考虑

##### 5.4.2.2.1 SIP 编解码的多实例考虑

为了可以将编解码应用于 ATCA，编解码内部不能有自己的资源如临时使用的静态变量等。编解码的多实例采用内存分配器机制，在创建编解码实例的时候创建内存分配器，编解码函数在需要使用内存的时候，向内存分配器申请内存资源，在使用完毕后将申请的内存资源释放给内存分配器。在销毁编解码实例的时候销毁内存分配器。

##### 5.4.2.2.2 SIP TP 的多进程实现考虑

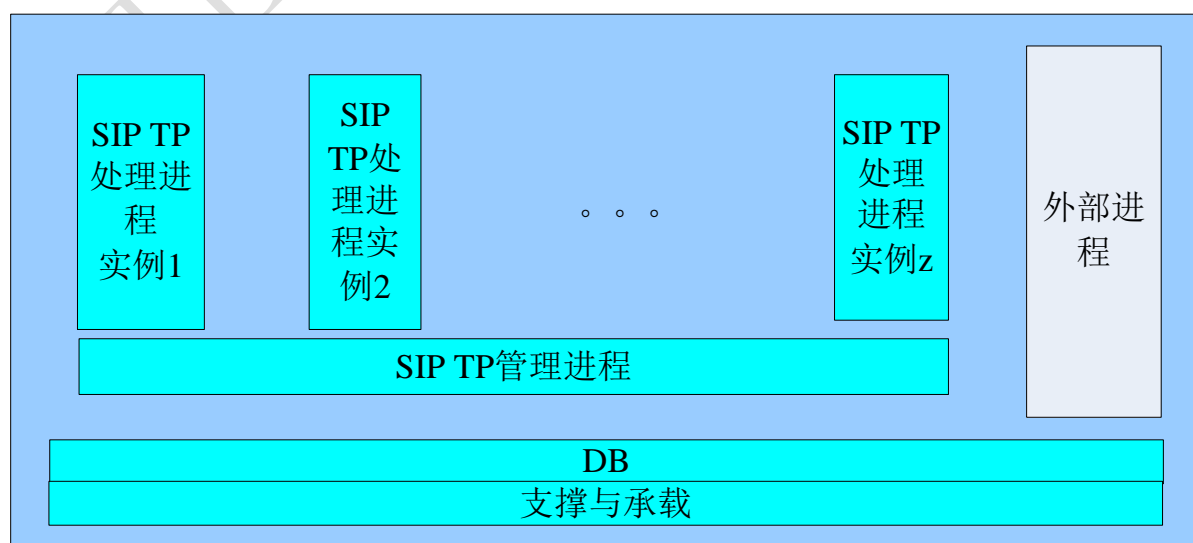




图 5.4.7 SIP TP 的多实例框架图

SIP 传输层包括 SOCKET 和 LINK 维护、上行分发和下行发送、THIG 隐藏、IPSec。另外信令压缩模块也在上下行报文收发中一起发挥作用。其中多进程对 THIG 隐藏没有什么影响。对信令压缩和 IPSec 的影响另外单独描述。

SOCKET 与 LINK 维护，SIP 传输层通过配置接口获取本板需要监听的端口、建立的链路。

不同的 SIP TP 处理进程实例创建的 SOCKET 端口应当不同，因此在配置的时候为各 SIPTP 处理进程实例配置的 IPPORT 不能相同。

对于没有固定实例的 SOCKET 端口和 LINK（或者配置对应实例为管理进程）则由管理进程负责，管理进程不负责消息处理，只进行消息接收和发送。管理进程在从承载收到 SIP 消息包时，调用 DB 接口确定 SIP TP 处理进程实例（在接口内进行负荷均衡处理）。

SIP TP 处理进程在需要向承载发送消息包时，如果发送所用的 SOCKET 端口由本实例创建，则使用该 SOCKET 端口发送消息。如果不是，则将消息发往 SIP TP 管理进程，由 SIP TP 管理进程发送。（对于 UDP 承载情况下，如果承载支持不需 SOCKET 可以直接发送，则 SIP TP 处理进程可以不将消息转交 SIP TP 管理进程，而直接发送）。

对于每个单板只配置了一个 IPPORT 的情况下，为了符合均衡，该端口只能配置在 SIP TP 管理进程。此时 SIP TP 管理进程负责与承载 SIP 消息接收和发送，在这种情况下 SIP 管理进程可能成为性能瓶颈。

管理进程在接收到承载或者 SIP TR/TU 传递过来的消息时调用数据库接口决定由哪个实例来处理该消息，后发送给该实例。

上行分发的几种场景：

场景 1，承载或者其他分发模块，能够直接确定 SIP TP 处理进程实例，将 SIP 消息直接发送到 SIP TP 处理进程，SIP TP 在处理完毕后调用配置接口确定上行的目的进程为 SIP TR/TU 进程，将该消息发送至 SIP TR/TU 进程；

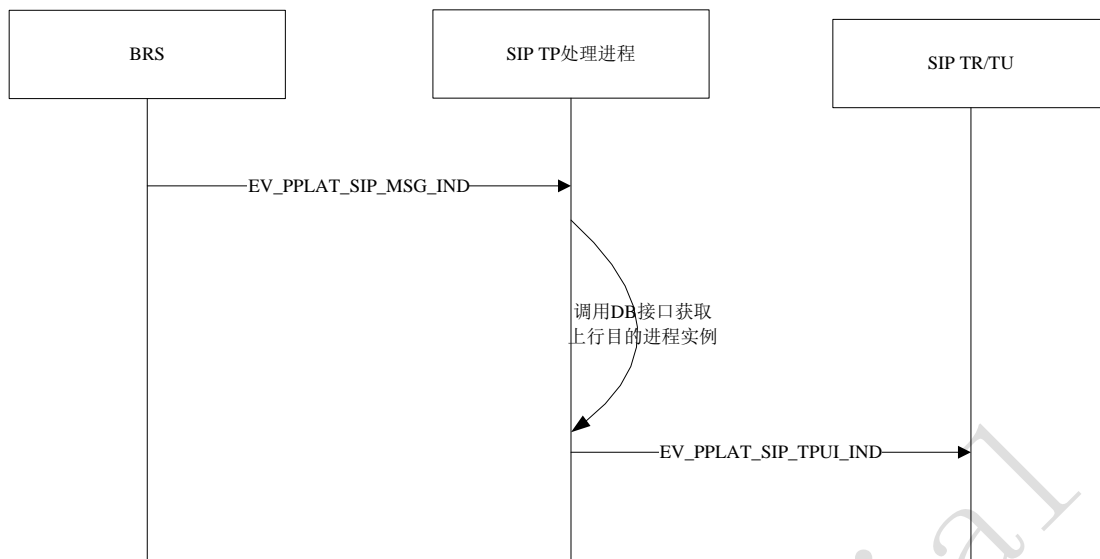


图 5.4.7 SIP TP 进程的上行分发场景(一)

场景 2，承载或者其他分发模块，不能够直接确定 SIP TP 处理进程实例，将 SIP 消息直接发送到 SIP TP 管理进程，SIP TP 管理进程调用 DB 接口确定处理进程实例，并将 SIP 消息发送至 SIP TP 处理进程。SIP TP 在处理完毕后调用 DB 接口确定上行目的进程为 SIP TR/TU 进程，将该消息发送至 SIP TR/TU 进程；

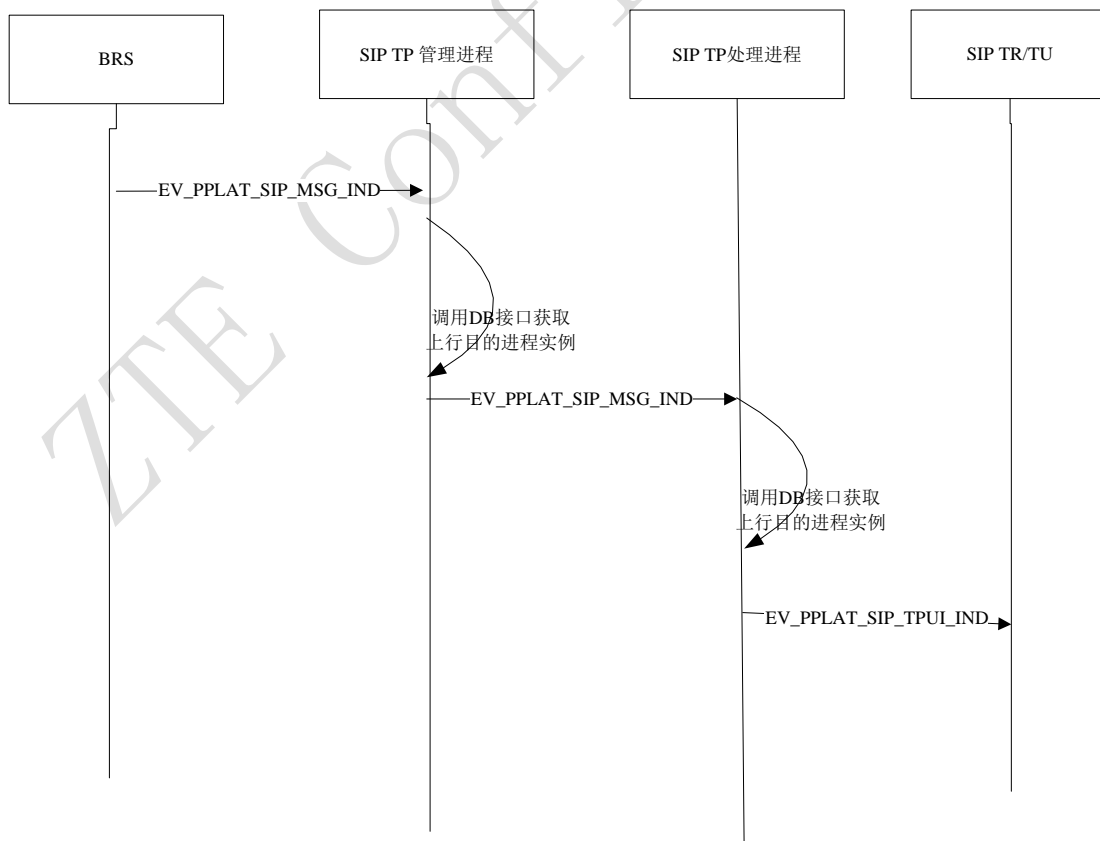


图 5.4.8 SIP TP 进程的上行分发场景(二)

下行分发场景：

场景 1, SIP TR/TU 进程能够确定 SIP TP 处理进程实例, SIP TR/TU 将消息发送至 SIP TP 处理进程; SIP TP 处理进程在处理完毕后将 SIP 消息通过承载发送出去。此时发送 SIP 消息所用 SOCKET 端口由 SIP TP 处理进程创建

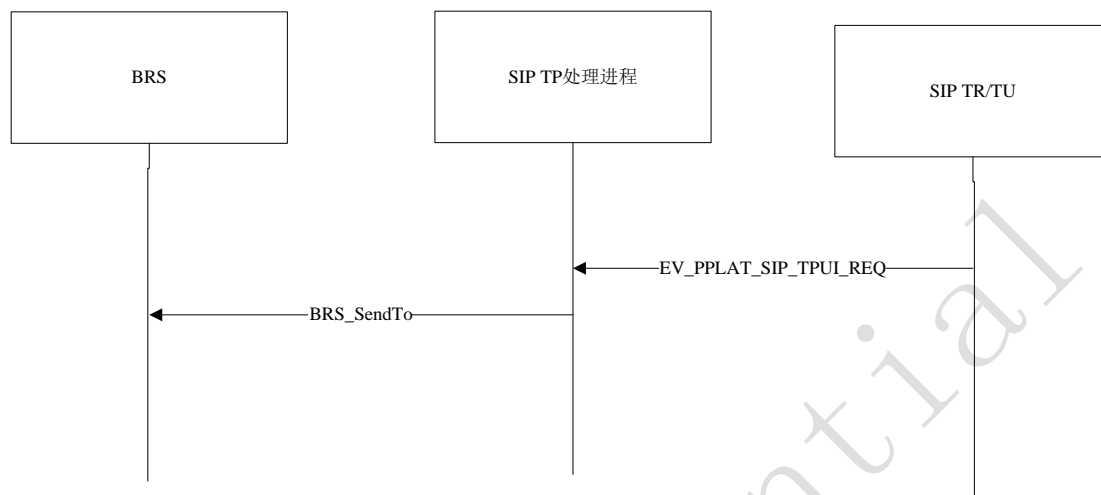


图 5.4.29 SIP TP 进程的下行分发场景(一)

场景 2, SIP TR/TU 进程不能能够确定 SIP TP 处理进程, SIP TR/TU 将消息发送至 SIP TP 管理进程; SIP TP 管理进程调用 DB 接口确定 SIP TP 处理进程实例, 并将消息发送至 SIP TP 处理进程; SIP TP 处理进程在处理完毕后将 SIP 消息通过承载发送出去。此时发送 SIP 消息所用 SOCKET 端口由 SIP TP 处理进程创建

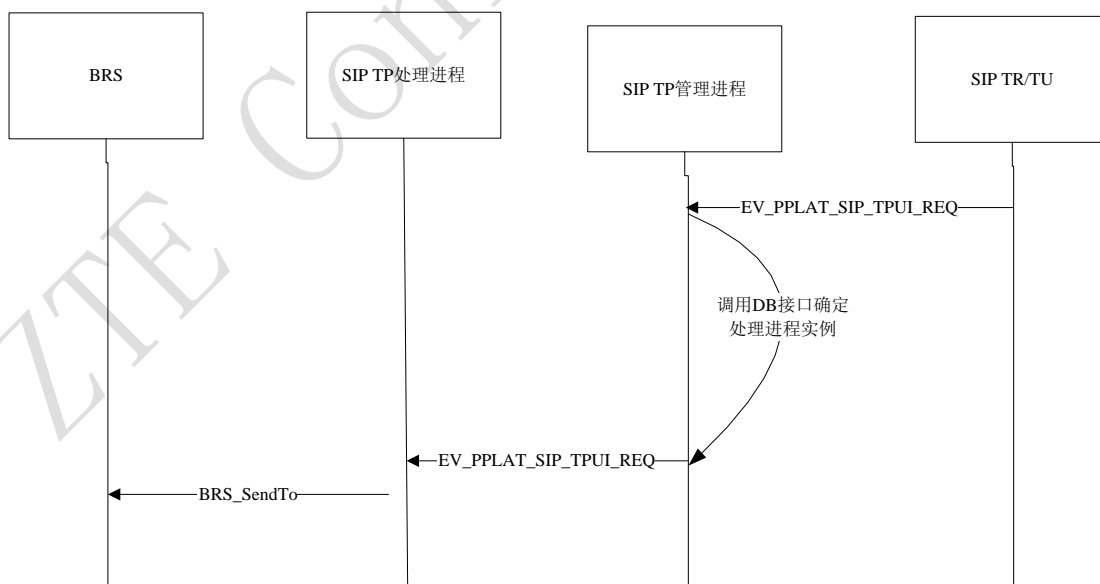


图 5.4.210 SIP TP 进程的下行分发场景(二)

场景 3, SIP TR/TU 进程能够确定 SIP TP 处理进程实例, SIP TR/TU 将消息发送至 SIP TP 处理进程; SIP TP 处理进程在处理完毕后将消息转发至 SIP TP 管理进程。SIP TP 管理进程将 SIP 消息通过承载发送出去。此时发送 SIP 消息所用 SOCKET 端口由 SIP TP 管理

进程创建。

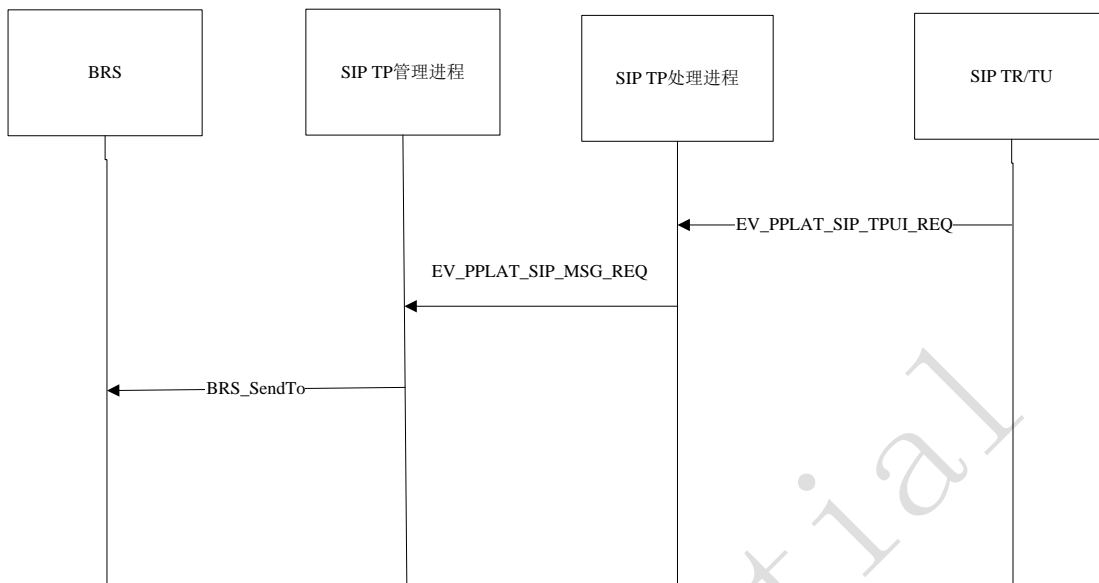


图 5.4.211 SIP TP 进程的下行分发场景(三)

场景 4, SIP TR/TU 进程不能能够确定 SIP TP 处理进程, SIP TR/TU 将消息发送至 SIP TP 管理进程; SIP TP 管理进程调用 DB 接口确定 SIP TP 处理进程实例, 并将消息发送至 SIP TP 处理进程; SIP TP 处理进程在处理完毕后将消息转发至 SIP TP 管理进程。SIP TP 管理进程将 SIP 消息通过承载发送出去。此时发送 SIP 消息所用 SOCKET 端口由 SIP TP 管理进程创建

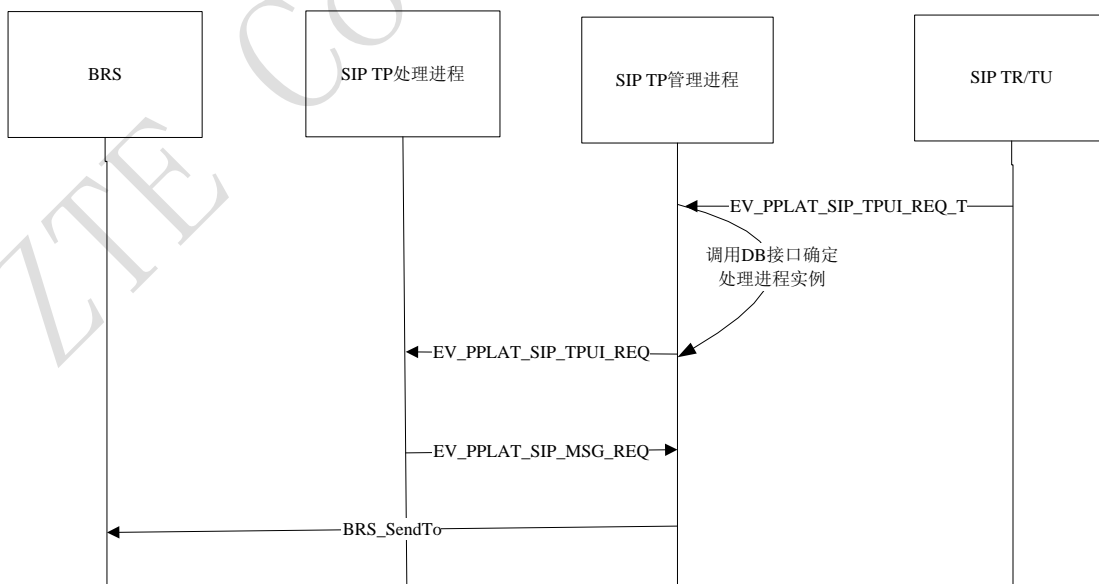


图 5.4.213 SIP TP 进程的下行分发场景(四)

#### 5.4.2.2.3 SIP TR/TU 的多进程实现方案

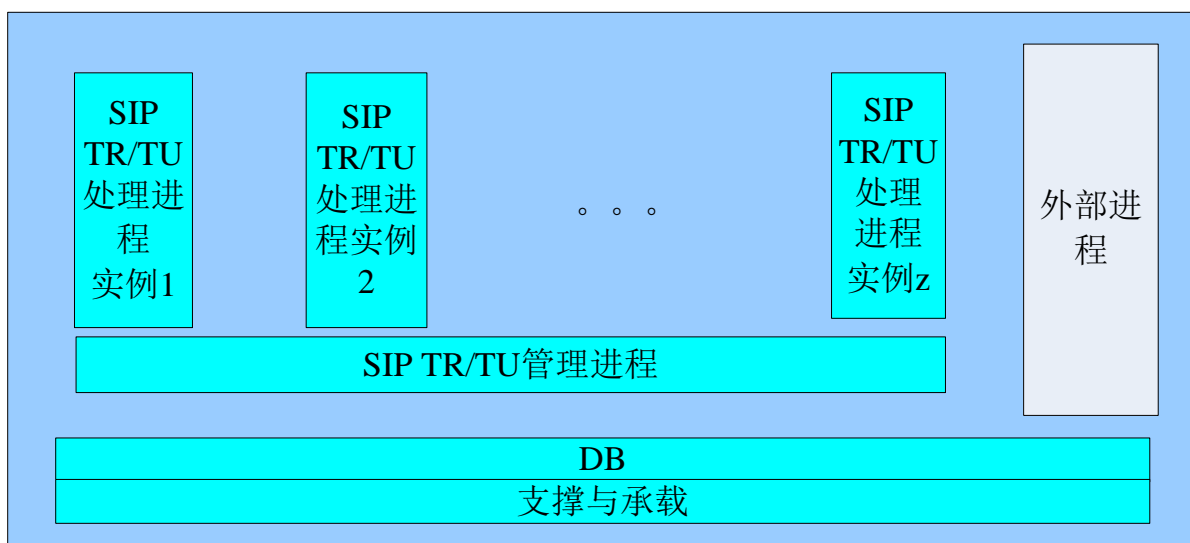


图 5.4.214 SIP TR/TU 进程的多实例框架

SIP TR/TU 的主要功能可以参见《NGN PPLAT V1.00.10\_系统方案》中 6.2.1 和 6.2.2。

多进程方案对其基本功能基本上没有什么影响，在这里就不单独描述了。

对于 SIP TR/TU 管理进程来说，其主要负责上下行消息的分发，并不负责具体消息的处理。

对于 SIP TR/TU 处理进程主要负责 SIP TR/TU 具体功能的执行。

对于 SIP TU 进程与上层应用间分发，由于 SIP TU 数据区和上层应用数据区实例中互相保有对方的进程标识，因此对于后续请求，直接使用保存的进程标识进行分发。这里考虑的主要是上行和下行初始请求的分发。

上行分发的几种场景：

场景 1，SIP TP 进程能够直接确定目的能够直接确定 SIP TR/TU 处理进程，并将消息发送至 SIP TR/TU 处理进程，在这种情况下，要求同一个对话内的消息应发送至同一个 TU 处理进程实例；SIP TR/TU 处理进程处理完毕后调用接口确定上行目的进程为上层应用进程 APP，并并消息发送给 APP；

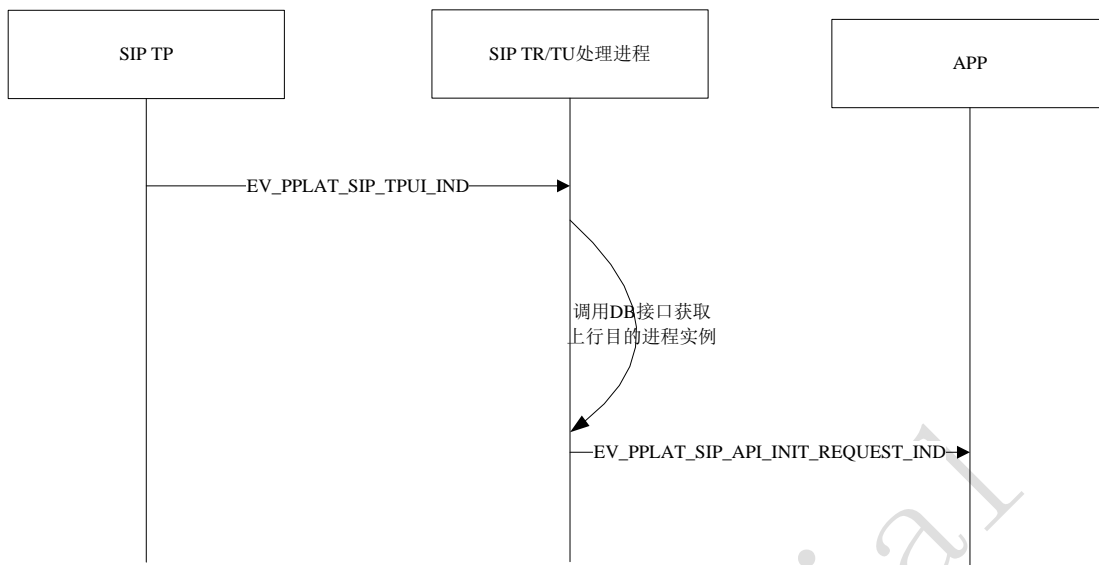


图 5.4.2.15 SIP TR/TU 进程的上行分发场景(一)

场景 2，SIP TP 进程不能够直接确定 SIP TR/TU 处理进程实例，将 SIP 消息直接发送到 SIP TR/TU 管理进程，SIP TR/TU 管理进程调用 DB 接口确定处理进程实例，并将 SIP 消息发送至 SIP TR/TU 处理进程，在这种情况下要求同一个对话内的消息发送至同一个 TU 处理进程实例。SIP TR/TU 在处理完毕后调用 DB 接口确定上行目的进程为应用进程，将该消息发送至应用进程；

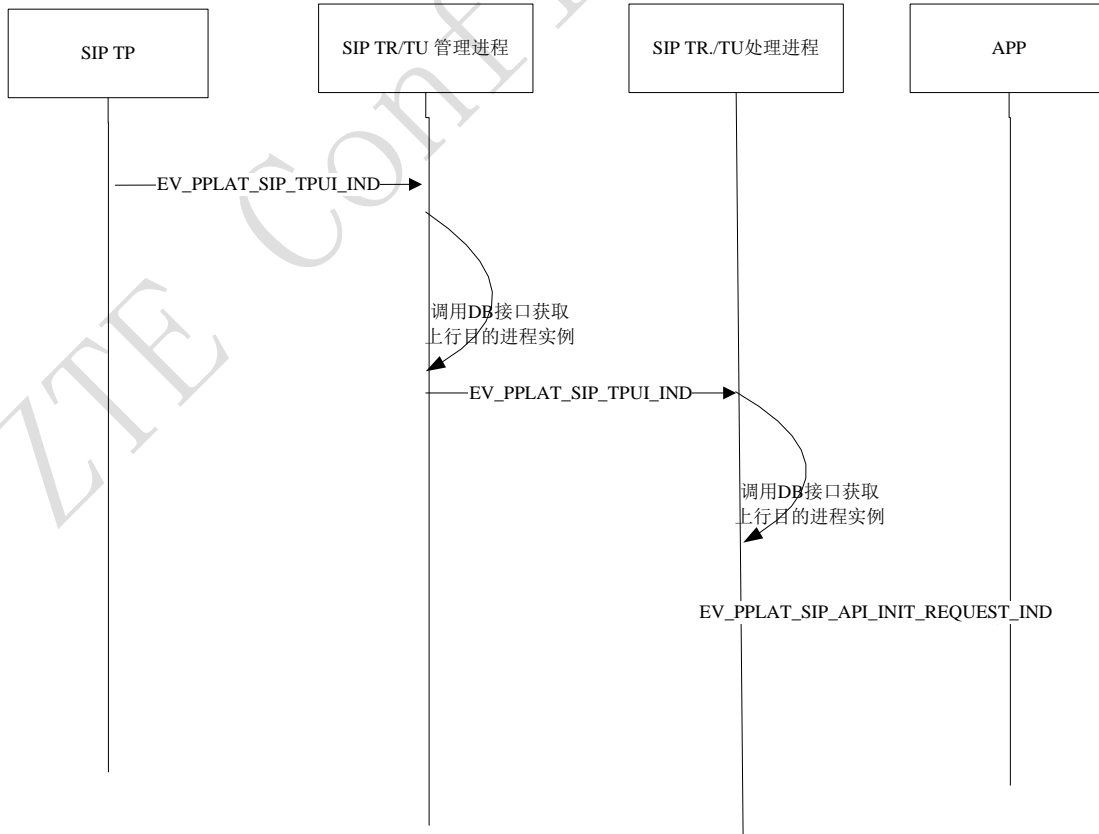


图 5.4.2.16 SIP TR/TU 进程的上行分发场景(二)

下行分发场景：

场景 1，业务进程能够确定 SIP TU 处理进程实例，并将消息发送至 SIP TU 处理进程；SIP TU 处理进程在处理完毕后调用下行分发接口确定目的接口位 SIP TP。

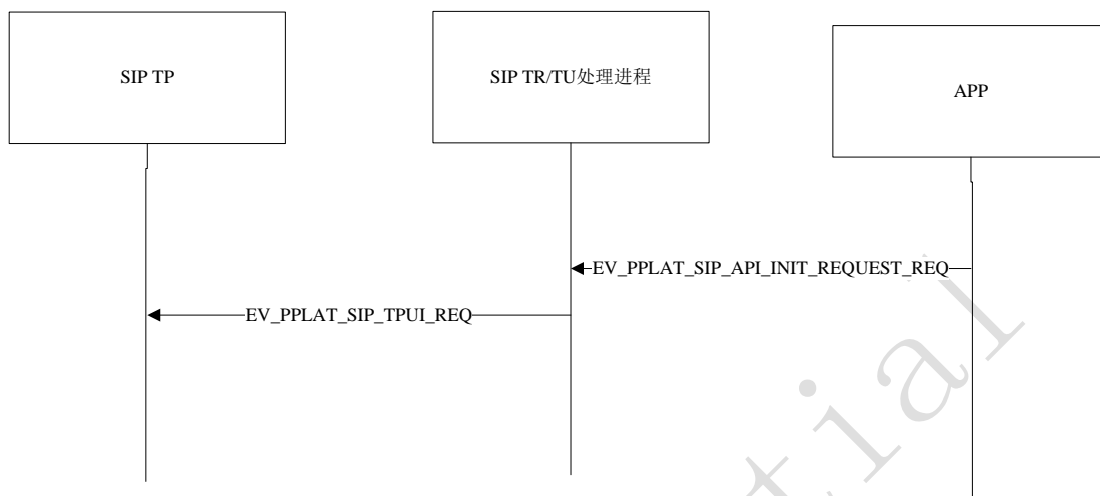


图 5.4.217 SIP TR/TU 进程的下行分发场景(一)

场景 2，业务进程不能能够确定 SIP TU 处理进程，将消息发送至 SIP TU 管理进程；SIP TU 管理进程调用 DB 接口确定 SIP TU 处理进程实例，并将消息发送至 SIP TU 处理进程；SIP TU 处理进程在处理完毕后调用下行分发接口，确定目的进程为 SIP TP。

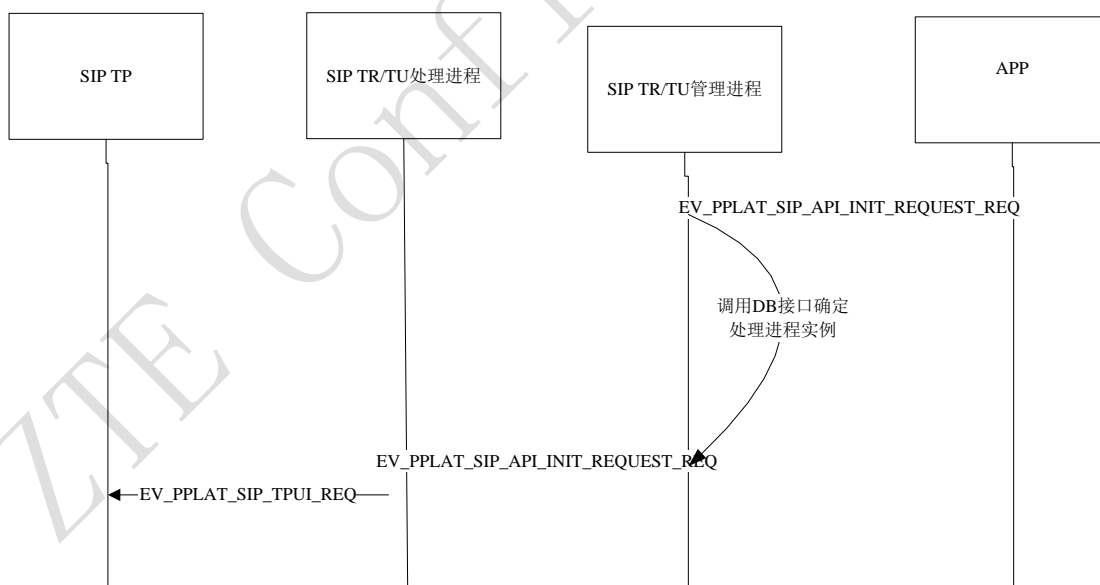


图 5.4.218 SIP TR/TU 进程的下行分发场景(二)

#### 5.4.2.2.4 完整的分发图景及分发点

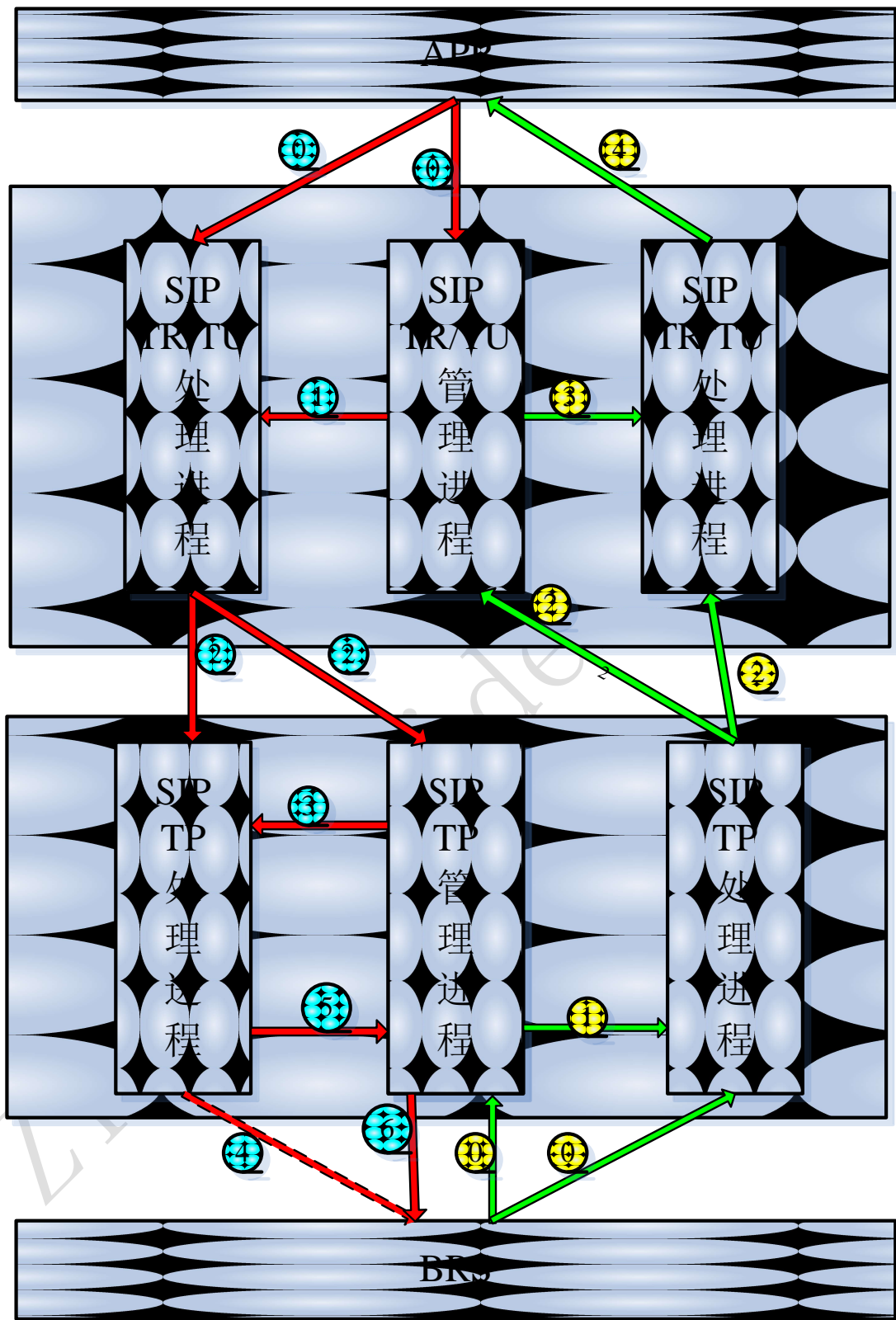


图 5.4.219 SIP 平台的完整分发图景

整个 SIP 平台的完整分发图景如上图，在上图中绿色箭头为上行消息的分发流程，红色箭头为下行消息的分发流程，其中：

SIP TP 处理进程有上行分发点一个，如图中的绿 2，该方分发的确定的目的进程为 SIP



TR/TU 处理进程或者 SIP TR/TU 管理进程。

SIP TP 管理进程有上行分发点一个,绿 1 为 SIP TP 管理进程收到底层承载模块发送过来的 SIP 消息后选择 SIP TP 处理进程。

SIP TR/TU 处理进程有上行分发点一个,绿 4 为 SIP TR/TU 处理进程在处理完毕后选择上行应用进程。该选择通常是对初始 SIP 请求的,对于后续请求 SIP TU 数据区内保存有对端应用进程标识,直接发送。

SIP TR/TU 管理进程有上行分发点一个,绿 4 为 SIP TR/TU 管理进程收到 SIP TP 进程上报消息后选择 SIP TR/TU 处理进程。

SIP TR/TU 处理进程有下行分发点一个,红 2 为 SIP TR/TU 在将下行消息处理完毕后选择下行 SIP TP 进程。

SIP TR/TU 管理进程有下行分发点一个,红 1 为 SIP TR/TU 管理进程收到业务指示后选择 SIP TR/TU 处理进程。该分发也只是在上层应用发送初始请求时发生,对于后续请求业务使用其保存的 SIP TU 处理进程标识直接发往 SIP TU 处理进程。

SIP TP 处理进程有下行分发点两个,红 4 为 SIP TP 处理完毕后通过承载发送 SIP 消息。此时发送消息使用的 SOCKET 在本实例创建。红 5 为 SIP TP 处理进程将 SIP 消息转发至 SIP TP 管理进程,交由 SIPTP 管理进程发送,此时发送所用的 SOCKET 由 SIP TP 管理进程创建和维护。

SIP TP 管理进程有下行分发点两个,红 3 为在收到 SIP TR/TU 下行消息后选择 SIP TP 处理进程。红 6 为在收到 SIP TP 转发的 SIP 消息后,通过承载将该消息发送出去。

注:在上述分发过程中,如果 SIP TP 与承载间的消息收发始终通过 SIP TP 管理进程。即 SIP TP 管理进程负责创建和维护 SOCKET,以及从 SOCKET 收包发包。而处理进程则负责 SIP 消息的解码编码以及其他功能,这种情况下 SIP TP 管理进程可能成为瓶颈,需要综合考虑 SOCKET 的收发包与 SIP TP 其他功能对 CPU 资源的占用比例。

#### 5.4.2.2.5 SSS2.0 中的 SIP 平台部分的分发图景

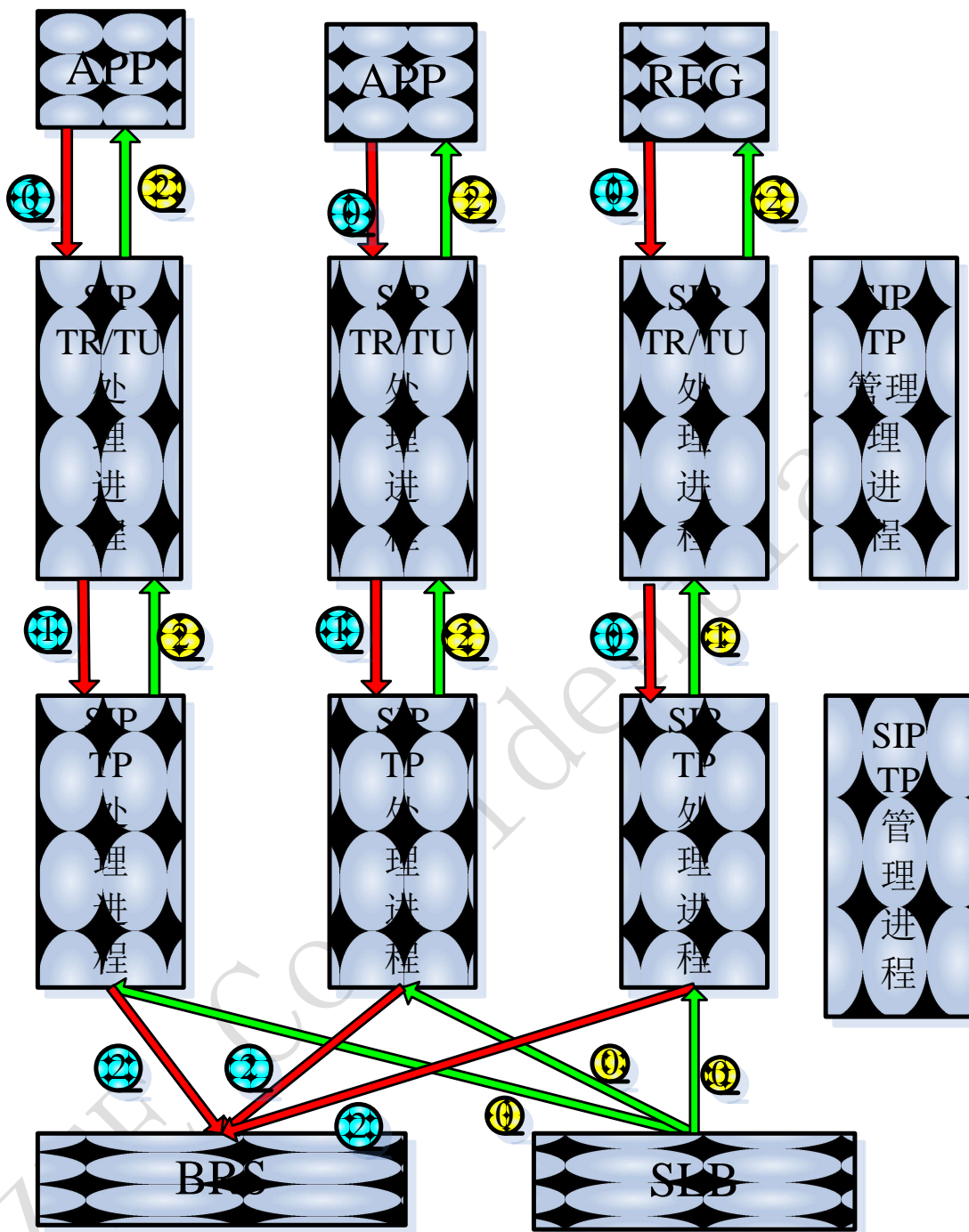


图 5.4.220 SSS2.0 中的 SIP 消息分发

在 SSS2.0 中，在每个 SIP 处理板上，SIP TP、SIP TR/TU 以及上层应用的实例部署是一一对应的，在每个单板上仅有一个注册的应用进程实例。

IPI 板子会将初始请求发送到 SLB，SLB 会通过 DB 选择处理进程实例，并将初始消息发送到对应的 SIP TP 处理进程实例。其中注册消息会发送到和注册进程对应的 SIP TP 处理进程实例。

上行的时候：

SIP TP 处理进程收到从 SLB 发送过来的初始请求消息，处理完毕后，发送到和本 SIP TP 处理进程对应的 SIP TR/TU 处理进程。

SIP TR/TU 处理进程收到从 SIP TP 发送过来的初始请求消息，处理完毕后，发送到和本 SIP TR/TU 处理进程对应的应用处理进程。

下行的时候：

上层应用向和本应用对应的 SIP TR/TU 处理进程发送初始请求。

SIP TR/TU 收到从上层用发送过来的初始请求消息，处理完毕后，发送到河本 SIP TR/TU 处理进程对应的 SIP TP 处理进程。

SIP TP 收到处理进程收到 SIP TR/TU 处理进程发送的消息，处理完毕后，通过承载将消息发送出去。

对于对话内消息，上行的时候由于各 SIP TP 处理进程监听的 SOCKET 端口不一样，IPI 可以直接将消息发送到对应的 SIP TP 处理进程。其后续的上行路径和初始请求的一致；下行的时候路径和初始请求消息一致，在此不再描述。

#### 5.4.2.2.6 SIP TP 管理进程的实现原理

SIP TP 管理进程主要负责上下行消息的分发，其分发的策略由具体项目决定。当在某些配置情况下，SIP TP 管理进程需要从承载收包或者向承载发包，这种情况下，SIP TP 管理进程需要负责链路和 SOCKET 的维护，这部分功能和 SIP TP 对应的功能类似。另外如果此时还要求支持 IPSec 功能，那么和对应 SOCKET 相关的 IPSec 功能也要求 SIP TP 管理进程实现。

#### 5.4.2.2.7 SIP TU 管理进程的实现原理

SIP TU 管理进程主要负责上下行消息的分发，其分发的策略由具体项目决定。在某些实现情况下，上下行消息还可以经过 SIP TU 管理进程，比如 SSS2.0 的分发实现。

除此之外，SIP TU 管理进程负责与操作维护相关一些功能与消息与处理，在网管不愿意看到多个 SIP 处理进程实例的情况下，网管可以将请求消息发给 SIP TU 管理进程，由 SIP TU 管理进程负责消息的处理和转发。

#### 5.4.2.2.8 多 SIP TP 管理进程的考虑

暂缺

#### 5.4.2.2.9 多实例情况下的主备倒换考虑

注：修订记录编号 0008

V4 平台下的主备倒换机制，可用下面流程图表示

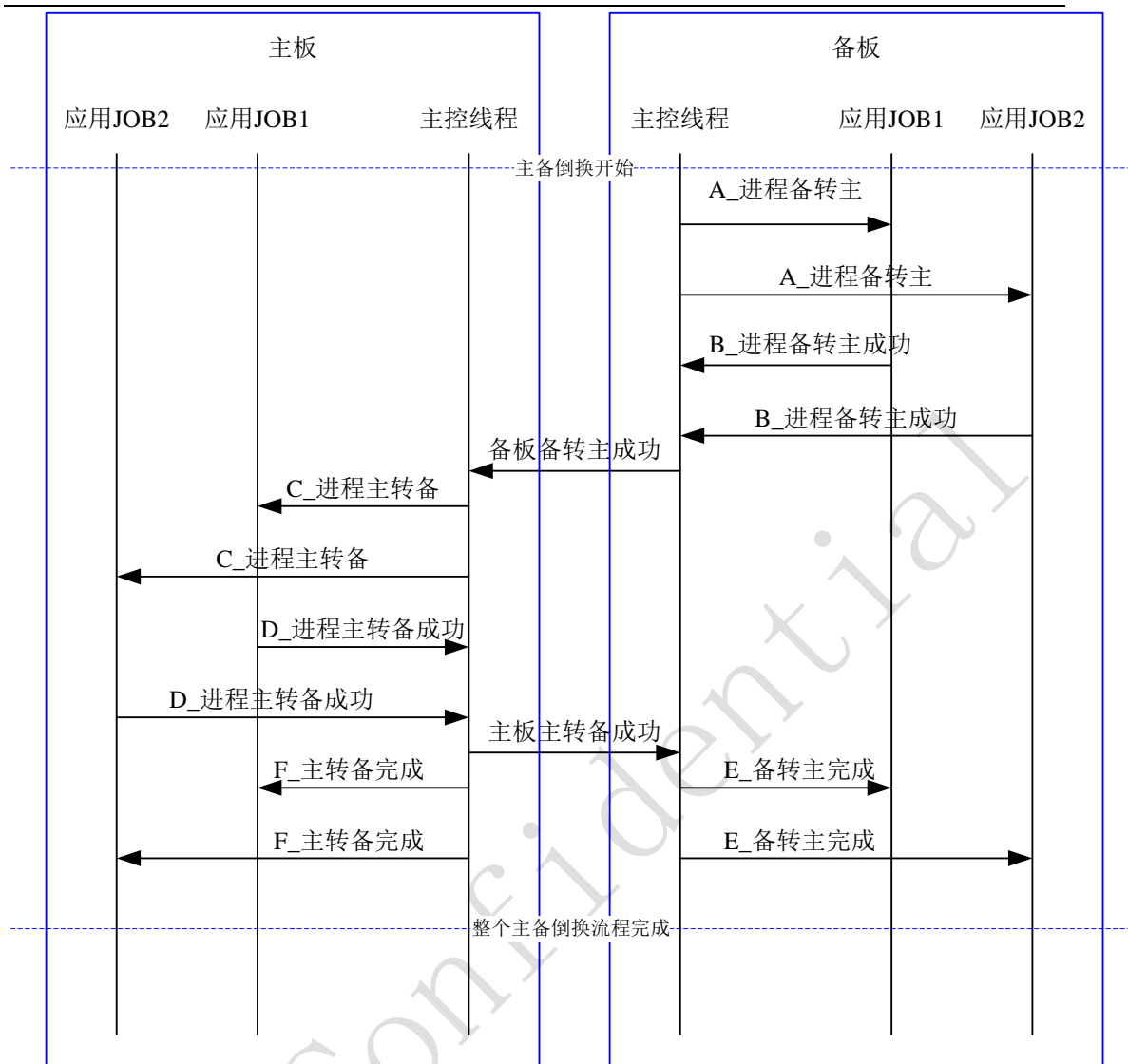


图 5.4.221 V4 平台下的主备倒换流程

V4 平台下主备倒换流程说明如下：

- 消息A: EV\_SLAVE\_TO\_MASTER; 系统主备倒换的开始消息，备板JOB收到此消息后，根据各自情况进行一些初始处理，小量的数据同步，资源申请等工作；此时禁止执行依赖于其他JOB主备状态的流程处理，因为此时其它JOB的主备处于不确定状态，可能无法正常应答。
- 消息B: EV\_CHANGEOVER\_SUCCESS/EV\_CHANGEOVER\_FAIL; 若登记本JOB需要对EV\_SLAVE\_TO\_MASTER消息做应答时，需要在处理完EV\_SLAVE\_TO\_MASTER消息后给主控线程响应。若在SCS规定的时间内未回应，则备板将重新启动，主备倒换失败。
- 消息C: EV\_MASTER\_TO\_SLAVE; 主板收到备板处理完成备转主消息后，主板的主控线程向所有的主板JOB发送EV\_MASTER\_TO\_SLAVE，开始主转备操作。此事件各JOB主要完成数据同步和释放资源等操作，此过程完成后，可以将JOB切换为备用态，但考虑到各JOB的一致性，不建议在此事件中触发备用状态的JOB交互流程。
- 消息D: 用途同消息B。

- e. 消息E: EV\_SM\_FINISH; 备转主完成, 各JOB收到此消息后, 本板已经切换为主用单板, 并且开始接受业务消息, 此时可以完成以下工作: 申请资源, 激活检测以及其他JOB间的交互, **最重要的一条, 必须也只允许在此事件中, 将本JOB状态切换到主用状态。**
- f. 消息F: EV\_MS\_FINISH; 主转备完成, 与消息E没有先后关系。各JOB收到此消息后, 标志着本板已经切换为备用状态, 此事件中可以完成备用状态的切换, 备用板JOB间流程的触发, 备用板定时器的重设等工作。

根据以上的分析, 可以看出, V4 平台下的主备倒换机制, 与 V3 平台相差不大, 最大的区别是: 增加了 EV\_SM\_FINISH/EV\_MS\_FINISH 两个事件通知, 以保证所有 JOB 倒换完成后切换状态。

对于正常运行时的同步过程, 与 V1.00.10 中方案一致, 详见《NGN PPLAT V1.00.10\_系统方案》; 对于硬倒换过程与 V1.00.10 中方案一致, 只要运行过程中的同步工作进行的没有问题, 硬倒换就不会存在问题。对于软倒换, 过程与上图中描述一致。

在 V4 平台下, SIP 平台的主备倒换需要做如下考虑:

- (1) 各个 JOB 收到 EV\_SLAVE\_TO\_MASTER/ EV\_MASTER\_TO\_SLAVE 后进行主备倒换过程, 不需要回送确认响应。即上图中 B 的过程不需要。
- (2) 各个 JOB 收到 EV\_SM\_FINISH/ EV\_MS\_FINISH 后完成主备倒换切换状态。

在 NGN PPLAT V1.00.10 版本中在收到 EV\_SLAVE\_TO\_MASTER/ EV\_MASTER\_TO\_SLAVE 进行主备倒换相关操作后即进行状态的切换。SIP V1.01.10 版本采用 V4 平台, 需要增加对 EV\_SM\_FINISH/ EV\_MS\_FINISH 事件的处理, 在收到这两个事件后, 才进行状态的切换、定时器的设置杀除、socket 的初始化关闭、链路的初始化关闭等。

SIP TU 管理进程不存在数据区或者全局性数据需要同步, 因此不需要特别考虑。

对于 SIP TP 管理进程, 如果该进程创建了链路、SOCKET 数据区, 或者 IPSecSA 数据区, 则对于这一部分的同步和 SIP TP 处理进程一致。

#### 5.4.2.2.10TU 到上层应用的消息分发的兼容性考虑

在 V1.0 中, SIP 协议栈为上层应用服务, 上层应用可能需要对不同的 SIP 消息对应不同的业务处理进程, 为了确保 SIP 协议栈的通用性, 采用上层应用上电时向协议栈主动注册其特性、SIP 协议栈根据注册信息进行对话外 SIP 消息的上行分发(TU → APP)。上层应用的注册信息包括其对外呈现所处理的 SIP 方法和分发的进程 ID。(SIP 平台还提供重载分发决策函数供项目重载使用。此时分发结果完全由重载函数的返回值所决定) SIP 协议栈根据上层应用注册信息来构造业务分发表(需要持久化保持, 防止 SIP 协议栈模块重启的情况), SIP 协议栈 TU 在收到 SIP 消息处理后根据此表分发到相应的上层应用进程。SIP 协议栈优先采用分发决策函数返回的业务进程 ID 作为分发目标。当此目标不存在, 使用分发表中业务注册的进程 ID 作为分发目标。对于对话内消息, SIP 协议栈实例数据区已经记录了上层

应用的分发信息，上行 SIP 消息直接发送到已知进程。

只有当收到上行对话外 SIP 消息时，SIP 协议栈才执行上层应用分发过程，根据应用注册的信息将 SIP 消息分发到正确的应用处理进程。

在 ATCA 平台下，SIP TU 处理进程可能存在多个实例，上层应用进程也能存在多个实例。这个时候对于根据重载分发函数分发没有影响，但对于注册机制，则需作如下更改，由于对于每种 SIP 方法只能有一个上层应用进程，所以如果存在多个上层应用实例需要向同一个 TU 处理进程注册的情况时则由上层应用进程的管理进程向 TU 处理进程注册，TU 会将初始请求消息上报给上层应用的管理进程，由上层应用的管理进程负责分发；由于存在多个 TU 处理进程，上层应用进程需要向所有的 TU 处理进程注册。

SIP 平台不支持上层应用通过异步消息指示上层应用处理进程标识方案，即对如下分发流程 SIP 平台不支持：

SIP TU 收到初始请求消息；

SIP TU 向上层应用管理进程发送请求

上层应用选择处理进程后发送异步消息通知 SIP TU

SIP TU 根据上层应用指示将初始请求发送至指定进程

#### 5.4.2.2.11 SIP 平台采用分布式部署情况下的分发考虑

注：修订记录编号 0001

在 ATCA 平台下，考虑 SIP 平台采用分布式的部署，即 SIP TP 与 SIP TR/TU 分开部署，分布在不同的处理模块上，如下图所示情况，SIP TP 分布在信令模块上，SIP TR/TU 与上层业务一起分布在业务模块上。



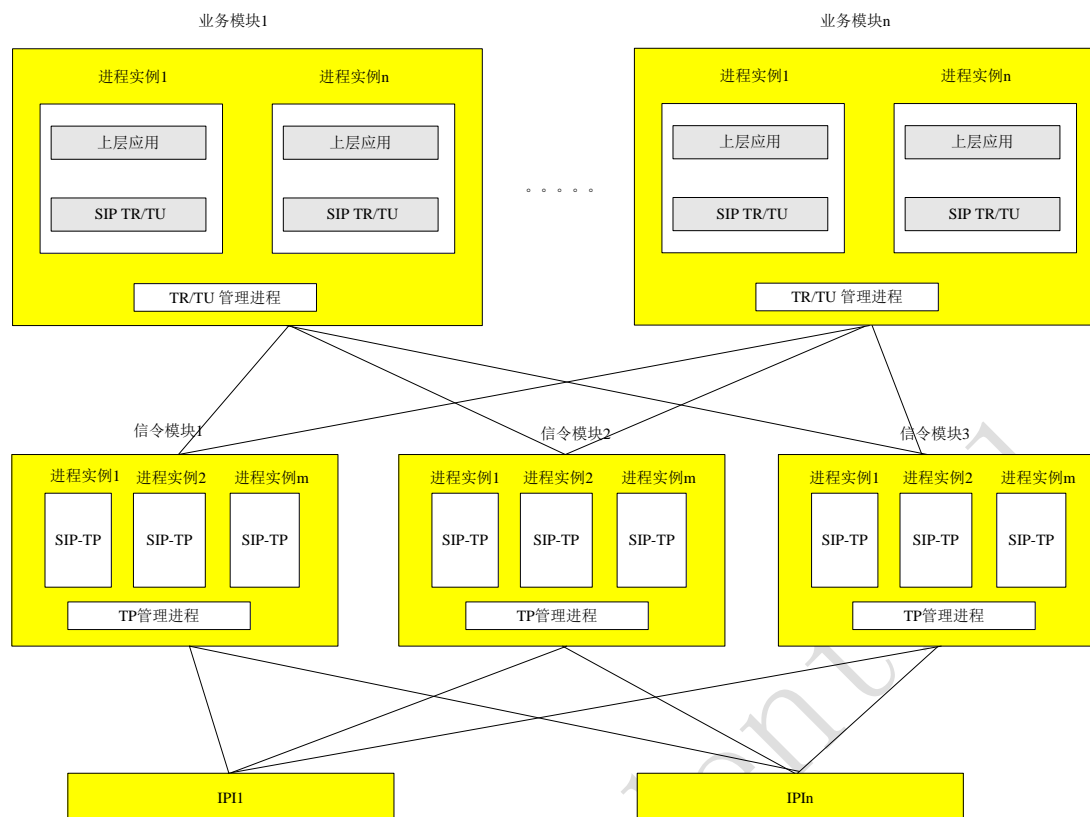


图 5.4.221 SIP 平台采用的某种分布式部署

对于图 5.4.21 中所描述的分布式部署，考虑 SIP 平台的分发方案。为了发挥多核的作用，希望 IPI 发到信令模块上的消息，能够均匀分发到不同的 SIP TP 实例，由于 IPI 到信令模块的分发是根据目的地址进行相应分发的，这就需要对外暴露所有 TP 实例的地址。在实际应用中是不希望对外暴露太多的端口的。为此考虑对外只暴露 TP 管理进程 TPM 的地址，消息经过 TPM，然后由 TPM 进行均匀分发到 TP 实例，这种实现方式，容易造成 TPM 成为性能瓶颈。

综合考虑上面所遇到的问题，可以采用如下实现方案：对外暴露 TP 管理进程的地址端口，作为监听端口；并利用 SIP 消息的特性，通过在 SIP 消息中携带地址，端口号信息，使后续消息（包括响应消息和后续对话内的请求消息）直接由 IPI 发到 SIP 消息中所确定的地址端口号所在的 TP 实例上。这样既解决了对外暴露多个地址、端口号的问题，也解决了 TP 管理进程造成性能瓶颈的问题。这种方案需要 TPM 与 TP 均绑定 socket，并且要保证不同的 TP 实例绑定的端口号不同。TPM 不做消息的处理，只是对消息进行相应的转发，TP 层的相应功能均在 TP 实例中实现。考虑到某一 TP 实例可能会出现故障 down 掉，可以在两个或多个 TP 实例上绑定相同的端口号（在不同的模块上），这样需要 V4 平台提供的分发方案支持优选功能，即：IPI 到 TP 实例的分发优先选择某个 TP 实例，在这个 TP 实例 down 掉后，V4 平台感知后，发到其他备份的 TP 实例。

此种方案下的分发实现描述如下：

## ✧ 初始请求消息

对于上行消息：IPI 根据相应的分发策略，将消息分发到某一模块上的 TPM 上，TPM 根据相应的策略选取一个 TP 实例，TP 实例处理完后，根据相应的策略选取一个 TR/TU 实例，将消息发给该 TR/TU 实例。如果不能直接获取到 TR/TU 的实例信息，则根据相应策略选取一个 TR/TU 管理进程，先将消息发送到 TR/TU 管理进程，然后由管理进程根据相应策略分发给 TR/TU 进程实例；TR/TU 进程实例可以将 TP 携带上来的实例信息保存至其数据区中，以使后续的消息经过相同的 TP 处理，减少 TP 实例的选择过程，而且也能够避免网元内的乱序问题；TR/TU 根据业务注册的分发表，将消息发送到相应的应用进程实例处理。

对于下行消息：如果上层应用与 SIP TR/TU 在同一个进程实例内，则上层应用可以直接将消息发给本实例内的 TR/TU，TR/TU 调用相应的 DB 接口获取本端的 IP-PORT，并在本网元添加的 record-route、via 中添加相应的端口号以使后续收到的响应及后续请求，直接发到 SIP 消息中所确定的 TP 实例上；在其中的 zte-did 参数中添加 TR/TU 的实例信息，以使后续的消息由 TP 直接发到 zte-did 中所确定的 TR/TU 实例上。TR/TU 处理完后，将消息发送到相应端口号所在的 TP 实例上，TP 实例处理完后直接将消息发送出去。TP 实例发送消息可以使用 TR/TU 所确定的本端端口，可考虑将 IP 包中的本端 IP-PORT 修改为 TPM 的监听端口，这需要 V4 平台支持此功能；另外 TP 实例发送消息也可以使用其他的端口，而不是 TR/TU 所确定的端口号，这两种方式均会造成，实际 IP 包中的地址与 TOP-VIA 中的不一致，使对方网元误认为消息经过了 NAT 处理。

## ✧ 响应消息

对于上行消息：IPI 根据相应的分发策略，将消息直接分发到 TOP-VIA 中端口号所在的 TP 实例上，TP 实例处理完后，根据 TOP-VIA 中 zte-did 参数所确定的 TR/TU 实例，将消息直接发给该 TR/TU 实例。TR/TU 根据数据区中所保存的上层应用实例信息，将消息发送到相应的应用进程实例处理。

对于下行消息：上层应用根据其数据区中保存的 TR/TU 实例信息，将消息发给 TR/TU，TR/TU 根据数据区中的相关内容重写 record-route，包括数据区索引及 TR/TU 所在实例信息。另外根据数据区中记录的 TP 实例信息，调用 DB 接口，获取该实例上的某一个端口号，并将端口号重写到 record-route 中，以保证 MO 侧发送的后续请求消息由相同的 TP 实例处理。TR/TU 根据数据区中保存的 TP 实例信息，将消息发给相应的 TP 实例，TP 实例处理完后将消息发出。

## ✧ 对话内请求消息

对于上行消息：IPI 根据相应的分发策略，将消息直接分发到 top-route 中端口号所在的 TP 实例上，TP 实例处理完后，根据 top-route 中 zte-did 参数所确定的 TR/TU 实例，将消息直接发给该 TR/TU 实例。如果消息中没有 top-route（如 CANCEL 和非成功响应的 ACK 消息），则相应的分发过程同初始请求。TR/TU 根据数据区中所保存的上层应用实例信息，



将消息发送到相应的应用进程实例处理。

对于下行消息：上层应用根据其数据区中保存的 TR/TU 实例信息，将消息发给 TR/TU，TR/TU 根据数据区中的相关内容，确定 TP 实例信息，并根据该信息添加 via 的 send-by port 参数，保证后续响应消息经过相同的 TP 实例处理。TP 实例处理完后将消息发出。

## 5.5 多实例情况下信令压缩相关分发流程

信令压缩功能出现在 UNI 侧，可能涉及到信令压缩流程的由 UE 和 PROXY 的 UNI 侧。这里仅考虑 PROXY 的 UNI 侧的情况，对于 UE 情况不做考虑。

在多模块的情况下，SIP 协议栈在收到用户注册消息时，根据用户请求创建 DID，并将在上行消息中携带 DID 实例的 Handle（Handle 中含有模块号）。上层应用使用 DID 实例 Handle 中的模块号发起 CID 创建请求，SIP TU 将创建请求发送到指定的 SIP TP 进程。这样 DID 与 CID 始终在同一个 SIP TP 进程创建。IPI 板子在分发的时候根据源 IP 分发，和同一个用户相关的 SIP 消息始终经过同一个 TP 进程处理。

在 SIP TP 多实例的情况下，对于 IPI 板子的要求仍然是按照源 IP 分发。

场景 1，IPI 根据源 IP 分发后，BRS 将消息直接发送至 SIP TP 处理进程（此时对应的 SOCKET 在 SIP TP 处理进程创建），在这种情况下 SIP 处理进程根据请求创建 DID，并将在上行消息中的 DID 实例的 Handle 内携带进程标识。上层应用使用 DID 实例 Handle 中的进程标识发起 CID 创建请求，SIP TU 将创建请求发送到指定的 SIP TP 处理进程。这样 DID 与 CID 始终在同一个 SIP TP 处理进程创建。

场景 2，IPI 根据源 IP 分发后，BRS 将消息直接发送至 SIP TP 管理进程（此时对应的 SOCKET 在 SIP TP 管理进程创建），SIP TP 管理进程选择 SIP TP 处理进程后将消息发送至 SIP TP 处理进程，SIP 处理进程根据请求创建 DID，并将在上行消息中的 DID 实例的 Handle 内携带进程标识。上层应用使用 DID 实例 Handle 中的进程标识发起 CID 创建请求，SIP TU 将创建请求发送到指定的 SIP TP 处理进程。这样 DID 与 CID 始终在同一个 SIP TP 处理进程创建。在这里要求 SIP TP 管理进程在为上行消息选择 SIP TP 处理进程时，需要根据源 IP 分发，以保证和同一用户的相关的 SIP 消息在同一个 SIP TP 处理进程处理。

注：在 SSS2.0 中不使用信令压缩功能；另外，这里 IPI 板子根据源地址分发是指 UDP 承载，且目的 IPPORT 可以在多个板子上同时存在的情况下；对于 TCP 承载或者目的端口仅在单一单板存在，则直接分发

## 5.6 多实例情况下 IPSec 相关分发流程

IPSec 功能出现在 UNI 侧，可能涉及到 IPSec 流程的为 UE 和 PROXY 的 UNI 侧。这里仅考虑 PROXY 的 UNI 侧的情况，对于 UE 情况不做考虑。

在多模块的情况下，上层应用发起创建 IPSec 请求，SIP TU 进程选择 TP 所在模块，SIP TP 进程在创建成功后发送握手信息，并在握手信息中携带了 IPSec SA 数据区实例的

Handle。与该用户相关的后续信息的接收后发送都会携带该 Handle，从而保证与该用户相关的信息都在同一个 SIP TP 进程处理。

在多实例情况下，由于 IPSec 功能是和 SOCKET 紧密绑定的，因此 SIP TP 的 IPSec 相关功能也随 SOCKET 走。如果 IPSec 相关的 SOCKET 端口配置在 SIP TP 处理进程，那 IPSec SA 的数据区创建与维护也在对应的 SIP TP 处理进程。如果 IPSec 相关的 SOCKET 端口配置在 SIP TP 管理进程，那 IPSec SA 的数据区创建与维护也在对应的 SIP TP 管理进程。

在创建 IPSecSA 数据区的流程中，SIP TU 在选择 SIP TP 时，通过 DB 接口获得对应的 SIP TP 进程（该进程可能是 SIP TP 处理进程也可能是 SIP TP 管理进程）

SIP TP 处理进程在收到 IPSec 创建请求时创将 IPSecSA，并将 SA 的 handle 在握手消息内携带，Handle 内带有处理进程标识。

SIP TP 管理进程在收到 IPSec 创建请求，判断对应的 SOCKET 是否在本进程创建，否则调用 DB 接口确定 SIP TP 处理进程，并将请求发送至 SIP TP 处理进程。如果在本进程创建则 SIP TP 管理进程创建 IPSecSA，并将 SA 的 handle 在握手消息内携带，Handle 内带有管理进程标识。

如果收发消息所用的 SOCKET 端口在 SIP TP 处理进程创建，则后续与承载间的 SIP 消息的接收和发送不经过 SIP TP 管理进程。下行的时候，SIP TU 可以根据 SA handle 内的进程标识直接确定 SIP TP 处理进程。其基本流程如下图

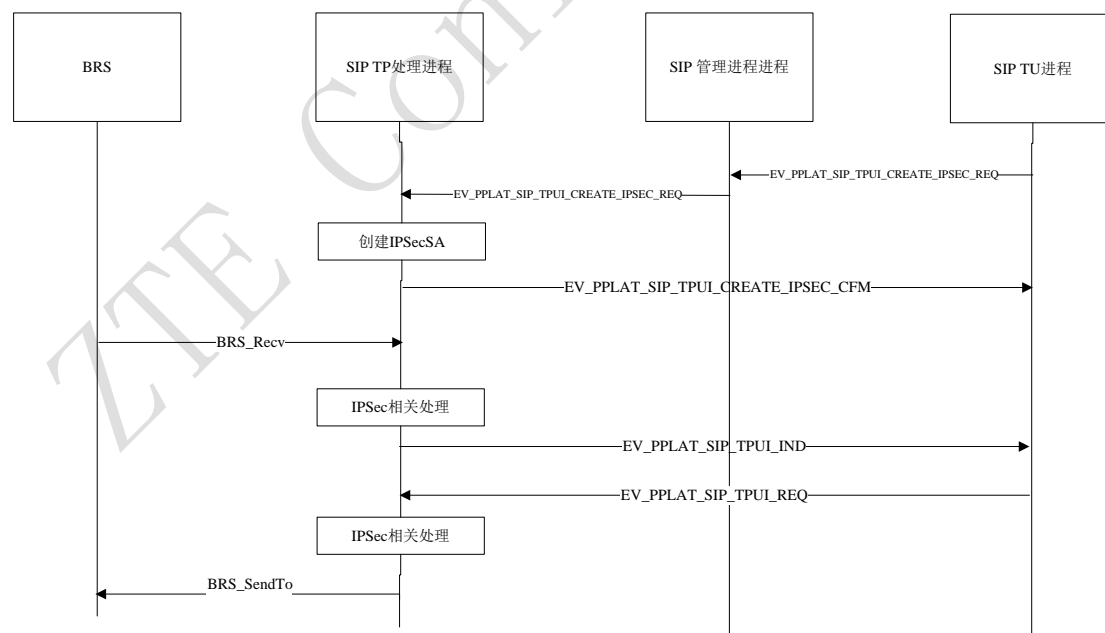


图 5.6.1 IPSec 相关分发流程（一）

如果收发消息所用的 SOCKET 端口在 SIP TP 管理进程创建，则下行的时候 SIP TU 根据 SA Handle 内的进程标识确定 SIP TP 进程为 SIP TP 管理进程，SIP TP 管理进程在收到

SIP 消息后调用 DB 接口选择 SIP TP 处理进程，SIP 处理完毕后将消息发送至 SIP TP 管理进程，SIP TP 管理进程在进行 IPsec 相关处理完毕后通过承载发送消息；在上行情况，SIP TP 管理进程从承载接收到 SIP 消息，在进行 IPsec 处理后，调用 DB 接口选择 SIP TP 处理进程，并将 SIP 消息发送至 SIP TP 处理进程。SIP TP 处理完毕后，将消息发送至 SIP TU 进程。其基本流程如下图：

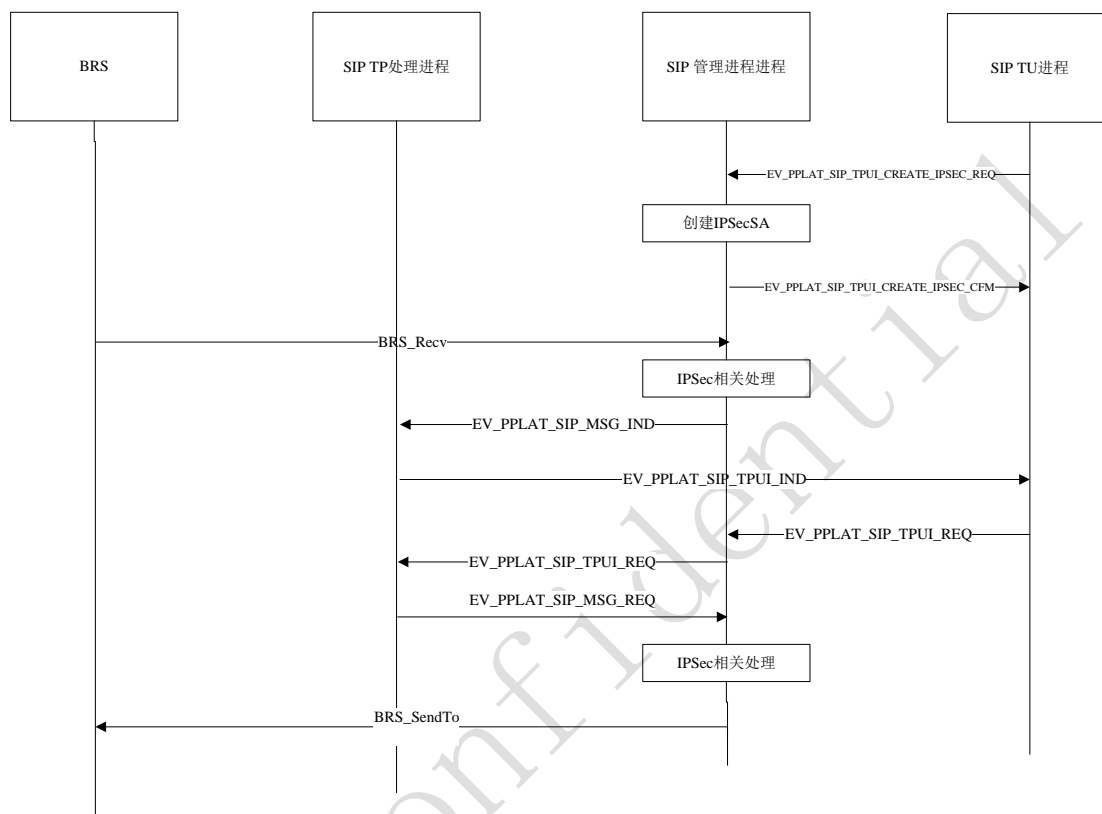


图 5.6.2 IPsec 相关分发流程（二）

注：在 SSS2.0 中不使用 IPsec 功能。

## 5.7 进程启动考虑

在 SIP 平台的多实例方案中，存在 SIP TP 处理进程、SIP TP 管理进程、SIP TR/TU 处理进程、SIP TR/TU 管理进程。

其中 SIP TP 的处理进程启动依赖于 SIP TP 管理进程，SIP TP 的处理进程需要在 SIP TP 管理进程启动成功后进行启动。SIP TP 处理进程在启动后，需要向 SIP TP 管理进程注册，在注册消息中携带 SIP TP 处理进程标识。

SIP TR/TU 的处理进程启动依赖于 SIP TR/TU 管理进程，SIP TR/TU 的处理进程需要在 SIP TR/TU 管理进程启动成功后进行启动。SIP TR/TU 处理进程在启动后，需要向 SIP TR/TU 管理进程注册，在注册消息中携带 SIP TR/TU 处理进程标识。

上述依赖过程通过阻塞实现，即在 SIP TP 或者 SIP TR/TU 处理进程收到上电消息后，调用判断相应管理进程状态的阻塞函数。

## 5.8 兼容性考虑

### 5.8.1 SIP 平台对外接口

SIP V1.01.10 版本中与上层应用相关的接口的修改如下：

- (1) PPLAT\_SIP\_MSG\_ATTR\_T 结构中使用了一个 Reserv 的 bit 位，具体修改参见接口说明。该修改不影响兼容性
- (2) PPLAT\_SIP\_Parser\_GetBodyFromTLV 函数中修改了 PPLAT\_SIP\_PARSER\_GETBODY\_IN\_T 中的 wCntDisposition 描述，具体修改参见接口说明。该修改不影响兼容性。
- (3) 增加编解码函数接口，该修改不影响兼容性。
- (4) 在 URL 结构定义中增加参数成员，结构变大了。对于那种通过分配固定大小的缓冲区，然后通过指针强转来使用该结构的地方，可能存在内存越界。
- (5) 在原有的接口函数中，增加了对新的头部、消息体类型的支持，该修改不影响兼容性。

### 5.8.2 SIP TP、SIP TR/TU 上电启动

在 SIP TP 和 SIP TR 进程的启动过程，为了兼容原有实现，考虑不启动管理进程的情况，通过编译宏来隔离阻塞和向管理进程注册处理。

### 5.8.3 JID 与原有 PID 兼容

在 SIP V1.00.10 版本的实现中，SIP 平台在内部直接看到 PID 的内部构成。而对于 ATCA 平台来说，与 PID 对应的概念是 JID，且其成员不同。

SIP 平台看到 PID 内部构成体现在如下几个方面

- (1) 在 IPSec Handle、CID Handle、DIDhandle 中定义了 Module 号，并且使用这些 module 来拼装 PID，上述结构对上层应用直接暴露，在 ATCA 的实现中 Handle 内的 Module 号被替换为进程标识。由于上层应用对于上述 Handle 的使用是透明的，因此该修改并不影响兼容性。
- (2) 在 SIP 平台封装的 DB 接口中定义了 Module 号并且使用这些 module 来拼装 PID，在 ATCA 的实现中 Module 号被替换为进程标识。出于兼容性考虑，SIP 平台通过编译宏来隔离新旧结构的差异。
- (3) 在 SIP 平台内部的对 PID 的拼装，由于 ATCA 中与 PID 对应的 JID 结构与 PID 不同，出于兼容性考虑。在 SIP 平台内部拼装 PID 的地方采用编译宏来隔绝与不同平台对应的实现。
- (4) 在 SIP 与业务接口中以 PPLAT\_SAP\_ID\_T 方式中的进程号和模块号方

式暴露的，上层应用使用这些来拼装 SIP 的 PID。在 ATCA 平台中上述进程号和模块号被替换为 PPLAT\_SIP\_PID\_T，具体接口修改可参见接口说明书。出于兼容性考虑，SIP 平台通过编译宏来隔离新旧结构的差异。如果上层应用与 V1.01 版本运行的在 ATCA 平台上需要对此进行同步修改。

#### 5.8.4 SIP 平台向上层应用分发

相对于 SIP V1.00.10 版本，SIP V1.01.10 版本还修改可重载接口 PPLAT\_SIP\_AppDistribution 出参和入参，出于兼容性考虑，在 SIP 平台内部使用该接口的地方通过编译宏来隔离两种实现方式。

#### 5.8.5 Record-route 相关处理

##### 5.8.5.1 添加 record-route 与修改 record-route

注：修订记录编号 0002

原有 SIP V1.00.10 版本中，在添加本地 record-route 时，对于添加域名形式的，只添加了域名，而在有些项目中要求指定端口号，可以通过在 PPLATAPP 中增加项目开关项，来控制此场景下是否添加指定端口到 record-route 中，如果指定添加，则将本端的端口号添加到 record-route 中。

响应下发时，在入呼侧需要修改 record-route，在 SIP V1.00.10 版本中，修改了 zte-did 的内容，如果 record-route 中添加了端口号，可能需要修改其中的端口号。可以通过在 PPLATAPP 中增加项目开关项，来控制此场景下是否修改 record-route 中的端口号，如果指定修改，则将入呼侧数据区的本端端口号修正到 record-route 中。

##### 5.8.5.2 Record-route 优化

注：修订记录编号 0013

SIP 平台此修改是为了兼容 SSS 中移测试的需求，对 UA 情况下 record-route 的填写处理进行优化，SIP 模块对于 Record-Route 的处理原则如下：

1、当 SIP 接收初始会话请求时（作为 UAS），需要创建路由集；（补充：sss1.6 的实现中，不论该初始 Invite 请求是否携带 Record-Route 头部，sss 回复的响应中均将自身 ip 地址加入 R-R 中，便于后续请求路由以及分发；但是由于 pss2.0 不存在分发的问题，因此 sip 平台的处理维持原来的处理不变，即回复的响应中不将自身 ip 地址加入 R-R 中）

2、当 SIP 发送初始会话请求时（作为 UAC），携带 R-R 头部且置头部为自身 ip 地址；接收 Invite 的响应（18x/2xx）时，需要创建路由集；

3、当 SIP 接收后续请求时（SIP 作 UAC 或 UAS），若该后续请求中未携带 R-R 头部，则 SIP 回复的该对话内请求对应响应中不携带 R-R 头部；若该后续请求中携带了 R-R 头部，

则 SIP 回复的该对话内请求对应响应中携带 R-R 头部，且 R-R 与初始 Invite 事务协商的保持一致而非该对话内请求中的 R-R 头部；

4、当 SIP 发送后续请求时（SIP 作 UAC 或 UAS），不携带 R-R 头部，携带 Route 头部指定路由（Route 头部从初始 Invite 事务创建的路由集获得）；接收后续请求的响应不再更新路由集；

#### 5.8.6 MAX-FORWARDS

在 B2BUA 模式下，上层业务可以选择是否关联两条 LEG 的 max-forwards，防止发生环回。

如果上层业务需要关联两条 LEG 的 max-forwards，SIP 平台只负责将 max-forwards 送上去，上层业务将 max-forwards 减 1 之后再将其放在请求下发。如果 SIP 平台送上去的 max-forwards 为 0，则上层业务应当回复 483 响应。

如果上层业务不需要关联两条 LEG 的 max-forwards，则 SIP 平台每次会生成新的 max-forwards，值为 70。

#### 5.8.7 链路、IP-PORT 变更的通知

注：修订记录编号 0003

SIP V1.00.10 版本中，在链路、IP-PORT 发生变更时（增加、删除、修改），SIP 平台收到变更通知消息，消息中指明了是增加、删除还是修改某个配置，SIP 平台直接根据消息进行相应操作。在 SIP V1.01.10 版本中，DB 通知 SIP 平台的变更通知消息中，没有指明是哪种变更（增加、删除、修改），这种情况下需要 SIP 平台内部根据通知消息，与数据区中的相关内容进行匹配，然后进行相关的处理。对于这两种情况的实现，可以考虑通过编译宏进行隔离。

#### 5.8.8 缺省实现接口

对于缺省实现接口变化引起的兼容性说明，请参见 11 章接口变更说明。

### 5.9 扩展性的考虑

参见《NGN PPLAT V1.00.10\_系统方案》

### 5.10 系统性能的考虑

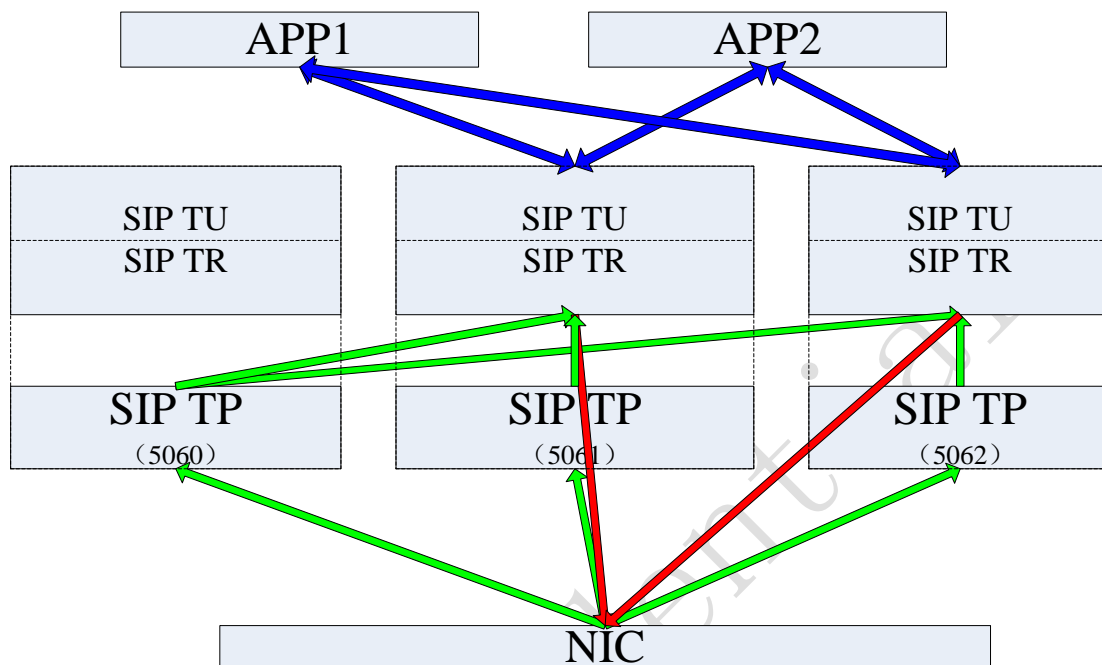
SIP 平台为了能够适应不同项目的需求，在设计的时候，将各分发点的实现封装在外，可以由项目进行重载，对于这些分发点的不同实现方式会使得 SIP 平台的性能有不同的表现。

SIP 平台 V1.01.10 版本首先应用于 SSS2.0 版本的 ATCA 平台下，对于 SIP 协议的性



能要求依据 SSS1.6 版本的情况提出改进，总的要求是与 SSS1.6 相比有提高，性能最低要求是不低于 SSS1.6 的 SIP 协议栈。

SSS1.6 中的分发示意图如下：



SSS2.0 中的分发示意图如图 5.4.20。

从上述示意图可知，从接入板（对于 SSS1.6 是 NIC，对于 SSS2.0 是 IPI）到上层应用间：

对于上行初始请求，SSS1.6 需要经过 3 次板间通讯，对于 SSS2.0 需要 2 次板间通讯和 2 次板内通讯。

对于下行初始请求，SSS1.6 需要经过 2 次板间通讯和 1 次板内通讯，对于 SSS2.0 需要 1 次板间通讯和 2 次板内通讯。

对于上行对话内消息，SSS1.6 需要经过 2 次板间通讯和 1 次板内通讯，对于 SSS2.0 需要 1 次板间通讯和 2 次板内通讯。

由于板内通讯的消耗比板间通讯要小的多，而其他的 SIP TP 和 SIP TU 的处理又基本不变，所以对于 SSS2.0 来说相，对于 SSS1.6 SIP 协议栈性能有所提升。

具体性能需求可以参见《NGN PPLAT SIP V1.01.10\_项目研制任务书.doc》。

## 5.11 安全性的考虑

无

## 5.12 部署考虑

参见《NGN PPLAT V1.00.10\_系统方案》

## 6 构架说明

### 6.1 组件及接口概览

参见《NGN PPLAT V1.00.10\_系统方案》

### 6.2 组件功能描述

#### 6.2.1 TU 层功能描述

参见《NGN PPLAT V1.00.10\_系统方案》

#### 6.2.2 TR 层功能描述

参见《NGN PPLAT V1.00.10\_系统方案》

#### 6.2.3 TP 层功能描述

参见《NGN PPLAT V1.00.10\_系统方案》

#### 6.2.4 PARSER 模块功能描述

参见《NGN PPLAT V1.00.10\_系统方案》

#### 6.2.5 SIGCOMP 模块功能描述

参见《NGN PPLAT V1.00.10\_系统方案》

#### 6.2.6 SIP0MM 功能描述

参见《NGN PPLAT V1.00.10\_系统方案》

#### 6.2.7 SIP TP 管理进程功能描述

SIP TP 管理进程的主要功能有：

1. SIP TP 上行分发和下行发送子模块，负责 IP 承载和 SIP TR 之间的消息和报文的上下行发送。当收到 IP 承载的上行消息和报文后，选择 SIP TP 处理实例并将上行消息发送至 SIP TP 处理实例。当收到 TR 的下行消息后，选择 SIP TP 处理进程实例，并将消息发送至 SIP TP 处理进程。在收到 SIP TP 处理进程发送的下行消息时，调用平台 SOCKET 发送接口，实现 SIP 消息的下行发送。

2. SOCKET 和 LINK 维护子模块。当具体项目将 SOCKET 和链路配置在 SIP TP 管理进程时，还需负责归属 SIP TP 管理进程的 SOCKET 的创建、绑定、侦听、连接、关闭等操作。负责链路的建立，以及通过 TP 层的心跳检测实现对链路状态的检测。并可执行网管发出的链路闭塞、解闭塞以及状态查询消息实现链路的动态管理。通过执行 DB 发出的配置更改消息，实现对 SOCKET 和链路的配置更改。另外还实现主备切换功能。

3. IPSEC 子模块的功能。该功能也是在由 1、2 引发的，需要实现子模块的功能主要



负责 IPSEC SA 的创建，SA 生命周期的修改、IPSEC SA 的删除，为 IP 承载在 UNI 侧的 IPsec ESP 和 AH 在隧道模式和传输模式下的解封装处理，以及协议规定加解密算法的实现提供支持。

4、OMM 相关功能消息的分发，以及其他功能。

### 6.2.8 SIP TR/TU 管理进程功能描述

SIP TR/TU 管理进程主要负责上下行消息的分发，当收到从 SIP TP 发送过来的上行消息时，选择 SIP TR/TU 处理进程实例，并将该消息发送至 SIP TP 处理进程；当收到从上层应用发送过来的消息消息时，选择 SIP TR/TU 处理进程实例，并将消息发送至 SIP TR/TU 处理进程。

SIP TR/TU 管理进程还有可能负责和 OMM 相关功能消息的分发。

## 7 组件标准化设计

参见《NGN PPLAT V1.00.10\_系统方案》

## 8 协作说明

### 8.1 功能协作

#### 8.1.1 TU 层处理

##### 8.1.1.1 PD-F-PPLAT-SIP-TUUI-0010 UA 模式下业务指示重传 1xx

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-TU-0020 重发 INVITE 的可靠 1XX/2XX 响应

**功能描述：**在 UA 模式下，SIP 协议栈收到 INVITE 请求，依据上层应用指示，进行 1XX 响应的重传。

**前置条件：**

C0010 在 UA 模式下，SIP 协议站收到 INVITE 请求

C0020 上层应用指示发送可靠 1xx

**后置条件：**

R0010 SIP 协议站根据应用指示负责 1xx 响应的重传

**正常过程：**

N0010 上层应用收到 INVITE 请求后，决定发送可靠的 1xx 响应，发送 EV\_PPLAT\_SIP\_API\_RESPONSE\_REQ 到 SIP TU，在消息中是否需要重传传输标志置为需要。

N0020 TU 收到 EV\_PPLAT\_SIP\_API\_RESPONSE\_REQ 后，设置重传定时器和重传保护定时器，通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 向事务层发送响应请求

N0030 事务层匹配到事务实例，通过 EV\_PPLAT\_SIP\_TPUI\_REQ 发送到 SIP 传输层

N0040 SIP 传输层通过 SOCKET 接口发送 SIP 响应消息。

N0050 SIP TU 重传定时器超时后重传 1xx 响应，重设重传定时器。

N0060 SIP 传输层收到 PRACK 消息通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

N0070 事务层通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

N0080 SIP TU 杀死重传定时器和重传保护定时器，通过 EV\_PPLAT\_SIP\_API\_SUBSEQUENT\_IND 发送至上层应用。

N0090 用例结束。

可选过程：A1 上层应用指示不需要重传 1xx

C0020 上层应用指示不需要重传 1xx

后置条件：

R0010 SIP 协议站发送 1xx，并不进行重传

A0010 上层应用收到 INVITE 请求后，决定发送可靠的 1xx 响应，发送 EV\_PPLAT\_SIP\_API\_RESPONSE\_REQ 到 SIP TU，在消息中是否需要重传标志置为不需要。

A0020 TU 收到 EV\_PPLAT\_SIP\_API\_RESPONSE\_REQ 后，通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 向事务层发送响应请求

A0030 事务层匹配到事务实例，通过 EV\_PPLAT\_SIP\_TPUI\_REQ 发送到 SIP 传输层

A0040 SIP 传输层通过 SOCKET 接口发送 SIP 响应消息。

A0050 用例结束。

A2 1xx 应答的重传保护定时器超时

前置条件：

C0010 上层应用指示需要重传 1xx

C0020 1xx 应答的重传保护定时器超时

后置条件：

R0010 释放呼叫

A0010 SIP TU 1xx 应答的重传保护定时器超时

A0020 SIP TU 释放将对话状态切换到待释放，同时通知上层应用释放呼叫

A0030 用例结束。

A3 1xx 的 PRACK 未到时业务发送需要重传的 2xx

前置条件：

C0010 上层应用指示发送需要重传的 1xx

C0020 1xx 未收到对应 PRACK

C0030 上层应用指示发送需要重传 2xx

后置条件:

R0010 1xx 停止重发

A0010 上层应用发送 EV\_PPLAT\_SIP\_API\_ESTABLISH\_REQ 到 SIP TU

A0020 TU 收到 EV\_PPLAT\_SIP\_API\_ESTABLISH\_REQ 后杀掉 1xx 重传定时器、1xx 重传保护定时器，释放缓存的 1xx，通过 EV\_PPLAT\_SIP\_TPUI\_SYNC\_IND 向事务层发送响应请求

A0030 事务层匹配到事务实例，通过 EV\_PPLAT\_SIP\_TPUI\_REQ 发送到 SIP 传输层

A0040 用例结束。

异常过程: E1 并发可靠临时响应

前置条件:

C0010 SIP TU 发出了需要重传的临时响应未收到对应 PRACK

00020 上层应用指示发送新的需要重传时响应

后置条件:

R0010 新的需要重传的临时响应被丢弃，同时向上层应用报错

E0010 上层应用发送 1xx 响应，发送 EV\_PPLAT\_SIP\_API\_RESPONSE\_REQ 到 SIP TU，在消息中是否需要重传标志置为需要。

E0020 TU 收到 EV\_PPLAT\_SIP\_API\_RESPONSE\_REQ 后，检查先前发送的可靠临时响应尚未收到 PRACK；SIP TU 丢弃新的临时响应，发送 EV\_PPLAT\_SIP\_API\_ABORT\_IND 给上层应用。

E0030 用例结束。

包含: 无

特殊要求: 无

#### 8.1.1.2 PD-F-PPLAT-SIP-TUUI-0020 UA 模式下业务指示发出需要重传的 2xx 响应

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-TU-0020 重发 INVITE 的可靠 1XX/2XX 响应

功能描述: 在 UA 模式下，SIP 协议栈收到 INVITE 请求，依据上层应用指示发出需要重传的 2XX 响应。

前置条件:

C0020 上层应用指示发送需要重传的 2xx

后置条件:

R0010 SIP 协议站发送 2xx，并负责其重传

正常过程:

N0010 上层应用收到 INVITE 请求后，决定发送 2xx 响应，发送

EV\_PPLAT\_SIP\_API\_ESTABLISH\_REQ 到 SIP TU, 在消息中是否需要重传标志置为需要。

N0020 TU 收到 EV\_PPLAT\_SIP\_API\_ESTABLISH\_REQ 后, 设置重传定时器和重传保护定时器, 通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 向事务层发送响应请求

N0030 事务层匹配到事务实例, 通过 EV\_PPLAT\_SIP\_TPUI\_REQ 发送到 SIP 传输层

N0040 SIP 传输层通过 SOCKET 接口发送 SIP 响应消息。

N0050 SIP TU 重传定时器超时后重传 2xx 响应, 重设重传定时器。

N0060 SIP 传输层收到 ACK 消息通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

N0070 事务层通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

N0080 SIP TU 杀死重传定时器和重传保护定时器, 通过 EV\_PPLAT\_SIP\_API\_SUBSEQUENT\_IND 发送至上层应用。

N0090 用例结束。

**可选过程:** A1 上层应用指示发送不需要重传的 2xx

**前置条件:**

C0020 上层应用指示发送不需要重传的 2xx

**后置条件:**

R0010 SIP 协议站发送 2xx, 并不负责重传

A0010 上层应用收到 INVITE 请求后, 决定发送 2xx 响应, 发送 EV\_PPLAT\_SIP\_API\_ESTABLISH\_REQ 到 SIP TU, 在消息中是否需要重传的标志置为不需要。

A0020 TU 收到 EV\_PPLAT\_SIP\_API\_ESTABLISH\_REQ 后, 通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 向事务层发送响应请求

A0030 事务层匹配到事务实例, 通过 EV\_PPLAT\_SIP\_TPUI\_REQ 发送到 SIP 传输层

A0040 SIP 传输层通过 SOCKET 接口发送 SIP 响应消息。

A0050 用例结束。

**A2 2xx 应答的重传保护定时器超时**

**前置条件:**

C0010 上层应用指示发送需要重传的 2xx

C0020 2xx 应答的重传保护定时器超时

**后置条件:**

R0010 释放呼叫

A0010 SIP TU 2xx 应答的重传保护定时器超时

A0020 SIP TU 释放将对话状态切换到待释放, 同时通知上层应用释放呼叫

A0030 用例结束。

**异常过程:** 无

包含：无

特殊要求：无

### 8.1.1.3 PD-F-PPLAT-SIP-TUUI-0030 UA 模式下 SIP 协议站收到重传的 1xx 响应

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-TU-0020 重发 INVITE 的可靠 1XX/2XX 响应

**功能描述：**在 UA 模式下，SIP 协议栈发送 INVITE 请求后，在收到重传的 1xx 响应时，根据配置决定是否吸收重传的响应。

**前置条件：**

C0010 在 UA 模式下，SIP 协议站发送 INVITE 请求

C0020 SIP 协议栈收到重传的 1xx 响应，

C0030 SIP 协议栈被配置为吸收重传的 1xx 响应

**后置条件：**

R0010 SIP 协议站吸收重传的 1xx 响应，不将重传消息上报业务。

**正常过程：**

N0010 SIP 传输层收到 1xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

N0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

N0030 SIP TU 通过 RSeq 匹配后确认为重传的 1xx，读取配置为吸收。不将消息发送至上层应用

N0040 用例结束。

**可选过程：A1 不吸收重传 1xx**

**前置条件：**

C0010 在 UA 模式下，SIP 协议站发送 INVITE 请求

C0020 SIP 协议栈收到重传的 1xx 响应，

C0030 SIP 协议栈被配置为不吸收重传的 1xx 响应

**后置条件：**

R0010 SIP 协议站不吸收重传的 1xx 响应，将重传消息上报业务。

A0010 SIP 传输层收到 1xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

A0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

A0030 SIP TU 通过 RSeq 匹配后确认为重传的 1xx，读取配置为不吸收。发送响应指示 EV\_PPLAT\_SIP\_API\_RESPONSE\_IND 到上层应用

A0040 用例结束。

异常过程：无

包含：无

特殊要求：无

#### 8.1.1.4 PD-F-PPLAT-SIP-TUUI-0040 UA 模式下 SIP 协议站收到重传的 2xx 响应

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-TU-0020 重发 INVITE 的可靠 1XX/2XX 响应

功能描述：在 UA 模式下，SIP 协议栈发送 INVITE 请求后，在收到重传的 2xx 响应时，根据配置决定是否吸收重传的响应。

前置条件：

C0010 在 UA 模式下，SIP 协议站发送 INVITE 请求

C0020 SIP 协议栈收到重传的 2xx 响应，

C0030 SIP 协议栈被配置为吸收重传的 2xx 响应

后置条件：

R0010 SIP 协议站吸收重传的 2xx 响应，不将重传消息上报业务。

正常过程：

N0010 SIP 传输层收到 2xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

N0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

N0030 SIP TU 检查后确认为重传的 2xx，读取配置为吸收。不将消息发送至上层应用

A0040 用例结束。

可选过程：A1 不吸收重传 2xx

前置条件：

C0010 在 UA 模式下，SIP 协议站发送 INVITE 请求

C0020 SIP 协议栈收到重传的 2xx 响应，

C0030 SIP 协议栈被配置为不吸收重传的 1xx 和 2xx 响应

后置条件：

R0010 SIP 协议站不吸收重传的 2xx 响应，将重传消息上报业务。

A0010 SIP 传输层收到 2xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

A0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

A0030 SIP TU 检查后确认为重传的 2xx，读取配置为不吸收。发送响应指示 EV\_PPLAT\_SIP\_API\_ESTABLISH\_IND 到上层应用。

A0040 用例结束。

异常过程：无

包含：无

特殊要求：无

#### 8.1.1.5 PD-F-PPLAT-SIP-TUUI-0050 TU 管理进程收到下行初始请求

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

**功能概述：**TU 管理进程收到下行初始请求时负责选择 SIP TU 处理进程，并将消息转发至 SIP TU 处理进程

**前置条件：**C1

C0010 TU 管理进程收到 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ

**后置条件：**

R0010 TU 管理进程将 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ 转发至 TU 处理进程

**正常过程：**

N0010: TU 管理进程收到 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ 消息，用例开始。

N0020: 调用下行消息分发选择接口，选择 SIP TU 处理进程成功

N0030: 发送 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ 消息到选择的 TU 处理进程

N0040: 用例结束。

**可选过程：**无

**异常过程：**选择 TU 处理进程失败

E0010: TU 管理进程收到 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ 消息，用例开始。

E0020: 调用下行消息分发选择接口，选择 SIP TU 处理进程失败

E0030: TU 管理进程丢弃 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ 消息，上报失败观察。发送 EV\_PPLAT\_SIP\_API\_ABORT\_IND 至上层应用。

E0040: 用例结束。

包含：无

特殊要求：无

#### 8.1.1.6 PD-F-PPLAT-SIP-TUUI-0060 TU 管理进程收到下行 Forking 初始请求



注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

**功能概述：**TU 管理进程收到下行初始 Forking 请求时负责选择 SIP TU 处理进程，并将消息转发至 SIP TU 处理进程

**前置条件：**C1

C0010 TU 管理进程收到 EV\_PPLAT\_SIP\_API\_FORKING\_REQ

**后置条件：**

R0010 TU 管理进程将 EV\_PPLAT\_SIP\_API\_FORKING\_REQ 转发至 TU 处理进程

**正常过程：**

N0010: TU 管理进程收到 EV\_PPLAT\_SIP\_API\_FORKING\_REQ 消息，用例开始。

N0020: 调用下行消息分发选择接口，选择 SIP TU 处理进程成功

N0030: 发送 EV\_PPLAT\_SIP\_API\_FORKING\_REQ 消息到选择的 TU 处理进程

N0040: 用例结束。

**可选过程：**无

**异常过程：E1 选择 TU 处理进程失败**

E0010: TU 管理进程收到 EV\_PPLAT\_SIP\_API\_FORKING\_REQ 消息，用例开始。

E0020: 调用下行消息分发选择接口，选择 SIP TU 处理进程失败

E0030: TU 管理进程丢弃 EV\_PPLAT\_SIP\_API\_FORKING\_REQ 消息，上报失败观察。发送 EV\_PPLAT\_SIP\_API\_ABORT\_IND 至上层应用。

E0040: 用例结束。

**包含：**无

**特殊要求：**无

#### 8.1.1.7 PD-F-PPLAT-SIP-TUUI-0070 TU 管理进程收到上行消息

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

**功能概述：**TU 管理进程收到上行消息时负责选择 SIP TU 处理进程，并将消息转发至 SIP TU 处理进程



前置条件: C1

C0010 TU 管理进程收到 EV\_PPLAT\_SIP\_TPUI\_IND

后置条件:

R0010 TU 管理进程将 EV\_PPLAT\_SIP\_TPUI\_IND 转发至 TU 处理进程

正常过程:

N0010: TU 管理进程收到 EV\_PPLAT\_SIP\_TPUI\_IND 消息, 用例开始。

N0020: 调用上行消息分发选择接口, 选择 SIP TU 处理进程成功

N0030: 发送 EV\_PPLAT\_SIP\_TPUI\_IND 消息到选择的 TU 处理进程

N0040: 用例结束。

可选过程: 无

异常过程: E1 选择 TU 处理进程失败

E0010: TU 管理进程收到 EV\_PPLAT\_SIP\_TPUI\_IND 消息, 用例开始。

E0020: 调用下行消息分发选择接口, 选择 SIP TU 处理进程失败

E0030: TU 管理进程丢弃 EV\_PPLAT\_SIP\_TPUI\_IND 消息, 上报失败观察。如果是请求, 则构造并发送错误响应

E0040: 用例结束。

包含: 无

特殊要求: 无

#### 8.1.1.8 PD-F-PPLAT-SIP-TUUI-0080 UA 模式下初始请求携带会话检测刷新方参数

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-TU-0030 支持初始请求携带会话检测刷新方参数

**功能概述:** TU 收到业务指示发送初始请求 INVITE, 如果支持会话检测功能, 且支持初始请求携带会话检测刷新方参数的开关开启, 则 INVITE 的 Session-Expire 添加 refresher=uac。

前置条件:

C0010 支持会话检测

C0020 初始请求携带会话检测刷新方参数的开关开启

后置条件:

R0010 会话定时器生效

正常过程:

N0010: TU 收到上层应用 INVITE 请求, 用例开始。

N0020: TU 层按照本地配置添加 Session-Expires 参数, 并携带参数 refresher=uac

N0030: TU 层添加 supported timer 的 options tag

N0040: 用例结束。

**可选过程:** 初始请求携带会话检测刷新方参数的开关开启, 收到初始 INVITE 2XX 响应过程

A0010 TU 层收到事务层的 INVITE 2XX 响应, 用例开始

A0020 TU 层检查其中是否携带 Session-Expires 参数, 未携带, 用例结束

A0030 根据 Session-Expires, 启动会话保护定时器, 检查 refresher 参数, 如果携带了 refresher 参数, 则按照 2XX 的 refresher 参数更新刷新方; 如果不携带, 则按照保存数据区的刷新方进行会话刷新, 设置会话更新定时器为 Session-Expires/2.

A0040 会话更新定时器超时, 主动构造会话刷新 UPDATE 消息 发送

A0050 更新会话刷新超时定时器

A0060 重设会话更新定时器

A0070 用例结束

**异常过程:** 无

**包含:** 无

**特殊要求:** 无

### 8.1.2 TP 层处理

注: TP 管理进程和链路维护、SOCKET、IPSec 相关功能的处理与写作与 SIP TP 处理进程一致, 这里就不再单独描述, 参见《NGN PPLAT V1.00.10\_系统方案.doc》

#### 8.1.2.1 PD-F-PPLAT-SIP-TPUI-0010 TP 管理进程收到下行消息 EV\_PPLAT\_SIP\_TPUI\_REQ

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

**功能概述:** TP 管理进程收到下行消息时负责选择 SIP TP 处理进程, 并将消息转发至 SIP TP 处理进程

**前置条件:** C1

C0010 TP 管理进程收到 EV\_PPLAT\_SIP\_TPUI\_REQ

后置条件:

R0010 TP 管理进程将 EV\_PPLAT\_SIP\_TPUI\_REQ 转发至 TU 处理进程

正常过程:

N0010: TP 管理进程收到 EV\_PPLAT\_SIP\_TPUI\_REQ 消息, 用例开始。

N0020: 调用下行消息分发选择接口, 选择 SIP TP 处理进程成功

N0030: 发送 EV\_PPLAT\_SIP\_TPUI\_REQ 消息到选择的 TP 处理进程

N0040: 用例结束。

可选过程: 无

异常过程: 选择 TP 处理进程失败

E0010: TP 管理进程收到 EV\_PPLAT\_SIP\_TPUI\_REQ 消息, 用例开始。

E0020: 调用下行消息分发选择接口, 选择 SIP TU 处理进程失败

E0030: TP 管理进程丢弃 EV\_PPLAT\_SIP\_TPUI\_REQ 消息, 上报失败观察。发送 EV\_PPLAT\_SIP\_TPUI\_ERR\_IND 至上层应用。

E0040: 用例结束。

包含: 无

特殊要求: 无

#### 8.1.2.2 PD-F-PPLAT-SIP-TPUI-0020 TP 管理进程收到下行消息 EV\_PPLAT\_SIP\_MSG\_REQ

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

功能概述: TP 管理进程收到下行消息 EV\_PPLAT\_SIP\_MSG\_REQ 时负责将消息通过承载发送出去。

前置条件: C1

C0010 TP 管理进程收到 EV\_PPLAT\_SIP\_MSG\_REQ

后置条件:

R0010 TP 管理进程通过承载将 SIP 消息发送出去

正常过程:

N0010: TP 管理进程收到 EV\_PPLAT\_SIP\_MSG\_REQ 消息, 用例开始。

N0020: TP 管理进程成功通过承载将消息发送出去。

N0030: 用例结束。

可选过程：无

异常过程：TP 管理进程未能将消息发送出去

E0010: TP 管理进程收到 EV\_PPLAT\_SIP\_MSG\_REQ 消息，用例开始。

E0020: TU 管理未能将消息通过承载发送出去，丢弃消息，上报失败观察。发送 EV\_PPLAT\_SIP\_TPUI\_ERR\_IND 至 TR。

E0030: 用例结束。

包含：无

特殊要求：无

### 8.1.2.3 PD-F-PPLAT-SIP-TPUI-0030 TP 管理进程收到上行消息

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

功能概述：TP 管理进程从承载或者接入模块收到上行消息时，负责选择 TP 处理进程，并将消息转发到选择的 TP 处理进程。

前置条件：C1

C0010 TP 管理进程收到上行消息

后置条件：

R0010 发到选择的 TP 处理进程。

正常过程：

N0010: TP 管理进程上行消息，用例开始。

N0020: TP 管理进程选择 SIP TP 处理进程成功；

N0030: TP 管理进程发送 EV\_PPLAT\_SIP\_MSG\_IND 至 SIP TP 处理进程

N0040: 用例结束。

可选过程：无

异常过程：TP 管理进程选择 SIP TP 处理进程失败

E0010: TP 管理进程收到上行消息，用例开始。

E0020: TP 管理进程选择 SIP TP 处理进程失败；

E0030: TP 丢弃消息，上报失败观察。

E0040: 用例结束。

包含：无

特殊要求：无

#### 8.1.2.4 PD-F-PPLAT-SIP-TPUI-0040 TP 处理进程收到消息 EV\_PPLAT\_SIP\_MSG\_IND

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

SIP TP 处理进程收到 EV\_PPLAT\_SIP\_MSG\_IND 的处理可以参见《NGN PPLAT V1.00.10\_系统方案.doc》中 TP 处理进程上行报文处理部分。

### 8.1.3 分发处理

#### 8.1.3.1 PD-F-PPLAT-SIP-DISPATCH-0010 呼叫建立过程中的入呼侧分发

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

**功能描述：**协作描述了 TP 层接收到初始请求，经事务层，TU 层，上层应用到呼叫最终建立过程中，消息在各层间的传递及分发协作。

**前置条件：**

C0010 呼叫尚未建立

C0020

**后置条件：**

R0010 呼叫建立成功

**正常过程：N1**

N0010 底层承载或者接入模块收到呼叫建立请求消息(INVITE)将消息分发的 SIP 传输层管理进程

N0020 SIP 传输层管理进程根据策略选择 SIP TP 处理进程，并将 SIP 消息发送到 SIP TP 处理进程

N0030 SIP TP处理进程收到SIP消息后进行预解码，根据策略选择上行目的进程 SIP TR/TU 进程 为 SIP TR/TU 管理进程，并发送 SIP 消息 EV\_PPLAT\_SIP\_TPUI\_IND到SIP TR/TU管理进程

N0040 SIP TR/TU 管理进程根据策略选择 SIP TR/TU 处理进程，并将 SIP 消息发送到 SIP TR/TU 处理进程

N0050 SIP TR/TU 处理进程在创建事务，创建 TULEG 处理完毕后，根据策略选择上层应用进程，并发送 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_IND 到上层应用进程，在消息中携带本处理进程的标识。

N0060 上层应用在处理完毕后，根据收到消息中的进程表示发送 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_IND\_ACK到SIP TU处理进程，在消息中携带上层应用进程标识，SIP TU将消息中的进程标识保存在数据区中

N0070 上层应用根据保存的 SIP TU 进程标识发送到

EV\_PPLAT\_SIP\_API\_RESPONSE\_REQ (200) 到SIP TU处理进程

N0080 SIP TR/TU处理进程在处理完毕后, 根据策略选择下行目的进程SIP TP进程为SIP TP管理进程, 发送EV\_PPLAT\_SIP\_TPUI\_REQ(200)到SIP TP管理进程

N0090 SIP TP 管理进程根据策略选择 SIP TP 处理进程, 发送EV\_PPLAT\_SIP\_TPUI\_REQ(200)到SIP TP 处理进程。

N0100 SIP TP 处理进程将收到的 SIP 消息通过 SOCKET 接口发送 SIP 消息

N0110 承载将 ACK 消息发送到 SIP TP 处理进程

N0120 SIP TP 处理进程根据策略确定上行目的进程 SIP TR /TU 进程为 SIP TR/TU 处理进程

N0130 SIP TR/TU 处理进程处理完毕后, 根据 TULEG 内保存的上层应用进程标识发送消息到上层应用进程

N0140 用例结束

可选过程: A1 承载或者接入模块直接将初始请求发送至

A0010 底层承载或者接入模块收到呼叫建立请求消息(INVITE)将消息分发的 SIP 传输层处理进程

A0020 转 N0030

A2 SIP TP 选择上行目的进程为 SIP TR/TU 处理进程

A0010 承 N0020

A0020 SIP TP处理进程收到SIP消息后进行预解码, 根据策略选择上行目的进程

SIP TR/TU 进程 为 SIP TR/TU 处理进程, 并发送 SIP 消息EV\_PPLAT\_SIP\_TPUI\_IND 到 SIP TR/TU 处理进程

A0030 转 N0050

A3 SIP TR/TU 选择下行目的进程为 SIP TP 处理进程

A0010 承 N0070

A0020 SIP TR/TU 处理进程在处理完毕后, 根据策略选择下行目的进程 SIP TP 进程为SIP TP 处理进程, 发送 EV\_PPLAT\_SIP\_TPUI\_REQ(200)到 SIP TP 处理进程

A0030 转 N0100

A4 承载选择或者接入模块将后续请求(ACK)发送到 SIP TP 管理进程

A0010 承 N0100

A0020 承载将 ACK 消息发送到 SIP TP 管理进程

A0030 SIP TP 管理进程根据策略选择处理进程, 并将消息发送到处理进程

A0040 转 N0120

A4 SIP TP 选择上行目的进程 SIP TR/TU 管理进程

A0010 承 N0110

A0020 SIP TP处理进程根据策略确定上行目的进程SIP TR /TU进程为SIP TR/TU管理进程

A0030 SIP TR/TU管理进程根据策略确定SIP TR/TU处理进程

A0040 转 N0130

**异常过程：**E01 SIP TP 管理进程选择 SIP TP 处理进程失败

E0010 报告失败观察，丢弃消息

E02 SIP TP 处理进程选择 SIP TR/TU 目的进程失败

E0010 报告失败观察，丢弃消息

E03 SIP TR/TU 管理进程选择 SIP TR/TU 处理进程失败

E0010 报告失败观察，丢弃消息。如果是下行消息，则向上层应用上报  
EV\_PPLAT\_SIP\_API\_ABORT\_IND 消息

E04 SIP TR/TU 处理进程选择上层应用进程失败

E0010 报告失败观察，如果是请求，构造并回送响应。

E05 SIP TR/TU 处理进程选择下行目的进程失败

E0010 报告失败观察，丢弃消息，发送 EV\_PPLAT\_SIP\_API\_ABORT\_IND 给上层应

用

**包含：**无

**特殊要求：**无

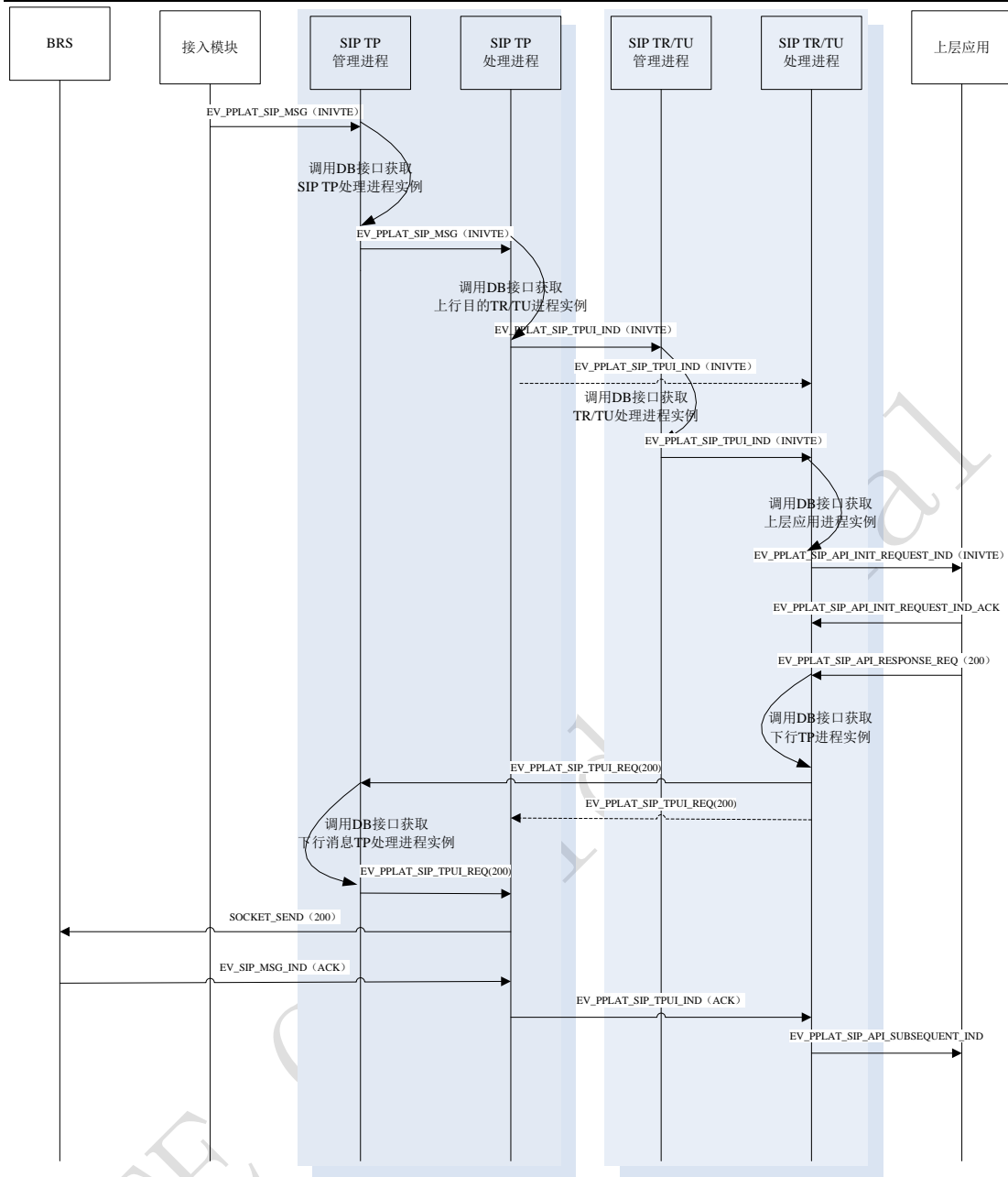


图 8.1.12

### 8.1.3.2 PD-F-PPLAT-SIP-DISPATCH-0020 呼叫建立过程中的出呼侧分发

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

**功能描述：**协作描述了 SIP 平台接收到上层应用发送初始请求，经 TU 层，事务层，传输层到呼叫最终建立过程中，消息在各层间的传递及分发协作。

**前置条件：**

C0010 呼叫尚未建立

C0020

**后置条件：**



R0010 呼叫建立成功

正常过程: N1

N0010 上层应用发送初始请求消息(INVITE)将消息

EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ 分发的 SIP TR/TU 管理进程

N0020 SIP TR/TU 管理进程根据策略选择 SIP TU 处理进程, 并将 SIP 消息

EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ 发送到 SIP TU 处理进程

N0030 SIP TR/TU 处理进程处理后根据消息中携带的上层应用标识发送

EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ\_ACK给上层应用进程, 并在消息中携带TR/TU处理进程标识;

N0040 SIP TR/TU处理完毕后根据策略选择下行目的进程SIP TP进程为SIP TP管理进程, 发送SIP消息EV\_PPLAT\_SIP\_TPUI\_REQ到SIP TP管理进程

N0050 SIP TP 管理进程根据策略选择 SIP TP 处理进程, 并将 SIP 消息 EV\_PPLAT\_SIP\_TPUI\_REQ 发送到 SIP TP 处理进程

N0060 SIP TP 处理进程在处理完毕后通过承载将消息发送出去

N0070 承载或者底层接入模块发送INVITE 200OK消息到SIP TP管理进程

N0080 SIP TP管理进程根据策略选择SIP TP处理进程, 并将消息发送到SIP TP处理进程

N0090 SIP TP处理进程处理外部后, 根据策略确定上行目的进程SIP TR/TU进程为 SIP TR/TU管理进程, 发送EV\_PPLAT\_SIP\_TPUI\_IND到SIP TR/TU管理进程

N0100 SIP TR/TU 管理进程根据策略确定 SIP TR/TU 处理进程, 发送 EV\_PPLAT\_SIP\_TPUI\_IND到SIP TR/TU处理进程;

N0110 SIP TR/TU 处理进程处理完毕后, 根据数据区内保存的上层应用进程标识发送 EV\_PPLAT\_SIP\_API\_ESTABLISH\_RESPONSE\_REQ 到上层应用

N0120 上层应用发送 EV\_PPLAT\_SIP\_API\_SUBSEQUENT\_REQ(ACK) 到 SIP TR/TU 处理进程

N0130 SIP TR/T 处理进程处理完毕后, 根据策略选择下行目的进程 SIP TP 进程为 SIP TP 管理进程, 并发送 EV\_PPLAT\_SIP\_TPUI\_REQ 到 SIP TP 管理进程

N0140 SIP TP 管理进程根据策略选择 SIP TP 处理进程, 并发送 EV\_PPLAT\_SIP\_TPUI\_REQ 到 SIP TP 处理进程

N0150 SIP TP 处理进程处理完毕后, 通过承载发送 SIP 消息。

N0160 用例结束

可选过程: A1 上层应用直接将初始请求发送至 SIP TR/TU 处理进程

A0010 上层应用直接将初始请求 EV\_PPLAT\_SIP\_API\_INIT\_REQUEST\_REQ (INVITE) 发送至 SIP TR/TU 处理进程

A0020 转 N0030

A2 SIP TR/TU 处理进程选择下行目的进程为 SIP TP 处理进程

A0010 承 N0030

A0020 SIP TR/TU处理进程处理完毕后，根据策略选择下行目的进程

SIP TP 进程为 SIP TP 处理进程，并发送 SIP 消息  
EV\_PPLAT\_SIP\_TPUI\_REQ 到 SIP TP 处理进程

A0030 转 N0060

A3 SIP 底层承载或者接入模块直接将 SIP 消息发送到 SIP TP 处理进程

A0010 承 N0060

A0020 底层承载或者接入模块直接将 SIP 消息发送到 SIP TP 处理进程

A0030 转 N0090

A4 SIP TP 处理进程选择（200 OK）上行目的进程为 SIP TR/TU 处理进程

A0010 承 N0080

A0020 SIP TP 处理进程处理外部后，根据策略确定上行目的进程 SIP TR/TU 进程为  
SIP TR/TU 处理进程，发送 EV\_PPLAT\_SIP\_TPUI\_IND 到 SIP TR/TU 处理进程

A0030 转 N0110

A4 SIP SIP TR/TU 处理进程选择（ACK）下行目的进程为 SIP TP 处理进程

A0010 承 N0120

A0020 SIP TR/T处理进程处理完毕后，根据策略选择下行目的进程SIP TP进程为SIP  
TP处理进程，并发送EV\_PPLAT\_SIP\_TPUI\_REQ到SIP TP处理进程

A0030 转 N0150

**异常过程：**E01 SIP TU 管理进程选择 SIP TU 处理进程失败

E0010 报告失败观察，丢弃消息，如果是下行消息，则向上层应用上报  
EV\_PPLAT\_SIP\_API\_ABORT\_IND 消息

E02 SIP TU 处理进程选择下行目的进程失败

E0010 报告失败观察，向上层应用上报 EV\_PPLAT\_SIP\_API\_ABORT\_IND 消息，丢  
弃消息，

E03 SIP TP 管理进程选择 SIP TP 处理进程失败

E0010 报告失败观察，丢弃消息。

E04 SIP TP 处理进程选择上行目的进程失败

E0010 报告失败观察，丢弃消息。

E05 SIP TR/TU 处理进程选择上行目的进程失败

E0010 报告失败观察，丢弃消息，如果是请求则发送响应

包含：无

特殊要求：无

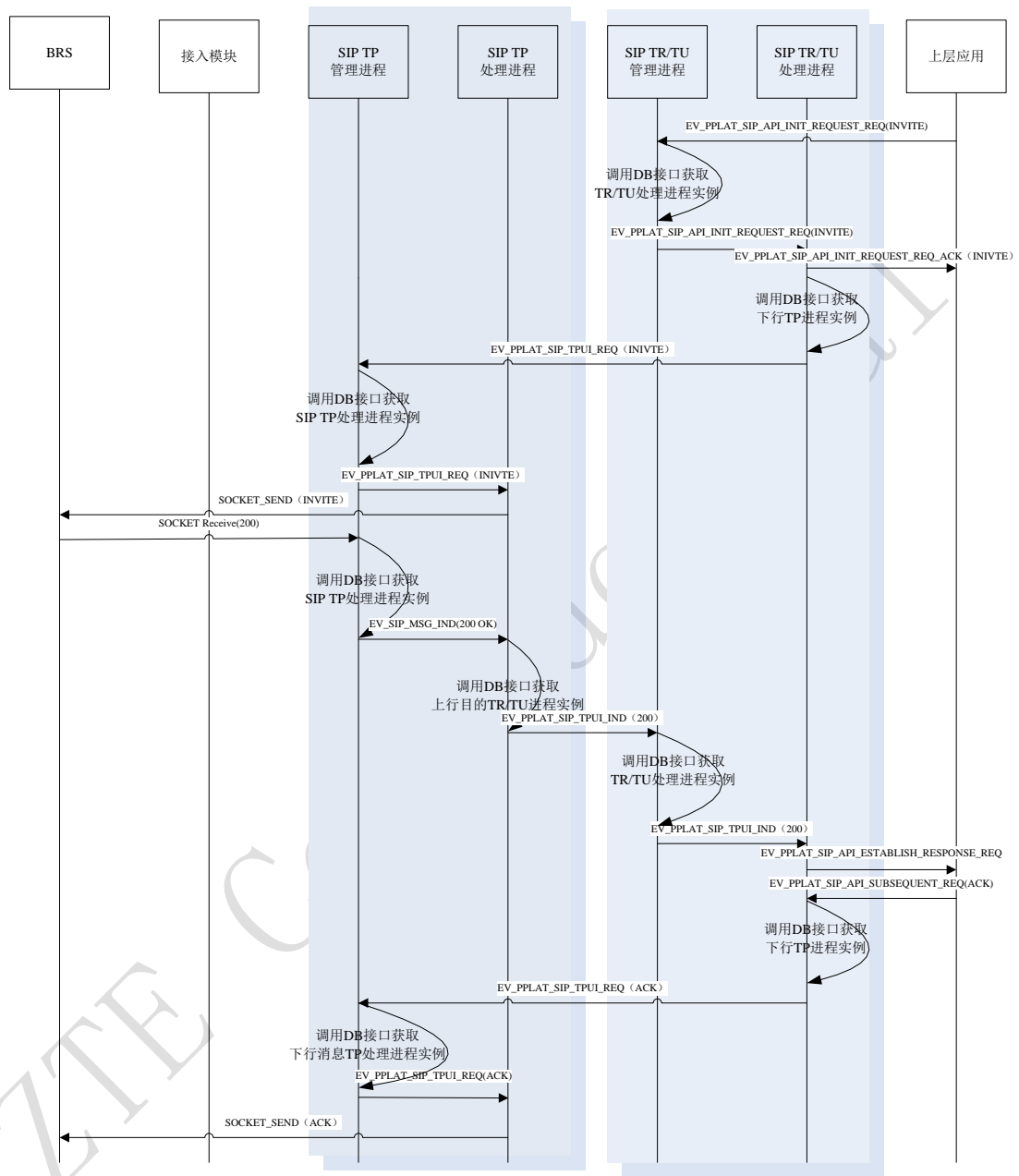


图 8.1.13

## 8.1.4 内存分配器

### 8.1.4.1 PD-F-PPLAT-SIP-ALLOCATOR-0010 创建内存分配器

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为  
SR-F-PSS-SIP-PARSER-0090 编解码支持内存分配器

功能概述：SIP 平台各模块可以创建内存分配器

前置条件: C1

C0010 创建内存分配器

后置条件:

R0010 成功创建内存分配器。

正常过程: N1 初始化内存分配器

N0010: 收到创建内存分配器，用例开始。

N0020: 调用 P\_SIP\_ALLOCATOR\_create，创建内存分配器

N0030: 成功返回 TRUE，失败返回 FALSE，用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.4.2 PD-F-PPLAT-SIP-ALLOCATOR-0020 销毁内存分配器

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为

SR-F-PSS-SIP-PARSER-0090 编解码支持内存分配器

功能概述: SIP 平台各模块可以销毁内存分配器

前置条件: C1

C0010 销毁内存分配器

后置条件:

R0010 成功销毁内存分配器。

正常过程: N1 销毁内存分配器

N0010: 收到销毁内存分配器，用例开始。

N0020: 调用 P\_SIP\_ALLOCATOR\_destroy，销毁内存分配器。

N0030: 成功返回 TRUE，失败返回 FALSE，用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.4.3 PD-F-PPLAT-SIP-ALLOCATOR-0030 向内存分配器申请内存

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为

SR-F-PSS-SIP-PARSER-0090 编解码支持内存分配器

功能概述: SIP 平台各模块可以向内存分配器申请内存

前置条件: C1

C0010 向内存分配器申请内存

后置条件:

R0010 成功申请内存。

正常过程: N1 向内存分配器申请内存

N0010: SIP 平台各模块需要申请内存, 用例开始。

N0020: 调用 P\_SIP\_ALLOCATOR\_allocate, 向内存分配器申请内存。

N0030: 成功返回内存地址, 失败返回 NULL, 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.4.4 PD-F-PPLAT-SIP-ALLOCATOR-0040 向内存分配器释放内存

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为

SR-F-PSS-SIP-PARSER-0090 编解码支持内存分配器

功能概述: SIP 平台各模块可以向内存分配器释放内存

前置条件: C1

C0010 向内存分配器释放内存

后置条件:

R0010 成功释放内存。

正常过程: N1 向内存分配器释放内存

N0010: SIP 平台各模块需要释放已从内存分配器申请的内存, 用例开始。

N0020: 调用 P\_SIP\_ALLOCATOR\_delallocate, 向内存分配器释放内存。

N0030: 成功返回 TRUE, 失败返回 FALSE, 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

### 8.1.5 OMM 处理

#### 8.1.5.1 PD-F-PPLAT-SIP-OMM-0010 TP 管理进程收到设置信令跟踪请求

功能概述: 处理信令跟踪设置请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TP 管理进程收到设置信令跟踪请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 管理进程收到设置信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceSetReq, 在接口内进行跟踪任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.2 PD-F-PPLAT-SIP-OMM-0020 TP 处理进程收到设置信令跟踪请求

功能概述: 处理信令跟踪设置请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TP 处理进程收到设置信令跟踪请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 处理进程收到设置信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceSetReq, 在接口内进行跟踪任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.3 PD-F-PPLAT-SIP-OMM-0030 TU 管理进程收到设置信令跟踪请求

功能概述: 处理信令跟踪设置请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TU 管理进程收到设置信令跟踪请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 管理进程收到设置信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceSetReq, 在接口内进行跟踪任务维护 and 应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.4 PD-F-PPLAT-SIP-OMM-0040 TU 处理进程收到设置信令跟踪请求

功能概述: 处理信令跟踪设置请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TPU 处理进程收到设置信令跟踪请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 处理进程收到设置信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceSetReq, 在接口内进行跟踪任务维护 and 应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.5 PD-F-PPLAT-SIP-OMM-0050 TP 管理进程收到取消信令跟踪请求

功能概述: 处理信令跟踪取消请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TP 管理进程收到取消信令跟踪请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 管理进程收到取消信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceCancelReq, 在接口内进行跟踪任务  
维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.6 PD-F-PPLAT-SIP-OMM-0060 TP 处理进程收到取消信令跟踪请求

功能概述: 处理信令跟踪取消请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TP 处理进程收到取消信令跟踪请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 处理进程收到取消信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceCancelReq, 在接口内进行跟踪任务  
维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.7 PD-F-PPLAT-SIP-OMM-0070 TU 管理进程收到取消信令跟踪请求

功能概述: 处理信令跟踪取消请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TU 管理进程收到取消信令跟踪请求



后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 管理进程收到取消信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceCancelReq, 在接口内进行跟踪任务  
维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.8 PD-F-PPLAT-SIP-OMM-0080 TU 处理进程收到取消信令跟踪请求

**功能概述:** 处理信令跟踪取消请求, 维护跟踪任务列表, 并回送应答消息

**前置条件:** C1

C0010 TU 处理进程收到取消信令跟踪请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 处理进程收到取消信令跟踪请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleSigTraceCancelReq, 在接口内进行跟踪任务  
维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.9 PD-F-PPLAT-SIP-OMM-0090 TP 管理进程收到设置失败观察请求

**功能概述:** 处理失败观察设置请求, 维护观察任务列表, 并回送应答消息

**前置条件:** C1

C0010 TP 管理进程收到设置失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 管理进程收到设置失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveSetReq, 在接口内进行观察任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.10 PD-F-PPLAT-SIP-OMM-0100 TP 处理进程收到设置失败观察请求

功能概述: 处理设置失败观察请求, 维护观察任务列表, 并回送应答消息

前置条件: C1

C0010 TP 处理进程收到设置失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 处理进程收到设置失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveSetReq, 在接口内进行观察任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.11 PD-F-PPLAT-SIP-OMM-0110 TU 管理进程收到设置失败观察请求

功能概述: 处理设置失败观察请求, 维护观察任务列表, 并回送应答消息

前置条件: C1

C0010 TU 管理进程收到设置失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 管理进程收到设置失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveSetReq, 在接口内进行观察任务  
维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.12 PD-F-PPLAT-SIP-OMM-0120 TU 处理进程收到设置失败观察请求

功能概述: 处理设置失败观察请求, 维护观察任务列表, 并回送应答消息

前置条件: C1

C0010 TU 处理进程收到设置失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 处理进程收设置失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveSetReq, 在接口内进行观察任务  
维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.13 PD-F-PPLAT-SIP-OMM-0130 TP 管理进程收到取消失败观察请求

功能概述: 处理取消失败观察请求请求, 维护跟踪任务列表, 并回送应答消息

前置条件: C1

C0010 TP 管理进程收到取消失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 管理进程收到取消失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveCancelReq, 在接口内进行跟踪任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.14 PD-F-PPLAT-SIP-OMM-0140 TP 处理进程收到取消失败观察请求

**功能概述:** 处理取消失败观察请求, 维护观察任务列表, 并回送应答消息

**前置条件:** C1

C0010 TP 处理进程收到取消失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TP 处理进程收到取消失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveCancelReq, 在接口内进行观察任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.15 PD-F-PPLAT-SIP-OMM-0150 TU 管理进程收到取消失败观察请求

**功能概述:** 处理取消失败观察请求, 维护观察任务列表, 并回送应答消息

**前置条件:** C1

C0010 TU 管理进程收到取消失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 管理进程收到取消失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveCancelReq, 在接口内进行观察任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

#### 8.1.5.16 PD-F-PPLAT-SIP-OMM-0160 TU 处理进程收到取消失败观察请求

功能概述: 处理取消失败观察请求, 维护观察任务列表, 并回送应答消息

前置条件: C1

C0010 TU 处理进程收到取消失败观察请求

后置条件:

R0010 回送应答消息

正常过程:

N0010: TU 处理进程收到取消失败观察请求, 用例开始。

N0020: 调用 PPLAT\_SIP\_OMM\_handleFailObserveCancelReq, 在接口内进行观察任务维护和应答消息发送

N0030: 用例结束。

可选过程: 无

异常过程: 无

包含: 无

特殊要求: 无

## 8.2 非功能协作

参见《NGN PPLAT V1.00.10\_系统方案》

## 9 组件说明

### 9.1 TU 层处理

### 9.1.1 功能需求

#### 9.1.1.1 SD-F-PPLAT-SIP-TU-0010 UA 模式下 1xx 响应的重发

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-PARSER-0010 支持扩展头部的编解码

##### 功能概述：

在 UA 模式下，TU 根据业务指示负责 1xx 响应的重发。对于 1xx 在收到对应的 PRACK 之前，TU 根据协议 RFC3262 要求重发 1xx 响应。

##### 前置条件：C1

C0001 SIP 协议栈为 UA 模式

C0002 上层应用发送 1xx 响应

C0003 消息属性置为需要重传

##### 后置条件：

R0001 UAS 重传 1xx

##### 正常过程：

N0010 上层应用指示 TU 层发送需要重传的 1xx 响应

N0020 TU 层进行正常处理后，设置重传定时器 T1 和重传保护定时器 64\*T1，缓存 1xx 响应。

N0030 TU 将 1xx 响应消息交给 TR 层处理。

N0040 重传定时器超时，TU 重发 1xx 响应；重设重传定时器为上次定时器时长 2 倍。

N0060 TU 收到 TR 层上传的 PRACK 消息，该消息与缓存的 1xx 匹配

N0070 TU 杀掉重传定时器和重传保护定时器，释放缓冲的 1xx 响应

N0080 TU 将 PRACK 上报给应用

N0090 用例结束

##### 可选过程：A1 UAS 方式下对于 1xx 的重传保护定时器超时处理

A0010 上层应用指示 TU 层发送需要重传的 1xx 响应

A0020 TU 层进行正常处理后，设置重传定时器 T1 和重传保护定时器 64\*T1，缓存 1xx 响应。

A0030 TU 将 1xx 响应消息交给 TR 层处理。

A0040 重传定时器超时，TU 重发 1xx 响应；重设重传定时器为上次定时器时长 2 倍。

A0050 重传保护定时器超时，TU 切换到待释放状态，同时通知上层应用释放呼叫

A0060 用例结束

##### 异常过程：E1 并发可靠临时响应

前置条件:

C0010 SIP TU 发出了需要重传的临时响应未收到对应 PRACK

00020 上层应用指示发送新的需要重传的临时响应

后置条件:

R0010 新的需要重传的临时响应被丢弃, 同时向上层应用报错

N0010 TU 从上层应用收到需要重传的 1xx 响应;

N0020 TU 当前 LEG 里面已经缓存了一个等待应答 1xx 响应, TU 丢弃新的 1xx 响应;  
使用 EV\_PPLAT\_SIP\_API\_ABORT\_IND 上报上层应用

N0030 用例结束。

包含: 无

特殊需求:

DB 要求: 获取 T1/T2 定时器配置接口

#### 9.1.1.2 SD-F-PPLAT-SIP-TU-0020 UA 模式下 2xx 响应的重发

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为  
SR-F-PSS-SIP-PARSER-0010 支持扩展头部的编解码

功能概述:

在 UA 模式下, TU 根据业务指示负责 2xx 响应的重传, 在收到 ACK 之前, TU 根据 RFC3261 协议重发 2xx。

前置条件: C1

C0001 SIP 协议栈为 UA 模式

C0002 上层应用发送 2xx 响应

C0003 消息属性置为需要重传

后置条件:

R0001 UAS 负责 2xx 重传

正常过程:

N0010 上层应用指示 TU 层发送需要重传的 2xx 响应

N0020 TU 层进行正常处理后, 设置重传定时器 T1 和重传保护定时器 T2, 缓存 1xx 响应。

N0030 TU 将 2xx 响应消息交给 TR 层处理。

N0040 重传定时器超时, TU 重发 2xx 响应; 重设重传定时器为上次定时器时长 2 倍。

N0050 重传保护定时器超时, TU 切换到待释放状态, 同时通知上层应用释放呼叫

N0060 用例结束

可选过程: A1 UAS 方式下对于 2xx 的重传保护定时器超时处理

A0010 上层应用指示 TU 层发送需要重传的 2xx 响应

A0020 TU 层进行正常处理后, 设置重传定时器 T1 和重传保护定时器 T2, 缓存 2xx 响应。

A0030 TU 将 2xx 响应消息交给 TR 层处理。

A0040 重传定时器超时, TU 重发 2xx 响应; 重设重传定时器为上次定时器时长 2 倍。

A0050 重传保护定时器超时, TU 切换到待释放状态, 同时通知上层应用释放呼叫

A0060 用例结束

异常过程: 无

包含: 无

特殊需求:

DB 要求: 获取 T1/T2 定时器配置接口

#### 9.1.1.3 SD-F-PPLAT-SIP-TU-0030 UA 模式下重发 2xx 响应的吸收

注: 对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-PARSER-0010 支持扩展头部的编解码

功能描述: 在 UA 模式下, SIP 协议栈发送 INVITE 请求后, 在收到重传的 2xx 响应时, 根据配置决定是否吸收重发的响应。

前置条件:

C0010 UA 模式下, SIP 协议栈收到重传的 2xx 响应,

C0020 SIP 协议栈被配置为吸收重传的 2xx 响应

后置条件:

R0010 SIP 协议栈吸收重传的 2xx 响应, 不将重传消息上报业务。

正常过程:

N0010 SIP 传输层收到 2xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

N0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

N0030 SIP TU 检查后确认为重传的 2xx, 读取配置为吸收。不将消息发送至上层应用

N0040 用例结束。

可选过程: A1 不吸收重传 2xx

前置条件:

C0010 在 UA 模式下, SIP 协议栈收到重传的 2xx 响应,

C0020 SIP 协议栈被配置为不吸收重传 2xx 响应

后置条件:



R0010 SIP 协议站不吸收重传的 2xx 响应，将重传消息上报业务。

N0010 SIP 传输层收到 2xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

N0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

N0030 SIP TU 检查后确认为重传的 2xx，读取配置为不吸收。发送响应指示 EV\_PPLAT\_SIP\_API\_ESTABLISH\_IND 到上层应用。

N0040 用例结束。

异常过程：无

包含：无

特殊要求：

DB 要求：获取重传响应吸收配置接口

#### 9.1.1.4 SD-F-PPLAT-SIP-TU-0040 UA 模式下重发 1xx 响应的吸收

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP-PARSER-0010 支持扩展头部的编解码

功能描述：在 UA 模式下，SIP 协议栈发送 INVITE 请求后，在收到重传的 1xx 响应时，根据配置决定是否吸收重发的响应。

前置条件：

C0010 UA 模式下，SIP 协议栈收到重传的 1xx 响应，

C0020 SIP 协议栈被配置为吸收重传的 1xx 响应

后置条件：

R0010 SIP 协议站吸收重传的 1xx 响应，不将重传消息上报业务。

正常过程：

N0010 SIP 传输层收到 1xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层

N0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。

N0030 SIP TU 通过 RSeq 匹配后确认为重传的 1xx，读取配置为吸收。不将消息发送至上层应用

N0040 用例结束。

可选过程：A1 不吸收重传 1xx

前置条件：

C0010 UA 模式下，SIP 协议栈收到重传的 1xx 响应

C0020 SIP 协议栈被配置为不吸收重传的 1xx 和 2xx 响应

后置条件：

R0010 SIP 协议站不吸收重传的 1xx 响应，将重传消息上报业务。

- N0010 SIP 传输层收到 1xx 响应通过 EV\_PPLAT\_SIP\_TPUI\_IND 消息发送到事务层
- N0020 事务层在匹配事务实例后通过 EV\_PPLAT\_SIP\_TRUI\_SYNC\_IND 消息发送到 SIP TU。
- N0030 SIP TU 通过 RSeq 匹配后确认为重传的 1xx，读取配置为不吸收。发送响应指示 EV\_PPLAT\_SIP\_API\_RESPONSE\_IND 到上层应用
- N0040 用例结束。

异常过程：无

包含：无

特殊要求：

DB 要求：获取重传响应吸收配置接口

#### 9.1.1.5 SD-F-PPLAT-SIP-TU-0050 TU 管理进程收到下行初始请求

参见 8.1.1 TU 层处理

#### 9.1.1.6 SD-F-PPLAT-SIP-TU-0060 TU 管理进程收到下行 Forking 初始请求

参见 8.1.1 TU 层处理

#### 9.1.1.7 SD-F-PPLAT-SIP-TU-0070 TU 管理进程收到上行消息

参见 8.1.1 TU 层处理

#### 9.1.1.8 SD-F-PPLAT-SIP-TU-0080 创建编解码实例

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为 SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

功能描述：创建编解码实例

前置条件：

C0010 SIP TR/TU 进程收到上电事件

后置条件：

R0010 创建编解码实例成功

正常过程：

- N0010 SIP TR/TU 进程收到上电事件；
- N0020 SIP TR/TU 调用编解码实例创建接口；
- N0030 编解码实例创建接口返回成功；
- N0040 用例结束。

异常过程：E1 编解码实例创建失败

- E0010 SIP TR/TU 进程收到上电事件；
- E0020 SIP TR/TU 调用编解码实例创建接口；

E0030 编解码实例创建接口返回失败;

E0040 SIP TU 发送告警消息到网管;

E0050 用例结束

包含: 无

特殊要求:

### 9.1.2 资源限制需求

无

### 9.1.3 其他需求

无

## 9.2 TP 层处理

### 9.2.1 功能需求

#### 9.2.1.1 SD-F-PPLAT-SIP-TP-0010 TP 管理进程收到下行消息 EV\_PPLAT\_SIP\_TPUI\_REQ

参见 8.1.2 TP 层处理

#### 9.2.1.2 SD-F-PPLAT-SIP-TP-0020 TP 管理进程收到下行消息 EV\_PPLAT\_SIP\_MSG\_REQ

参见 8.1.2 TP 层处理

#### 9.2.1.3 SD-F-PPLAT-SIP-TP-0030 TP 管理进程收到上行消息

参见 8.1.2 TP 层处理

#### 9.2.1.4 SD-F-PPLAT-SIP-TP-0040 TP 处理进程收到消息 EV\_PPLAT\_SIP\_MSG\_IND

参见 8.1.2 TP 层处理

#### 9.2.1.5 SD-F-PPLAT-SIP-TP-0050 TP 管理进程的链路维护

TP 管理进程的链路维护与 TP 处理进程的链路维护一样, 详见《NGN PPLAT V1.00.10\_系统方案》

#### 9.2.1.6 SD-F-PPLAT-SIP-TP-0060 TP 管理进程的 SOCKET 维护

TP 管理进程的 SOCKET 维护与 TP 处理进程的 SOCKET 维护一样,  
详见《NGN PPLAT V1.00.10\_系统方案》

#### 9.2.1.7 SD-F-PPLAT-SIP-TP-0070 TP 处理进程的 IPSec 相关处理

TP 管理进程的 IPSec 处理与 TP 处理进程的 IPSec 处理一样,

详见《NGN PPLAT V1.00.10\_系统方案》

#### 9.2.1.8 SD-F-PPLAT-SIP-TP-0010 创建编解码实例

注：对应需求文档《NGN PPLAT SIP V1.01.10\_软件需求说明书》中的条目为  
SR-F-PSS-SIP -0010 SIP 平台协议栈支持 ATCA

功能描述：创建编解码实例

前置条件：

C0010 SIP TP 进程收到上电事件

后置条件：

R0010 创建编解码实例成功

正常过程：

N0010 SIP TP 进程收到上电事件；

N0020 SIP TP 调用编解码实例创建接口；

N0030 编解码实例创建接口返回成功；

N0040 用例结束。

异常过程：E1 编解码实例创建失败

E0010 SIP TP 进程收到上电事件；

E0020 SIP TP 调用编解码实例创建接口；

E0030 编解码实例创建接口返回失败；

E0040 SIP TU 发送告警消息到网管；

E0050 用例结束

包含：无

特殊要求：

#### 9.2.2 资源限制需求

无

#### 9.2.3 其他需求

无

### 10 接口说明

接口变更说明

#### 10.1 用户接口

##### 10.1.1 SIP 编解码接口

##### 10.1.2 用户接口消息

## 10.2 数据库接口

## 10.2.1 SD-I-PPLAT-SIP-DBA-0010 分发接口

## 10.2.1.1 PPLAT\_SIP\_TPMNG\_DB\_getTpIDForUpMsg

函数名称		PPLAT_SIP_TPMNG_DB_getTpIDForUpMsg	类型（函数/宏）	函数
功能详细说明		SIP TP 管理进程在收到上行消息 EV_PPLAT_SIP_MSG_IND 后选择 SIP TP 处理进程接口		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_TPMNG_DB_getTpIDForUpMsg(IN PPLAT_SIP_GETTPIDFORUPMSG_IN_T * ptMsgInd,IN PPLAT_SIP_PARSER_CONFIG_T *ptConfig,IN OUT LP_PPLAT_SIP_APPID_T * ptDstId);				
序号	参数名	返回值说明		
		PPLAT_SIP_SUCCESS 成功; 其他 失败; 定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptMsgInd	底层接入模块发送给 SIP TP 进程的 SIP 消息结构, 结构定义 SIP 平台内部接口		
2	ptConfig	解码配置结构, 定义见《NGN PPLAT V1.01.10_接口说明书.doc》		
序号	参数名	输出参数说明		
1	ptDstId	SIP TP 处理进程标识		
算法及流程说明				
<pre>typedef struct tagPPLAT_SIP_GETTPIDFORUPMSG_T {     PPLAT_SIP_IPADDR_T  tSrcIpPort; /* 源地址 */     PPLAT_SIP_IPADDR_T  tDstIpPort; /* 目的地址 */     BYTE                 bProtocol; /* 承载协议类型 */     PPLAT_SIP_IPSEC_HANDLE_T tHandle; /* IPSecHandle */     PPLAT_SIP_EXTERNAL_INFO_T tExternalInfo;     WORD16                wMsgBufLen;     BYTE*                pbMsg; }PPLAT_SIP_GETTPIDFORUPMSG_T  typedef struct tagPPLAT_SIP_APPID_T {     #if(PPALT_PLATFORM_TYPE == PPALT_PLATFORM_TYPE_ZX3G)         WORD16 wModule; /* 模块号*/     #elseif(PPALT_PLATFORM_TYPE == PPALT_PLATFORM_TYPE_ATCA)         PPLAT_SIP_PID tPID; /* 目的进程标识*/     #else     #endif }PPLAT_SIP_PACK_PPLAT_SIP_APPID_T,LP_PPLAT_SIP_APPID_T;</pre>				

其他限制或说明

## 10.2.1.2 PPLAT\_SIP\_TPMNG\_DB\_getTpIDForDownMsg

函数名称		PPLAT_SIP_TPMNG_DB_getTpIDForDownMsg	类型（函数/宏）	函数
功能详细说明		SIP TP 管理进程在收到下行消息 EV_PPLAT_SIP_TPUI_REQ 后选择 SIP TP 处理进程接口		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_TPMNG_DB_getTpIDForDownMsg(IN PPLAT_SIP_TPUI_REQ_T *ptTransReq,IN PPLAT_SIP_PARSER_CONFIG_T *ptConfig,IN OUT LP_PPLAT_SIP_APPID_T * ptDstId);				
序号	参数名	返回值说明		
		PPLAT_SIP_SUCCESS 成功； 其他 失败； 定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptTransReq	TR 曾下发给 TP 的请求消息指针，结构定义见《NGN PPLAT V1.00.10_系统方案》		
2	ptConfig	解码配置结构，定义见《NGN PPLAT V1.01.10_接口说明书.doc》		
序号	参数名	输出参数说明		
1	ptDstId	SIP TP 处理进程标识		
算法及流程说明				
其他限制或说明				

## 10.2.1.3 PPLAT\_SIP\_TUMNG\_DB\_getTuIDForUpMsg

函数名称		PPLAT_SIP_TUMNG_DB_getTuIDForUpMsg	类型（函数/宏）	函数
功能详细说明		SIP TR/TU 管理进程在收到上行消息 EV_PPLAT_SIP_TPUI_IND 后选择 SIP TR/TU 处理进程接口		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_TUMNG_DB_getTuIDForUpMsg(IN PPLAT_SIP_TPUI_IND_T * ptInd,IN PPLAT_SIP_PARSER_CONFIG_T *ptConfig,IN OUT LP_PPLAT_SIP_APPID_T * ptDstId);				
序号	参数名	返回值说明		

		PPLAT_SIP_SUCCESS 成功; 其他 失败; 定义见公共宏值定义部分
序号	参数名	输入参数说明
1	ptInd	TP 发送给 TR 的上行消息结构指针, 结构定义见《NGN PPLAT V1.00.10_系统方案》
2	ptConfig	解码配置结构, 定义见《NGN PPLAT V1.01.10_接口说明书.doc》
序号	参数名	输出参数说明
1	ptDstId	SIP TR/TU 处理进程标识
算法及流程说明		
其他限制或说明		

## 10.2.1.4 PPLAT\_SIP\_TUMNG\_DB\_getTuIPForDownMsg

函数名称		PPLAT_SIP_TUMNG_DB_getTuIDForDownMsg	类型（函数/宏）	函数
功能详细说明		SIP TR/TU 管理进程在收到下行消息 EV_PPLAT_SIP_API_INIT_REQUEST_REQ 后选择 SIP TR/TU 处理进程接口		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_TUMNG_DB_getTuIDForDownMsg(IN PPLAT_SIP_INIT_REQUEST_REQ_T * ptReq,IN PPLAT_SIP_PARSER_CONFIG_T *ptConfig,IN OUT LP_PPLAT_SIP_APPID_T * ptDstId);				
序号	参数名	返回值说明		
		PPLAT_SIP_SUCCESS 成功； 其他 失败； 定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptReq	上层应用发送给 TU 的初始请求消息结构指针，结构定义见《NGN PPLAT V1.00.10_接口说明书.doc》		
2	ptConfig	解码配置结构，定义见《NGN PPLAT V1.01.10_接口说明书.doc》		
序号	参数名	输出参数说明		
1	ptDstId	SIP TR/TU 处理进程标识		
算法及流程说明				
其他限制或说明				

## 10.2.1.5 PPLAT\_SIP\_TUMNG\_DB\_getTuIDForForkingDownMsg

函数名称	PPLAT_SIP_TUMNG_DB_getTuIDForForkingDownMsg	类型 (函数/宏)	函数
功能详细说明	SIP TR/TU 管理进程在收到下行消息 EV_PPLAT_SIP_API_FORKING_REQ 后选择 SIP TR/TU 处理进程接口		

性能		
序号	分类	使用的全局及静态变量
1	全局	无
2	静态	无
资源限制		
函数声明		
WORD32 PPLAT_SIP_TUMNG_DB_getTuIDForForkingDownMsg(IN PPLAT_SIP_FORKING_REQ_T * ptReq, IN PPLAT_SIP_PARSER_CONFIG_T * ptConfig, IN OUT LP PPLAT_SIP_APPID_T * ptDstId);		
序号	参数名	返回值说明
		PPLAT_SIP_SUCCESS 成功; 其他 失败; 定义见公共宏值定义部分
序号	参数名	输入参数说明
1	ptReq	上层应用发送给 TU 的初始 Forking 请求消息结构指针, 结构定义见《NGN PPLAT V1.01.10_接口说明书.doc》
2	ptConfig	解码配置结构, 定义见《NGN PPLAT V1.01.10_接口说明书.doc》
序号	参数名	输出参数说明
1	ptDstId	SIP TR/TU 处理进程标识
算法及流程说明		
其他限制或说明		

#### 10.2.1.6 PPLAT\_SIP\_AppDistribution

函数名称		PPLAT_SIP_AppDistribution	类型（函数/宏）	函数
功能详细说明		SIP TR/TU 处理进程在上报初始请求消息时用于获取上层应用进程接口，该函数 SIP TU 向上层应用发送 EV_PPLAT_SIP_API_INIT_REQUEST_IND 时调用		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_AppDistribution(IN PPLAT_SIP_INIT_REQUEST_IND_T * ptInitReqInd,IN PPLAT_SIP_PARSER_CONFIG_T *ptConfig,IN OUT LP PPLAT_SIP_APPID_T * ptDstId);				
序号	参数名	返回值说明		
		PPLAT_SIP_SUCCESS 成功； 其他 失败； 定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptInitReqInd	SIP TU 发送给上层应用的初始请求消息结构指针，结构定义见《NGN PPLAT V1.01.10_接口说明书.doc》		
2	ptConfig	解码配置结构，定义见《NGN PPLAT V1.01.10_接口说明书.doc》		
序号	参数名	输出参数说明		
1	ptDstId	上层应用进程标识,定义参见 10.2.1.1		



算法及流程说明
其他限制或说明

## 10.2.1.7 PPLAT\_SIP\_DB\_getUpTRModuleNo

函数名称		PPLAT_SIP_DB_getUpTRModuleNo	类型（函数/宏）	函数
功能详细说明		SIP 协议栈封装的用于获取上行的 TR 分发模块接口,该函数由 TP 层在向 TR 发 IND 的时候确定 TR 所在模块模块号		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_DB_getUpTRModuleNo(IN PPLAT_SIP_TPUI_IND_T * ptTransInd,IN PPLAT_SIP_PARSER_CONFIG_T *ptConfig,IN OUT LP_PPLAT_SIP_APPID_T * ptDstId);				
序号	参数名	返回值说明		
		PPLAT_SIP_SUCCESS 成功; 其他 失败; 定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptTransInd	TP 发送给 TR 的 IND 消息指针,结构定义见 SIP 平台内部接口部分		
2	ptConfig	解码配置结构,定义见《NGN PPLAT V1.00.10_接口说明书.doc》		
序号	参数名	输出参数说明		
1	ptDstId	上行 TR 层处理进程标识,定义参见 10.2.1.1		
算法及流程说明				
其他限制或说明				

## 10.2.1.8 PPLAT\_SIP\_DB\_getDownModuleNo

函数名称		PPLAT_SIP_DB_getDownModuleNo	类型（函数/宏）	函数
功能详细说明		SIP 协议栈封装的用于获取下行的 TP 分发模块 和本端地址接口，该函数由 TR 层在向 TP 下发请求的时候确定 TP 所在模块模块号		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				

WORD32 PPLAT_SIP_DB_getDownModuleNo(IN PPLAT_SIP_TPUI_REQ_T* ptTransReq, IN PPLAT_SIP_PARSER_CONFIG_T *ptConfig, OUT LP_PPLAT_SIP_APPID_T * ptDstId, OUT PPLAT_SIP_IPADDR_T* ptLocalIpAddr);		
序号	参数名	返回值说明
1		PPLAT_SIP_SUCCESS 成功; 其他 失败; 定义见公共宏值定义部分
序号	参数名	输入参数说明
1	ptTransReq	TR 曾下发给 TP 的请求消息指针, 结构定义见 SIP 平台内部接口
2	ptConfig	解码配置结构, 定义见《NGN PPLAT V1.00.10_接口说明书.doc》
序号	参数名	输出参数说明
1	ptDstId	下行 TP 层处理进程标识, 定义参见 10.2.1.1
2	ptLocalIpAddr	本端 IP 地址端口, 定义见定义见《NGN PPLAT V1.00.10_接口说明书.doc》, IP 地址为网络序, 端口为主机序
3		
算法及流程说明		
其他限制或说明		
蓝色部分结构定义由 TR 层定义后替换		

## 10.2.2SD-I-PPLAT-SIP-DBA-0020 配置接口

### 10.2.2.1 PPLAT\_SIP\_DB\_getSipStackAttrCfg

函数名称	PPLAT_SIP_DB_getSipStackAttrCfg	类型（函数/宏）	函数
功能详细说明	获取 SIP 协议栈属性配置		
性能			
序号	分类	使用的全局及静态变量	
1	全局	无	
2	静态	无	
资源限制			
函数声明			
WORD32 PPLAT_SIP_DB_getSipStackAttrCfg(IN LP_PPLAT_SIP_DB_STACKATTR_CFG_T ptStackAttrCfg);			
序号	参数名	返回值说明	
1		PPLAT_SIP_SUCCESS -- 成功 其他 -- 失败 宏值定义见公共宏值定义部分	
序号	参数名	输入参数说明	
1			
2			
序号	参数名	输出参数说明	
1	ptStackAttrCfg	协议栈属性配置	
算法及流程说明			

```
typedef struct tagPPLAT_SIP_DB_STACKATTR_CFG_T
{
    #define PPLAT_SIP_UA_MODE 0 /* UA */
    #define PPLAT_SIP_PROXY_MODE 1 /* PROXY */

    BYTE bStackMode; /* UA/PROXY */

    #define PPLAT_SIP_CALLSTATEFUL 1 /* 有状态 CALL */
    #define PPLAT_SIP_TRANSSTATEFUL 2 /* 有状态事务 */
    #define PPLAT_SIP_STATELESS 3 /* 无状态 */

    BYTE bStackState; /* 协议栈状态 */

    #define PPLAT_SIP_VERSIONLEN_MAX 20 /* 最大版本长度 */

    BYTE baVersionStr[PPLAT_SIP_VERSIONLEN_MAX]; /* 协议栈版本 */

    /* SIP 协议栈作为 PROXY 时, 是否对 BYE 的 200OK, CANCEL 引起的 487 做一端到另一端的旁路处理 */
    BOOL8 bINeedByPass;
    /* SIP 协议栈在往码流中添加 Record-Route 的时是使用 IP 地址还是主机名 */
    BYTE bRecordRouteAddType;

    WORD32 dwCallWaitReleaseTimer; /* 呼叫等待释放定时器时长, 缺省 32s, 单位 ms, 具体含义见限制说明 */
    WORD32 dwSubWaitReleaseTimer; /* 订阅等待释放定时器时长, 缺省 32s, 单位 ms, 具体含义见限制说明 */
    /*路由重选配置*/
    BOOL8 bIIsNeedReselect; /* 是否需要路由重选 */
    WORD32 dwReselectInterval; /* 路由重选间隔 */
    /* FORK 配置 */
    #define PPLAT_SIP_FORK_PAR 1 /* 并行 */
    #define PPLAT_SIP_FORK_SEQ 2 /* 串行 */
    #define PPLAT_SIP_FORK_BOTH 3 /* 串并行都支持 */
    #define PPLAT_SIP_SIP_FORK_FORBID 4 /* 禁止 */
    BYTE bForkType;
    WORD32 dwCallSEQTime; /* 串行情况下呼叫流程串行呼叫时间间隔, 单位 ms, 缺省值 36s */
    WORD32 dwNonCallSEQTime; /* 串行情况下非呼叫流程串行呼叫时间间隔, 单位 ms, 缺省值 6s */
    BOOL8 bIIsCheckDialogID; /* 是否对对话内消息进行 DialogID 一致性检查,缺省不做检查 */
    /* 目前只支持一种算法 */
    #define PPLAT_SIP_SIGCMP_METHOD_LZ77 1 /* LZ77 算法 */
    BYTE bSigCmpMethod; /* 信令压缩算 */
    BOOL8 bIIsNeedPassResend1xx; /* UA 模式下是否上报重传 1xx */
    BOOL8 bIIsNeedPassResend2xx; /* UA 模式下是否上报重传 2xx */
}PPLAT_SIP_PACK
PPLAT_SIP_DB_STACKATTR_CFG_T,*LP_PPLAT_SIP_DB_STACKATTR_CFG_T;
```

其他限制或说明

bStackMode 是至 SIP 协议栈的缺省角色，在配置为 UA 时，SIP 协议栈只能充当 UA 角色。在配置为 PROXY 时，对于某个业务流程，上层应用可以通过 SIP 平台提供的接口修改协议栈在该流程中的角色。

等待释放定时器描述的是 SIP TU 层维护的 Dialog 从收到释放指示，到真正释放对应的数据区实例的时长，例如在收到 BYE 后，在等待释放定时器时间内收到的 200 OK 能够匹配到 Dialog，而在这之后则匹配不到。

路由重选是指 SIP 协议栈在使用其初始请求中的地址标识进行路由分析时可能得到多个下一跳网元的 IP 地址端口对，在使用其中一个地址发送初始请求后，重选时间内没有收到响应，则使用其他地址重发请求的功能

### 10.2.2.2 PPLAT\_SIP\_TPMNG\_DB\_getSipIpPortCfg

函数名称		PPLAT_SIP_TPMNG_DB_getSipIpPortCfg	类型（函数/宏）	函数
功能详细说明		获取指定 TP 管理进程的 IP 地址及端口对配置，TP 层在上电的时候使用		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_TPMNG_DB_getSipIpPortCfg(IN LP_PPLAT_SIP_APPID_T ptTpId, OUT LP_PPLAT_SIP_DB_IPPORT_CFG_T ptCfg);				
序号	参数名	返回值说明		
1		PPLAT_SIP_SUCCESS -- 成功 其他 -- 失败 宏值定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptTpId	TP 所在地标识，定义参见 10.2.1.1		
2				
序号	参数名	输出参数说明		
1	ptCfg	SIP IP 地址及端口配置，结构定义参见《NGN PPLAT V1.00.10_系统方案》		
算法及流程说明				
其他限制或说明				

### 10.2.2.3 PPLAT\_SIP\_TPMNG\_DB\_getAllSipLinkCfg

函数名称		PPLAT_SIP_DB_getAllSipLinkCfg	类型（函数/宏）	函数
功能详细说明		获取本 TP 管理进程的所有的链路号列表，TP 层在上电的时候使用		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				

函数声明		
WORD32 PPLAT_SIP_TPMNG_DB_getAllSipLinkCfg(OUT LP PPLAT_SIP_DB_LINKNOLST_CFG_T ptCfg);		
序号	参数名	返回值说明
1		PPLAT_SIP_SUCCESS -- 成功 其他 -- 失败 宏值定义见公共宏值定义部分
序号	参数名	输入参数说明
1		
2		
序号	参数名	输出参数说明
1	ptCfg	SIP 链路号列表, 结构定义参见《NGN PPLAT V1.00.10_系统方案》
算法及流程说明		
其他限制或说明		

#### 10.2.2.4 PPLAT\_SIP\_DB\_getLocalIPByDstIp

函数名称		PPLAT_SIP_DB_getLocalIPByDstIp	类型（函数/宏）	函数
功能详细说明		SIP 协议栈封装的用于获取下行的 TP 分发模块号、实例号和本端 IP-PORT，该函数由 TU 获取本端 IP-PORT 时使用，并获得相应的实例号及模块号		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_DB_getLocalIPByDstIp(IN T_IPADDR * ptDstIpAddr,IN BYTE bProtocol,OUT T_IPADDR* ptLocalIpAddr, INOUT PPLAT_SIP_EXTERNAL_INFO_T *ptExternalInfo );				
序号	参数名	返回值说明		
1		PPLAT_SIP_SUCCESS 成功； 其他 失败； 定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptDstIpAddr	目的 IP 地址端口，定义见《NGN PPLAT V1.00.10_接口说明书.doc》，IP 地址为网络序，端口为主机序		
2	bProtocol	协议类型		
3	ptExternalInfo	ptExternalInfo 中的本端 IP 地址作为入参，PORT 无效 /* 修订记录编号 0004 */		
序号	参数名	输出参数说明		
1	ptLocalIpAddr	本地 IP 地址端口，定义见《NGN PPLAT V1.00.10_接口说明书.doc》，IP 地址为网络序，端口为主机序		
算法及流程说明				
其他限制或说明				

## 10.2.2.5 PPLAT\_SIP\_TP\_UpdateExternalInfo

注：修订记录编号 0005

函数名称		PPLAT_SIP_TP_UpdateExternalInfo	类型（函数/宏）	函数
功能详细说明		SIP 协议栈封装的用于 TP 收到 TPM 或者其他接入设备发送过来的异步消息，进行扩展信息的更新		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
WORD32 PPLAT_SIP_TP_UpdateExternalInfo (IN BYTE bProtocol,IN PPLAT_SIP_IPADDR_T *ptRemoteIp, IN PPLAT_SIP_IPADDR_T *ptLocalIp, INOUT PPLAT_SIP_EXTERNAL_INFO_T *ptSipExtInfo)				
序号	参数名	返回值说明		
1		PPLAT_SIP_SUCCESS 成功； 其他 失败； 定义见公共宏值定义部分		
序号	参数名	输入参数说明		
1	ptLocalIp	本地地址，TPM 或接入设备异步消息中所携带的本端地址定义见定义见《NGN PPLAT V1.00.10_接口说明书.doc》，IP 地址为网络序，端口为主机序		
2	ptRemotelp	远端地址，TPM 或接入设备异步消息中所携带的远端地址定义见定义见《NGN PPLAT V1.00.10_接口说明书.doc》，IP 地址为网络序，端口为主机序		
3	bProtocol	协议类型		
序号	参数名	输出参数说明		
1	ptExternalInfo	ptExternalInfo 扩展信息，将调用 DB 接口返回的 TP 实例的 IP-PORT 放到该扩展信息中		
算法及流程说明				
其他限制或说明				

## 10.2.2.6 PPLAT\_SIP\_DB\_GetSubAccessAddr

注：修订记录编号 0006

函数名称		PPLAT_SIP_DB_GetSubAccessAddr	类型（函数/宏）	函数
功能详细说明		SIP 协议栈封装的用于获取后续请求消息的接入地址信息（注：发送地址与后续接入地址可以不同）		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				

函数声明		
WORD32 PPLAT_SIP_DB_GetSubAccessAddr (IN PPLAT_SIP_IPADDR_T * ptLocalIpAddr, IN PPLAT_SIP_IPADDR_T * ptRemoteIpAddr, IN PPLAT_SIP_EXTERNAL_INFO_T *ptExternalInfo OUT PPLAT_SIP_IPADDR_T *pSubAccessAddr);		
序号	参数名	返回值说明
1		PPLAT_SIP_SUCCESS 成功; 其他 失败; 定义见公共宏值定义部分
序号	参数名	输入参数说明
1	ptLocalIpAddr	本地 IP 地址端口, 定义见《NGN PPLAT V1.00.10_接口说明书.doc》, IP 地址为网络序, 端口为主机序
2	ptRemotelpAddr	对端 IP 地址端口, 定义见《NGN PPLAT V1.00.10_接口说明书.doc》, IP 地址为网络序, 端口为主机序
3	ptExternalInfo	ptExternalInfo 中的本端 IP-PORT 地址作为入参
序号	参数名	输出参数说明
1	pSubAccessAddr	后续接入地址信息
2		
算法及流程说明		
其他限制或说明		

### 10.3 操作维护接口

#### 10.3.1 SD-I-PPLAT-SIP-OMC-0010 SIP 平台信令跟踪接口

##### 10.3.1.1 PPLAT\_SIP\_OMM\_handleSigTraceSetReq

函数名称		PPLAT_SIP_OMM_handleSigTraceSetReq	类型（函数/宏）	函数
功能详细说明		SIP 进程处理网管发送的信令跟踪任务设置请求函数。在该函数中根据网管设置跟踪任务消息维护信令跟踪任务列表，并回送应答消息。 SIP TP 处理进程、SIP TP 管理进程、SIP TU 处理进程或者 SIP TU 管理进程在收到 EV_PPLAT_SIP_OMM_SIGTRACE_SET_REQ 调用该函数		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
VOID PPLAT_SIP_OMM_handleSigTraceSetReq (VOID pvReq);				
序号	参数名	返回值说明		
1		无		
序号	参数名	输入参数说明		
1	pvReq	设置信令跟踪任务请求指针，其结构与事件 EV_PPLAT_SIP_OMM_SIGTRACE_SET_REQ 对应的消息结构一致。		
序号	参数名	输出参数说明		
1				

算法及流程说明
其他限制或说明

## 10.3.1.1.2 PPLAT\_SIP\_OMM\_handleSigTraceCancelReq

函数名称		PPLAT_SIP_OMM_handleSigTraceCancelReq	类型（函数/宏）	函数
功能详细说明		SIP 进程处理网管发送的信令跟踪任务取消请求函数。在该函数中根据网管取消跟踪任务消息维护信令跟踪任务列表，并回送应答消息。 SIP TP 处理进程、SIP TP 管理进程、SIP TU 处理进程或者 SIP TU 管理进程在收到 EV_PPLAT_SIP_OMM_SIGTRACE_CANCEL_REQ 调用该函数		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
VOID PPLAT_SIP_OMM_handleSigTraceCancelReq(VOID pvReq);				
序号	参数名	返回值说明		
1		无		
序号	参数名	输入参数说明		
1	pvReq	取消信令跟踪任务请求指针，其结构与事件 EV_PPLAT_SIP_OMM_SIGTRACE_CANCEL_REQ 对应的消息结构一致。		
序号	参数名	输出参数说明		
1				
算法及流程说明				
其他限制或说明				

## 10.3.2 SD-I-PPLAT-SIP-OMC-0020 SIP 平台失败观察接口

## 10.3.2.1.1 PPLAT\_SIP\_OMM\_handleFailObserveSetReq

函数名称		PPLAT_SIP_OMM_handleFailObserveSetReq	类型（函数/宏）	函数
功能详细说明		SIP 进程处理网管发送的失败观察任务设置请求函数。在该函数中根据网管设置失败观察任务消息维护失败观察任务列表，并回送应答消息。 SIP TP 处理进程、SIP TP 管理进程、SIP TU 处理进程或者 SIP TU 管理进程在收到 EV_PPLAT_SIP_OMM_FAILOBSERVE_SET_REQ 调用该函数		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				



VOID PPLAT_SIP_OMM_handleFailObserveSetReq(VOID pvReq);		
序号	参数名	返回值说明
1		无
序号	参数名	输入参数说明
1	pvReq	设置失败观察任务请求指针，其结构与事件EV_PPLAT_SIP_OMM_FAILOBSERVE_SET_REQ对应的消息结构一致。
序号	参数名	输出参数说明
1		
算法及流程说明		
其他限制或说明		

#### 10.3.2.1.2 PPLAT\_SIP\_OMM\_handleFailObserveCancelReq

函数名称		PPLAT_SIP_OMM_handleFailObserveCancelReq	类型（函数/宏）	函数
功能详细说明		SIP 进程处理网管发送的失败观察任务取消请求函数。在该函数中根据网管取消失败观察任务消息维护失败观察任务列表，并回送应答消息。 SIP TP 处理进程、SIP TP 管理进程、SIP TU 处理进程或者 SIP TU 管理进程在收到EV_PPLAT_SIP_OMM_FAILOBSERVE_CANCEL_REQ 调用该函数		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
VOID PPLAT_SIP_OMM_handleFailObserveCancelReq(VOID pvReq);				
序号	参数名	返回值说明		
1		无		
序号	参数名	输入参数说明		
1	pvReq	取消失败观察任务请求指针，其结构与事件EV_PPLAT_SIP_OMM_FAILOBSERVE_CANCEL_REQ 对应的消息结构一致。		
序号	参数名	输出参数说明		
1				
算法及流程说明				
其他限制或说明				

#### 10.3.3 SD-I-PPLAT-SIP-OMC-0030 SIP 人机命令接口

注：修订记录编号 0009

##### 10.3.3.1.1 PPLAT\_SIP\_OMM\_handleCommandReq

函数名称	PPLAT_SIP_OMM_handleCommandReq	类型（函数/宏）	函数
------	--------------------------------	----------	----

功能详细说明		SIP 进程处理网管发送的人机命令。在该函数中根据网管输入人机命令参数，实现相应得功能。 SIP OMM 管理进程收到网管 EV_CCM_MML_CCM_APP_REQ 调用该函数
性能		
序号	分类	使用的全局及静态变量
1	全局	无
2	静态	无
资源限制		
函数声明		
VOID PPLAT_SIP_OMM_handleCommandReq(VOID pvReq);		
序号	参数名	返回值说明
1		无
序号	参数名	输入参数说明
1	pvReq	收到网管人机命令请求参数,其结构与事件 EV_CCM_MML_CCM_APP_REQ 对应的消息结构一致。
序号	参数名	输出参数说明
1		
算法及流程说明		
其他限制或说明		

## 10.4 系统支撑接口

### 10.4.1 采用 ATCA 后与原来接口相比需要修改的支撑封装函数/接口列表

序号	函数/接口宏名称	类型（函数/宏）	功能描述以及与原接口的映射关系或变化内容
11	PPLAT_SIP_OSS_GetSelfPID	函数	对非ATCA来说，是对支撑函数 OSS_GetSelfPID 的封装映射，对 ATCA 来说，是对支撑函数 OSS_GetSelfJID 的封装映射
12	PPLAT_SIP_OSS_Sender	函数	对非ATCA来说，是对支撑函数 OSS_Sender 的封装映射，对 ATCA 来说，也是对 OSS_Sender 的封装映射，但该支撑接口函数的出参PID改为了JID
13	PPLAT_SIP_OSS_GetPIDByPNO	函数	对非ATCA来说，是对支撑函数 OSS_GetPIDByPNO 的封装映射，对 ATCA 来说，是对支撑函数 OSS_GetJIDByJNO 的封装映射
16	PPLAT_SIP_OSS_ConstructPNO	函数	对非ATCA来说，是对支撑函数 OSS_ConstructPNO 的封装映射，对 ATCA 来说，是对支撑函数 OSS_ConstructJNO 的封装映射
17	PPLAT_SIP_OSS_GetPIDByName	函数	对非ATCA来说，是对支撑函数 OSS_GetPIDByName 的封装映射，对 ATCA 来说，本接口不起作用，且没有替代函数，由于各JOB的类型都是需要对外暴露的，使用 OSS_GetJIDByJNO 可以完成相应的功能。
18	PPLAT_SIP_OSS_SendAsynMsg	函数	对非ATCA来说，是对支撑函数 OSS_SendAsynMsg 的封装映射，对

			ATCA来说，也是对OSS_SendAsynMsg的封装映射，但增加入参ucPriority，PID改为JID
24	PPLAT_SIP_SCS_WaitJob	函数	对非ATCA来说，本接口不起作用，对ATCA来说，是对系统控制函数SCS_WaitJob的封装映射

#### 10.4.2 采用 ATCA 后与原来接口相比需要修改的支撑封装函数/接口和参数说明

##### 10.4.2.1 SD-I-PPLAT-SIP-OSS-0010 采用 ATCA 后需要修改的 SIP 平台支撑调度管理接口

###### 10.4.2.1.1 PPLAT\_SIP\_OSS\_GetSelfPID

###### 10.4.2.1.1.1 接口说明

函数名称		PPLAT_SIP_OSS_GetSelfPID	类型（函数/宏）	函数
功能详细说明		SIP平台在非ATCA时，是对支撑函数OSS_GetSelfPID(PID *ptPid)的封装映射。获取当前调用进程的标识。该接口由应用程序调用触发，接口下层为调度管理系统。在ATCA时，是对支撑函数OSS_GetSelfJID(JID *ptJid)的封装映射。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_GetSelfPID(PID *ptPid)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_OSS_SUCCESS PPLAT_SIP_OSS_SCHE_ERR_PTR_NULL	成功 入参指针为空	
序号	参数名	输入参数说明		
1				
2				
3				
序号	参数名	输出参数说明		
1	ptPid	通过该出参将自己的 PID 返回		
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

###### 10.4.2.1.1.2 参数详细说明

参数名	类型	取值范围	解释
ptPid	PID*		在非ATCA下，与如下结构完全一致。 typedef struct tagPID { WORD32 dwPno; /* 进程号 */ };

			WORD32 dwDevId; /* 设备ID,里面包含区号, 局号信息, 高8位为局号, 次高16位为区号*/ WORD16 wModule; /* MP编号 */ WORD16 wUnit; /* 单板号 */ BYTE ucSUnit; /* 单板上的处理器号 */ BYTE ucSubSystem; /* 子系统号 */ BYTE ucRouteType; /* 路由类型, 区别业务/管理消息, 左右板位, 主备通信*/ BYTE ucExtendFlag; /* 字节序标志 */ }PID 在ATCA下, 实现对如下JID结构到PID结构的映射 typedef struct tagJID { WORD32 dwJno; /* JOB号 */ WORD32 dwDevId; /* 设备ID*/ WORD16 wModule; /* MP编号 */ WORD16 wUnit; /* 单板号 */ BYTE ucSUnit; /* 单板上的处理器号 */ BYTE ucSubSystem; /* 子系统号 */ BYTE ucRouteType; /* 路由类型, 区别业务/管理消息, 左右板位, 主备通信*/ BYTE ucExtendFlag; /* 扩展标志 */ }JID;

#### 10.4.2.1.2 PPLAT\_SIP\_OSS\_Sender

##### 10.4.2.1.2.1 接口说明

函数名称		PPLAT_SIP_OSS_Sender		类型（函数/宏）	函数
功能详细说明		SIP平台在非ATCA下，是对支撑函数OSS_Sender(PID *ptPid)的封装映射。获取调用进程当前消息的发方标识。ptPid中的ucRouteType字段为物理板位信息，如果进程需要通过这个接口向逻辑上的主备单板发送信息，则需要将ucRouteType修改为主备路由类型。该接口由应用程序调用触发，接口下层为调度管理系统。相关接口有OSS_GetFirstPID。在ATCA下是对支撑函数OSS_Sender(JID *ptJid)的封装映射。			
性能		N/A			
序号	分类	使用的全局及静态变量			
1	全局	无			
2	静态	无			
资源限制					
N/A					
函数声明					
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_Sender(PID *ptPid)					
序号	参数名	返回值说明			
1	无	PPLAT_SIP_OSS_SUCCESS 成功 PPLAT_SIP_OSS_SCHE_ERR_PTR_NULL 入参指针为空。			
序号	参数名	输入参数说明			
1					
2					
3					
序号	参数名	输出参数说明			
1	ptPid	通过该出参将调用进程当前消息的发方PID返回			
2					

3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.2.1.2.2 参数详细说明

参数名	类型	取值范围	解释
ptPid	PID*		<p>在非ATCA下，与如下结构完全一致：</p> <pre>typedef struct tagPID {     WORD32 dwPno;           /* 进程号 */     WORD32 dwDevId;         /* 设备ID,里面包含区号，局号信息，高8位为局号，次高16位为区号*/     WORD16 wModule;         /* MP编号 */     WORD16 wUnit;           /* 单板号 */     BYTE ucSUnit;           /* 单板上的处理器号 */     BYTE ucSubSystem;       /* 子系统号 */     BYTE ucRouteType;       /* 路由类型，区别业务/管理消息，左右板位，主备通信*/     BYTE ucExtendFlag;      /* 字节序标志 */ }PID;</pre> <p>在ATCA下，实现如下JID结构到PID结构的映射：</p> <pre>typedef struct tagJID {     WORD32 dwJno;           /* JOB号 */     WORD32 dwDevId;         /* 设备ID*/     WORD16 wModule;         /* MP编号 */     WORD16 wUnit;           /* 单板号 */     BYTE ucSUnit;           /* 单板上的处理器号 */     BYTE ucSubSystem;       /* 子系统号 */     BYTE ucRouteType;       /* 路由类型，区别业务/管理消息，左右板位，主备通信*/     BYTE ucExtendFlag;      /* 扩展标志 */ }JID;</pre>

## 10.4.2.1.3 PPLAT\_SIP\_OSS\_GetPIDByPNO

## 10.4.2.1.3.1 接口说明

函数名称		PPLAT_SIP_OSS_GetPIDByPNO	类型（函数/宏）	函数
功能详细说明		SIP平台在非ATCA下，对支撑函数OSS_GetPIDByPNO(WORD32 dwPno, PID *ptPid)的封装映射。通过进程PNO获得本板进程的标识。该接口通过给定的进程PNO，得到本板内相应的进程标识。相关的接口有OSS_ConstructPID、OSS_GetPIDByName和OSS_GetFirstPID。在ATCA下，是对支撑函数OSS_GetJIDByJNO(WORD32 dwJno,JID *ptJid)的封装映射。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				

函数声明		
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_GetPIDByPNO(WORD32 dwPno, PID *ptPid)		
序号	参数名	返回值说明
1	无	PPLAT_SIP_OSS_SUCCESS 成功 PPLAT_SIP_OSS_SCHE_ERR_PTR_NULL 入参指针为空
序号	参数名	输入参数说明
1		
2		
3		
序号	参数名	输出参数说明
1	dwPno	进程号
2	ptPid	通过该出参将构造出的PID返回
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.2.1.3.2 参数详细说明

参数名	类型	取值范围	解释
dwPno	WORD32		对非ATCA来说,就是进程号PNO,对ATCA来说,实现JNO到PNO的映射
ptPid	PID*		<p>对非ATCA来说,与如下结构完全一致。</p> <pre>typedef struct tagPID {     WORD32 dwPno;           /* 进程号 */     WORD32 dwDevId;         /* 设备ID,里面包含区号,局号信息,高8位为局号,次高16位为区号*/     WORD16 wModule;         /* MP编号 */     WORD16 wUnit;           /* 单板号 */     BYTE ucSUnit;           /* 单板上的处理器号 */     BYTE ucSubSystem;       /* 子系统号 */     BYTE ucRouteType;       /* 路由类型,区别业务/管理消息,左右板位,主备通信*/     BYTE ucExtendFlag;      /* 字节序标志 */ }PID;</pre> <p>对ATCA来说,实现如下JID结构到PID的映射:</p> <pre>typedef struct tagJID {     WORD32 dwJno;           /* JOB号 */     WORD32 dwDevId;         /* 设备ID*/     WORD16 wModule;         /* MP编号 */     WORD16 wUnit;           /* 单板号 */     BYTE ucSUnit;           /* 单板上的处理器号 */     BYTE ucSubSystem;       /* 子系统号 */     BYTE ucRouteType;       /* 路由类型,区别业务/管理消息,左右板位,主备通信*/     BYTE ucExtendFlag;      /* 扩展标志 */ }JID;</pre>

## 10.4.2.1.4 PPLAT\_SIP\_OSS\_ConstructPNO

## 10.4.2.1.4.1 接口说明

函数名称		PPLAT_SIP_OSS_ConstructPNO	类型（函数/宏）	函数
功能详细说明		SIP平台在非ATCA下，是对支撑函数OSS_ConstructPNO(WORD16 wProcType, WORD16 wInstanceNo)的封装映射。构造一个有效的进程号。该接口由应用程序调用触发，接口下层为调度管理系统。在ATCA下，是对支撑函数OSS_ConstructJNO(WORD16 wJobType, WORD16 wInstanceNo)的封装映射		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD32 PPLAT_SIP_OSS_ConstructPNO(WORD16 wProcType, WORD16 wInstanceNo)				
序号	参数名	返回值说明		
1	无	对非ATCA来说，就是返回的有效的进程号。对ATCA来说，实现返回的有效的JNO到PNO的映射。		
序号	参数名	输入参数说明		
1	wProcType	进程类型		
2	wInstanceNo	进程实例号		
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.2.1.4.2 参数详细说明

参数名	类型	取值范围	解释
wProcType	WORD16	N/A	对非ATCA来说，与进程类别wProcType完全一致，对ATCA来说，实现从wJobType到wProcType的映射
wInstanceNo	WORD16	N/A	

## 10.4.2.2 SD-I-PPLAT-SIP-OSS-0020 采用 ATCA 后需要修改的 SIP 平台支撑进程间通信管理接口

## 10.4.2.2.1 PPLAT\_SIP\_OSS\_SendAsynMsg

## 10.4.2.2.1.1 接口说明

函数名称	PPLAT_SIP_OSS_SendAsynMsg	类型（函数/宏）	函数
------	---------------------------	----------	----

功能详细说明		<p>SIP平台在非ATCA时，是对支撑函数OSS_SendAsynMsg(WORD16 wMsgId, BYTE *pucData, WORD16 wMsgLen, BYTE ucCommType, PID *ptReceiver)的封装。发送普通异步消息。该接口由应用程序调用触发，由通讯管理模块提供，向上层提供发送普通异步消息的功能，包括CPU间的异步消息发送。对ATCA来说，是对支持函数OSS_SendAsynMsg(WORD32 dwMsgId, BYTE *pucData, WORD16 wMsgLen, BYTE ucVer, BYTE ucPriority, VOID *pReceiver)的封装映射。其中的BYTE ucVer, BYTE ucPriority字段在封装函数中给出，调用者可以不用关心，对封装函数中的BYTE ucCommType字段，封装函数做了屏蔽。</p> <p>与其他的支撑接口缺省实现函数一样，在核心代码中，采用后缀添加_wrap的方式，对该接口进行了封装。与其他支撑接口缺省实现不同的是，该接口函数采用了两个不同的封装函数：PPLAT_SIP_OSS_SendAsynMsg_wrap()和PPLAT_SIP_OSS_SendAsynMsg_Ext_wrap()。前者将路由类别强制设置为主板，在一般的场合可以使用该封装接口。而后者则是采用了简单封装的方式（对参数未进行设置），该封装接口主要用于一些不希望将路由类别强制设置为主板的场合。使用者需要关注。</p>
性能		N/A
序号	分类	使用的全局及静态变量
1	全局	无
2	静态	无
资源限制		
N/A		
函数声明		
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_SendAsynMsg(WORD16 wMsgId, BYTE *pucData, WORD16 wMsgLen, BYTE ucCommType, PID *ptReceiver )		
序号	参数名	返回值说明
1	无	PPLAT_SIP_OSS_SUCCESS :成功 PPLAT_SIP_OSS_ERR_COMM_SEND_ASYNC_MSG_FAILED 发送普通消息失败 PPLAT_SIP_OSS_ERR_COMM_INVALID_PARAMETER 参数无效
序号	参数名	输入参数说明
1	wMsgId	消息事件号
2	pucData	消息体的指针
3	wMsgLen	消息体长度
4	ucCommType	选用的协议类型。对非ATCA来说有效，对ATCA来说，本字段无效，封装函数进行屏蔽
5	ptReceiver	消息接收方的PID
6	ucVer	应用版本号。对非ATCA来说无效，对ATCA来说，封装函数内部添加，调用者不用关心
7	ucPriority	消息优先级。对非ATCA来说无效，对ATCA来说，封装函数内部添加，调用者不用关心
序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.2.2.1.2 参数详细说明



参数名	类型	取值范围	解释
wMsgId	WORD16		
pucData	BYTE*		
wMsgLen	WORD16		
ucCommType	BYTE	PPLAT_SIP_COMM_RELIABLE 或 PPLAT_SIP_COMM_RELIABLE	对非ATCA来说有效，对ATCA来说，本字段无效，封装函数进行屏蔽
ptReceiver	PID *		typedef struct tagPID { WORD32 dwPno; /* 进程号 */ WORD32 dwDevId; /* 设备ID,里面包含区号, 局号信息, 高8位为局号, 次高16位为区号*/ WORD16 wModule; /* MP编号 */ WORD16 wUnit; /* 单板号 */ BYTE ucSUnit; /* 单板上的处理器号 */ BYTE ucSubSystem; /* 子系统号 */ BYTE ucRouteType; /* 路由类型, 区别业务/管理消息, 左右板位, 主备通信*/ BYTE ucExtendFlag; /* 字节序标志 */ }PID;
ucVer	BYTE	N/A	应用版本号。对非ATCA来说无效，对ATCA来说，封装函数内部添加，调用者不用关心
ucPriority	BYTE	N/A	消息优先级。对非ATCA来说无效，对ATCA来说，封装函数内部添加，调用者不用关心

#### 10.4.2.3 SD-I-PPLAT-SIP-OSS-0020 采用 ATCA 后需要修改的 SIP 平台支撑系统控制接口

##### 10.4.2.3.1 PPLAT\_SIP\_SCS\_WaitJob

###### 10.4.2.3.1.1 接口说明

函数名称		PPLAT_SIP_SCS_WaitJob	类型（函数/宏）	函数
功能详细说明		SIP平台在非ATCA时本函数不其作用，在ATCA时，是对系统控制函数SCS_WaitJob(WORD32 dwJno, BYTE ucJobState, WORD32 dwTimeOut)的封装映射。用于某进程依赖于另一进程的状态时，对本进程的阻塞场景。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_SCS_STATUS SCS_WaitJob(WORD32 dwJno, BYTE ucJobState, WORD32 dwTimeOut)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_SCS_OK PPLAT_SIP_SCS_ERR  等待JOB状态成功 等待JOB状态失败		
序号	参数名	输入参数说明		
1	dwJno	等待JOB的JNO		

2	ucJobState	等待JOB状态, PPLAT_SIP_JOB_STATE_INIT (BYTE)0 /* 初始状态 */ PPLAT_SIP_JOB_STATE_MASTER (BYTE)1 /* 主用状态 */ PPLAT_SIP_JOB_STATE_SLAVE (BYTE)2 /* 备用状态 */ PPLAT_SIP_JOB_STATE_FAULT (BYTE)3 /* 故障状态 */ PPLAT_SIP_JOB_STATE_UNKNOWN (BYTE)255 /*未知状态*/
3	dwTimeOut	等待时长 PPLAT_SIP_NO_WAIT, 立即返回。 PPLAT_SIP_WAIT_FOREVER, 永久阻塞。 其它, 阻塞时长。
序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.2.3.1.2 参数详细说明

参数名	类型	取值范围	解释
dwJno	WORD32	N/A	等待JOB的JNO
ucJobState	BYTE	N/A	等待JOB状态
dwTimeOut	WORD32	N/A	等待时长

## 10.4.3 采用 ATCA 后与原来接口相比需要修改的支撑函数/接口相关结构体封装

```

#define PPLAT_SIP_JID_T JID
typedef struct tagJID
{
    WORD32 dwJno; /* JOB 号 */
    WORD32 dwDevId; /* 设备 ID*/
    WORD16 wModule; /* MP 编号 */
    WORD16 wUnit; /* 单板号 */
    BYTE ucSUnit; /* 单板上的处理器号 */
    BYTE ucSubSystem; /* 子系统号 */
    BYTE ucRouteType; /* 路由类型, 区别业务/管理消息, 左右板位, 主备通信*/
    BYTE ucExtendFlag; /* 扩展标志 */
}JID;

#define PPLAT_SIP_JOB_STATE_INIT JOB_STATE_INIT /* 初始状态 */
#define PPLAT_SIP_JOB_STATE_MASTER JOB_STATE_MASTER /* 主用状态 */

```

```
#define PPLAT_SIP_JOB_STATE_SLAVE JOB_STATE_SLAVE /* 备用状态 */
#define PPLAT_SIP_JOB_STATE_FAULT JOB_STATE_FAULT /* 故障状态 */
#define PPLAT_SIP_JOB_STATE_UNKNOWN JOB_STATE_UNKNOWN /*未知状态*/
#define PPLAT_SIP_SCS_OK SCS_OK /*等待JOB状态成功*/
#define PPLAT_SIP_SCS_ERR SCS_ERR/*等待 JOB 状态失败*/
#define PPLAT_SIP_NO_WAIT NO_WAIT/*立即返回*/
#define PPLAT_SIP_WAIT_FOREVER WAIT_FOREVER/*永久阻塞*/
```

#### 10.4.4 不需要修改的支撑封装函数/接口列表

序号	函数/接口宏名称	类型（函数/宏）	功能描述
1	PPLAT_SIP_OSS_Malloc	函数	对支撑函数OSS_Malloc的封装
2	PPLAT_SIP_OSS_Free	函数	对支撑函数OSS_Free的封装
3	PPLAT_SIP_OSS_SetRelativeTimer	函数	对支撑函数OSS_SetRelativeTimer的封装
4	PPLAT_SIP_OSS_SetLoopTimer	函数	对支撑函数OSS_SetLoopTimer的封装
5	PPLAT_SIP_OSS_KillTimerByTimerId	函数	对支撑函数OSS_KillTimerByTimerId的封装
6	PPLAT_SIP_OSS_KillTimer	函数	对支撑函数OSS_KillTimer的封装
7	PPLAT_SIP_OSS_GetCurrentSeconds	函数	对支撑函数OSS_GetCurrentSeconds的封装
8	PPLAT_SIP_OSS_GetSysClock	函数	对支撑函数OSS_GetSysClock的封装
9	PPLAT_SIP_OSS_GetClockRate	函数	对支撑函数OSS_GetClockRate的封装
10	PPLAT_SIP_OSS_GetCurTick	函数	对支撑函数OSS_GetCurTick的封装
14	PPLAT_SIP_OSS_SetNextState	函数	对支撑函数OSS_SetNextState的封装
15	PPLAT_SIP_OSS_SetDefaultNextState	函数	对支撑函数OSS_SetDefaultNextState的封装
19	PPLAT_SIP_OSS_GetMsgDataLen	函数	对支撑函数OSS_GetMsgDataLen的封装
20	PPLAT_SIP_OSS_SynAck	函数	对支撑函数OSS_SynAck的封装
21	PPLAT_SIP_OSS_GetMsgId	函数	对支撑函数OSS_GetMsgId的封装
22	PPLAT_SIP_OSS_NtoHL	函数	对支撑函数OSS_NtoHL的封装
23	PPLAT_SIP_OSS_HtoNL	函数	对支撑函数OSS_HtoNL的封装

#### 10.4.5 不需要修改的支撑封装函数/接口和参数说明

##### 10.4.5.1 SD-I-PPLAT-SIP-OSS-0030 采用 ATCA 后不需要修改的 SIP 平台支撑内存管理接口

###### 10.4.5.1.1 PPLAT\_SIP\_OSS\_Malloc

###### 10.4.5.1.1.1 接口说明

函数名称	PPLAT_SIP_OSS_Malloc	类型（函数/宏）	函数
功能详细说明	SIP平台对支撑函数OSS_Malloc的封装。引入OSS_Malloc的目的只是为了加入对内存使用状况的统计功能。该函数只在初始化申请内存的时候使用，在运行过程中不允许使用。另外，此函数在支撑内部也要使用。		
性能	N/A		

序号	分类	使用的全局及静态变量	
1	全局	无	
2	静态	无	
资源限制			
N/A			
函数声明			
BYTE * PPLAT_SIP_OSS_Malloc (WORD32 dwSize);			
序号	参数名	返回值说明	
1	无	NULL	申请失败
		非NULL	内存地址指针
序号	参数名	输入参数说明	
1	dwSize	申请内存长度，以字节为单位	
2			
3			
序号	参数名	输出参数说明	
1			
2			
3			
算法及流程说明			
N/A			
其他限制或说明			
N/A			

## 10.4.5.1.1.2 参数详细说明

参数名	类型	取值范围	解释
dwSize	WORD32	0-0xffffffff	

## 10.4.5.1.2 PPLAT\_SIP\_OSS\_Free

## 10.4.5.1.2.1 接口说明

函数名称		PPLAT_SIP_OSS_Free	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_Free的封装。此函数同PPLAT_SIP_OSS_Free配对使用，并且，该函数只在初始化申请内存的时候使用，在运行过程中不允许使用。另外，此函数在支撑内部也要使用。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_Free(BYTE* pucBuf);				
序号	参数名	返回值说明		

1	无	PPLAT_SIP_OSS_SUCCESS 释放成功 PPLAT_SIP_OSS_ERROR_UNKNOWN 释放失败
序号	参数名	输入参数说明
1	pucBuf	待释放的内存地址指针
2		
3		
序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.5.1.2.2 参数详细说明

参数名	类型	取值范围	解释
pucBuf	BYTE*		指针

## 10.4.5.2 SD-I-PPLAT-SIP-OSS-0040 采用 ATCA 后不需要修改的 SIP 平台支撑定时器管理接口

## 10.4.5.2.1 PPLAT\_SIP\_OSS\_SetRelativeTimer

## 10.4.5.2.1.1 接口说明

函数名称		PPLAT_SIP_OSS_SetRelativeTimer	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_SetRelativeTimer的封装。该接口由应用程序调用触发，用于设置相对定时器，包括预定义定时器。当时长不足1个TICK时，进位为1个TICK。其余情况下，采取四舍五入的方法来将毫秒转换成TICK。相对定时器精度为一个TICK		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD32 PPLAT_SIP_OSS_SetRelativeTimer(BYTE ucTimerNo, SWORD32 sdwDuration, WORD32 dwParam)				
序号	参数名	返回值说明		
1	无	成功时返回定时器标识 失败时返回PPLAT_SIP_INVALID_TIMER_ID		
序号	参数名	输入参数说明		
1	ucTimerNo	定时器号，范围为 [1, PPLAT_SIP_PROC_TIMER_NUM – 1]。		
2	sdwDuration	定时器的长度，以毫秒为单位，当时长为负时，表示设置预定义定时器		
3	dwParam	带参数定时器所带的参数,如果等于PPLAT_SIP_PARAM_NULL,则认为设定的是不带参数定时器		

序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.5.2.1.2 参数详细说明

参数名	类型	取值范围	解释
ucTimerNo	BYTE	[1, PPLAT_SIP_PRO C_TIMER_NUM - 1]	
sdwDuration	SWORD32		
dwParam	WORD32		

## 10.4.5.2.2 PPLAT\_SIP\_OSS\_SetLoopTimer

## 10.4.5.2.2.1 接口说明

函数名称		PPLAT_SIP_OSS_SetLoopTimer	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_SetLoopTimer的封装。该接口用于申请循环定时器，包括预定义定时器。当时长不足1个TICK时，进位为1个TICK。其余情况下，采取四舍五入的方法将毫秒转换成TICK。循环定时器精度为一个TICK。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD32 PPLAT_SIP_OSS_SetLoopTimer(BYTE ucTimerNo, SWORD32 sdwDuration, WORD32 dwParam)				
序号	参数名	返回值说明		
1	无	成功时返回定时器标识 失败时返回PPLAT_SIP_INVALID_TIMER_ID		
序号	参数名	输入参数说明		
1	ucTimerNo	定时器号，范围为 [1, PPLAT_SIP_PROC_TIMER_NUM - 1]。		
2	sdwTimerLength	定时长度，以毫秒为单位，当时长为负时，表示设置预定义定时器		
3	dwParam	带入的参数，如果等于PPLAT_SIP_PARAM_NULL,即认为是不带参数定时器		
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				

其他限制或说明
N/A

## 10.4.5.2.2 参数详细说明

参数名	类型	取值范围	解释
ucTimerNo	ucTimerNo	[1, PPLAT_SIP_PROC_TIMER_NUM - 1]	
sdwTimerLength	SWORD32		
dwParam	WORD32		

## 10.4.5.2.3 PPLAT\_SIP\_OSS\_KillTimerByTimerId

## 10.4.5.2.3.1 接口说明

函数名称		PPLAT_SIP_OSS_KillTimerByTimerId	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_KillTimerByTimerId的封装。该函数根据定时器标识删除定时器。如果进程已经收到该定时器超时消息，再杀死该定时器，则会报错。在没有收到该定时器超时消息时，杀死该定时器，则进程不会再收到该定时器超时消息。该函数的功能与PPLAT_SIP_OSS_KillTimer()相同		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_KillTimerByTimerId(WORD32 dwTimerId)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_OSS_SUCCESS 成功 PPLAT_SIP_OSS_TIMER_ERR_INPUT_PARAM_ERROR 入参错误 PPLAT_SIP_OSS_TIMER_ERR_GET_PCB_FAIL 获取PCB失败 PPLAT_SIP_OSS_TIMER_ERR_KILL_SYN_TIMER 正在杀死同步定时器 PPLAT_SIP_OSS_TIMER_ERR_KILLING_TIMER_IN_ANOTHER_PROCESS 正在杀死其它进程的定时器 PPLAT_SIP_OSS_TIMER_ERR_P_OPERATION_FAIL P操作失败 PPLAT_SIP_OSS_TIMER_ERR_V_OPERATION_FAIL V操作失败 PPLAT_SIP_OSS_ERROR_UNKNOWN 未知错误		
序号	参数名	输入参数说明		
1	dwTimerId	定时器标识，范围为 [0, PPLAT_SIP_MAX_TMCB_NUM - 1]		
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.5.2.3.2 参数详细说明

参数名	类型	取值范围	解释
dwTimerId	WORD32	[0, PPLAT_SIP_MAX_TMCB_NUM - 1]	

## 10.4.5.2.4 PPLAT\_SIP\_OSS\_KillTimer

## 10.4.5.2.4.1 接口说明

函数名称		PPLAT_SIP_OSS_KillTimer	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_KillTimer的封装。该函数根据定时器号和定时器参数来删除定时器。如果进程已经收到该定时器超时消息，再杀死该定时器，则会报错。在没有收到该定时器超时消息时，杀死该定时器，则进程不会再收到该定时器超时消息。该函数的功能与PPLAT_SIP_OSS_KillTimerByTid()相同		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_KillTimer(BYTE ucTimerNo,WORD32 dwParam)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_OSS_SUCCESS 成功 PPLAT_SIP_OSS_TIMER_ERR_INPUT_PARAM_ERROR 入参错误 PPLAT_SIP_OSS_TIMER_ERR_GET_PCB_FAIL 获取PCB失败 PPLAT_SIP_OSS_TIMER_ERR_KILL_NOT_EXIST_TIMER 杀死不存在的定时器 PPLAT_SIP_OSS_TIMER_ERR_P_OPERATION_FAIL P操作失败 PPLAT_SIP_OSS_TIMER_ERR_V_OPERATION_FAIL V操作失败 PPLAT_SIP_OSS_ERROR_UNKNOWN 未知错误		
序号	参数名	输入参数说明		
1	ucTimerNo	定时器号，范围为 [0, PPLAT_SIP_PROC_TIMER_NUM - 1]		
2	dwParam	定时器参数，若杀死的为无参定时器，则填入PPLAT_SIP_PARAM_NULL；若为带参定时器，则填相应的参数		
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.5.2.4.2 参数详细说明

参数名	类型	取值范围	解释
-----	----	------	----



ucTimerNo	BYTE	[0, PPLAT_SIP_PROC_TIMER_NUM - 1]	
dwParam	WORD32		

#### 10.4.5.2.5 PPLAT\_SIP\_OSS\_GetCurrentSeconds

##### 10.4.5.2.5.1 接口说明

函数名称		PPLAT_SIP_OSS_GetCurrentSeconds	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_GetCurrentSeconds的封装。获得自2000年1月1日0时0分0秒起经历的秒数		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD32 PPLAT_SIP_OSS_GetCurrentSeconds(VOID)				
序号	参数名	返回值说明		
1	无	自2000年1月1日0时0分0秒起经历的秒数		
序号	参数名	输入参数说明		
1				
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

##### 10.4.5.2.5.2 参数详细说明

参数名	类型	取值范围	解释

#### 10.4.5.2.6 PPLAT\_SIP\_OSS\_GetSysClock

##### 10.4.5.2.6.1 接口说明

函数名称	PPLAT_SIP_OSS_GetSysClock	类型（函数/宏）	函数
------	---------------------------	----------	----

功能详细说明		SIP平台对支撑函数OSS_GetSysClock的封装。取得当前的时间。时间格式为：×年×月×日×时×分×秒。	
性能		N/A	
序号	分类	使用的全局及静态变量	
1	全局	无	
2	静态	无	
资源限制			
N/A			
函数声明			
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_GetSysClock(T_SysSoftClock *ptSysClock)			
序号	参数名	返回值说明	
1	无	PPLAT_SIP_OSS_SUCCESS PPLAT_SIP_OSS_TIMER_ERR_INPUT_PARAM_ERROR PPLAT_SIP_OSS_TIMER_ERR_P_OPERATION_FAIL PPLAT_SIP_OSS_TIMER_ERR_V_OPERATION_FAIL	
序号	参数名	输入参数说明	
1			
2			
3			
序号	参数名	输出参数说明	
1	ptSysClk	当前系统时钟，为指针	
2			
3			
算法及流程说明			
N/A			
其他限制或说明			
N/A			

## 10.4.5.2.6.2 参数详细说明

参数名	类型	取值范围	解释
ptSysClk	T_SysSoftClock*		<pre>typedef struct tagSYS_CLOCK {     WORD16  wSysYear;    /* 年 */     BYTE    ucSysMon;    /* 月 */     BYTE    ucSysDay;    /* 日 */      BYTE    ucSysHour;   /* 时 */     BYTE    ucPadding0;  /* 填充字*/     BYTE    ucSysMin;    /* 分 */     BYTE    ucSysSec;    /* 秒 */      WORD16  wMilliSec;   /* 毫秒 */     BYTE    ucSysWeek;   /* 周 */     BYTE    ucPadding1;  /* 填充字*/ }T_SysSoftClock; /* 系统软时钟结构，表示绝对时间 */</pre>

## 10.4.5.2.7 PPLAT\_SIP\_OSS\_GetClockRate

## 10.4.5.2.7.1 接口说明

本文中的所有信息均为中兴通讯股份有限公司内部信息，不得向外传播。

函数名称		PPLAT_SIP_OSS_GetClockRate	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_GetClockRate的封装。每秒的Tick数。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD16 PPLAT_SIP_OSS_GetClockRate(VOID)				
序号	参数名	返回值说明		
1	无	每秒的Tick数		
序号	参数名	输入参数说明		
1				
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.5.2.7.2 参数详细说明

参数名	类型	取值范围	解释

## 10.4.5.2.8 PPLAT\_SIP\_OSS\_GetCurTick

## 10.4.5.2.8.1 接口说明

函数名称		PPLAT_SIP_OSS_GetCurTick	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_GetCurTick的封装。该函数用于获取自系统启动后经过的Tick数，严格来说是OSS子系统定时管理模块开始初使化后的TICK数。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_GetCurTick(T_Tick *ptTick)				

序号	参数名	返回值说明
1	无	PPLAT_SIP_OSS_SUCCESS 成功 PPLAT_SIP_OSS_TIMER_ERR_INPUT_PARAM_ERROR 入参错误 PPLAT_SIP_OSS_TIMER_ERR_P_OPERATION_FAIL P操作失败 PPLAT_SIP_OSS_TIMER_ERR_V_OPERATION_FAIL V操作失败
序号	参数名	输入参数说明
1		
2		
3		
序号	参数名	输出参数说明
1	ptTick	为如下类型的指针,用于存放获得的自系统启动后经过的Tick数: typedef struct tagTick { WORD32 dwTickCountHigh; /*倍数*/ WORD32 dwTickCountLow; /* 当前tick数MOD (2的32次方-1) 之值*/ }T_Tick;
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.5.2.8.2 参数详细说明

参数名	类型	取值范围	解释
ptTick	T_Tick*		typedef struct tagTick { WORD32 dwTickCountHigh; /*倍数*/ WORD32 dwTickCountLow; /* 当前tick数MOD (2的32次方-1) 之值*/ }T_Tick;

## 10.4.5.3 SD-I-PPLAT-SIP-OSS-0050 采用 ATCA 后不需要修改的 SIP 平台支撑调度管理接口

## 10.4.5.3.1 PPLAT\_SIP\_OSS\_SetNextState

## 10.4.5.3.1.1 接口说明

函数名称		PPLAT_SIP_OSS_SetNextState	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_SetNextState的封装。设置当前调用进程的下一个业务状态并返回任务。该接口只用于进程自己状态跃迁，由应用程序调用触发，接口下层为调度管理系统。相关接口有OSS_SetDefaultNextState， OSS_GetCurState， OSS_GetPriDataPtr。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
VOID PPLAT_SIP_OSS_SetNextState(WORD16 wNewState)				

序号	参数名	返回值说明
1	无	
序号	参数名	输入参数说明
1	wNewState	进程的业务状态
2		
3		
序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.5.3.1.2 参数详细说明

参数名	类型	取值范围	解释
wNewState	WORD16		

## 10.4.5.3.2 PPLAT\_SIP\_OSS\_SetDefaultNextState

## 10.4.5.3.2.1 接口说明

函数名称		PPLAT_SIP_OSS_SetDefaultNextState	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_SetDefaultNextState的封装。返回任务。该接口只用于进程自己状态默认跃迁，由应用程序调用触发，接口下层为调度管理系统。相关接口有OSS_SetNextState，OSS_GetCurState，OSS_GetPriDataPtr。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
VOID OSS_SetDefaultNextState(VOID)				
序号	参数名	返回值说明		
1	无			
序号	参数名	输入参数说明		
1				
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				

其他限制或说明
N/A

## 10.4.5.3.2.2 参数详细说明

参数名	类型	取值范围	解释

## 10.4.5.3.3 PPLAT\_SIP\_OSS\_GetMsgDataLen

## 10.4.5.3.3.1 接口说明

函数名称		PPLAT_SIP_OSS_GetMsgDataLen	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_GetMsgDataLen的封装。获得当前消息的长度。该接口由应用程序调用触发，相关接口有OSS_GetMsgId，OSS_GetMsgDataLen, OSS_GetMsgType。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD16 PPLAT_SIP_OSS_GetMsgDataLen(VOID)				
序号	参数名	返回值说明		
1	无	当前消息的长度		
序号	参数名	输入参数说明		
1				
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.5.3.3.2 参数详细说明

参数名	类型	取值范围	解释

## 10.4.5.3.4 PPLAT\_SIP\_OSS\_SynAck

## 10.4.5.3.4.1 接口说明

函数名称		PPLAT_SIP_OSS_SynAck	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_SynAck的封装。用户使用该接口将同步应答消息封装在缓冲区中，以便返回给同步调用的发起者进程。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_SynAck(BYTE *pucData,WORD16 wMsgLen)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_OSS_SUCCESS ：成功 PPLAT_SIP_OSS_ERR_COMM_SEND_SYNACK_MSG_FAILED 同步应答失败 PPLAT_SIP_OSS_ERR_COMM_INVALID_PARAMETER 参数无效		
序号	参数名	输入参数说明		
1	pucData	消息体的指针		
2	wMsgLen	消息体长度		
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.5.3.4.2 参数详细说明

参数名	类型	取值范围	解释
pucData	BYTE*		指针
wMsgLen	WORD16		

## 10.4.5.3.5 PPLAT\_SIP\_OSS\_GetMsgId

## 10.4.5.3.5.1 接口说明

函数名称		PPLAT_SIP_OSS_GetMsgId	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_GetMsgId的封装。用于获取当前消息的事件号。该接口由应用程序调用触发，相关接口有OSS_GetMsgId，OSS_GetMsgDataLen, OSS_GetMsgType。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				

函数声明		
WORD16 PPLAT_SIP_OSS_GetMsgId(VOID)		
序号	参数名	返回值说明
1	无	当前消息的事件号。
序号	参数名	输入参数说明
1		
2		
3		
序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.4.5.3.5.2 参数详细说明

参数名	类型	取值范围	解释

## 10.4.5.3.6 PPLAT\_SIP\_OSS\_NtoHL

## 10.4.5.3.6.1 接口说明

函数名称		PPLAT_SIP_OSS_NtoHL	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_NtoHL的封装。将四字节整型数值从网络字节序转化为本地字节序。SIP平台中主要用于端口号的字节序转换。在接收从处理器外面到达的数据的时候，需要调用本接口将消息中各项的字节序转化为本地字节序，以保证数值的在本地正确解释。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD32 PPLAT_SIP_OSS_NtoHL(WORD32 dwNetLong)				
序号	参数名	返回值说明		
1	无	转化成本地字节序的四字节数据。		
序号	参数名	输入参数说明		
1	dwNetLong	需要进行转化的网络双字节数据。		
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				



算法及流程说明
N/A
其他限制或说明
N/A

## 10.4.5.3.6.2 参数详细说明

参数名	类型	取值范围	解释
dwNetLong	WORD32		

## 10.4.5.3.7 PPLAT\_SIP\_OSS\_HtoNL

## 10.4.5.3.7.1 接口说明

函数名称		PPLAT_SIP_OSS_HtoNL	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_HtoNL的封装。将四字节整型数值从本地字节序转化为网络字节序。SIP平台中主要用于端口号的字节序转换。在跨处理器发送的时候，需要调用本接口统一消息中各项的字节序为网络字节序，以保证数值的正确传输。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
WORD32 PPLAT_SIP_OSS_HtoNL(WORD32 dwLocalLong)				
序号	参数名	返回值说明		
1	无	转化成网络字节序的四字节数据。		
序号	参数名	输入参数说明		
1	dwLocalLong	需要进行转化的网络四字节数据。		
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.5.3.7.2 参数详细说明

参数名	类型	取值范围	解释
dwLocalLong	WORD32		

## 10.4.5.4 SD-I-PPLAT-SIP-OSS-0060 采用 ATCA 后不需要修改的 SIP 平台支撑通用接口

## 10.4.5.4.1 PPLAT\_SIP\_OSS\_GetPhysAddress

## 10.4.5.4.1.1 接口说明

函数名称		PPLAT_SIP_OSS_GetPhysAddress	类型（函数/宏）	函数
功能详细说明		SIP平台对支撑函数OSS_GetPhysAddress的封装。获得本板的物理地址。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
PPLAT_SIP_OSS_STATUS PPLAT_SIP_OSS_GetPhysAddress (T_PhysAddress *ptPhysAddr)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_OSS_SUCCESS :成功 PPLAT_SIP_OSS_SCHE_ERR_PTR_NULL 入参指针为空		
序号	参数名	输入参数说明		
1				
序号	参数名	输出参数说明		
1	ptPhysAddr	物理地址结构指针		
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.4.5.4.1.2 参数详细说明

参数名	类型	取值范围	解释
ptPhysAddr	T_PhysAddress *		typedef struct tagT_PhysAddress { BYTE ucBureauId; /* bureau no */ BYTE ucRackId; /* rack no */ BYTE ucShelfId; /* shelf no */ BYTE ucSlotId; /* slot no */ BYTE ucCpuId; /* cpu no */ BYTE aucPad[3]; /* pad */ }T_PhysAddress;

## 10.4.6 支撑函数/接口相关宏定义封装

## 10.4.6.1 返回值宏定义封装

```
#define PPLAT_SIP_OSS_SUCCESS OSS_SUCCESS
```

```
#define PPLAT_SIP_PROC_TIMER_NUM    PROC_TIMER_NUM

#define PPLAT_SIP_PARAM_NULL        PARAM_NULL

#define PPLAT_SIP_INVALID_TIMER_ID  INVALID_TIMER_ID

#define                                PPLAT_SIP_OSS_TIMER_ERR_INPUT_PARAM_ERROR
                                OSS_TIMER_ERR_INPUT_PARAM_ERROR/* 入参错误 */

#define PPLAT_SIP_OSS_TIMER_ERR_GET_PCB_FAIL    OSS_TIMER_ERR_GET_PCB_FAIL /* 获取 PCB 失败 */

#define PPLAT_SIP_OSS_TIMER_ERR_KILL_SYN_TIMER    OSS_TIMER_ERR_KILL_SYN_TIMER/*
正在杀死同步定时器 */

#define                                PPLAT_SIP_OSS_TIMER_ERR_KILLING_TIMER_IN_ANOTHER_PROCESS
                                OSS_TIMER_ERR_KILLING_TIMER_IN_ANOTHER_PROCESS/* 杀死其它进程的定时器 */

#define PPLAT_SIP_OSS_TIMER_ERR_P_OPERATION_FAIL OSS_TIMER_ERR_P_OPERATION_FAIL/*
P 操作失败 */

#define PPLAT_SIP_OSS_TIMER_ERR_V_OPERATION_FAIL OSS_TIMER_ERR_V_OPERATION_FAIL/*
V 操作失败 */

#define PPLAT_SIP_OSS_ERROR_UNKNOWN    OSS_ERROR_UNKNOWN/* 未知错误 */

#define PPLAT_SIP_OSS_SCHE_ERR_PTR_NULL OSS_SCHE_ERR_PTR_NULL/* 入参指针为空 */

#define                                PPLAT_SIP_OSS_SCHE_ERR_PROCTYPE_NOTEXIST
                                OSS_SCHE_ERR_PROCTYPE_NOTEXIST/* 对应的进程类型不在本板上 */

#define                                PPLAT_SIP_OSS_ERR_COMM_SEND_ASYNC_MSG_FAILED
                                OSS_ERR_COMM_SEND_ASYNC_MSG_FAILED /* 发送普通消息失败 */

#define                                PPLAT_SIP_OSS_ERR_COMM_INVALID_PARAMETER
                                OSS_ERR_COMM_INVALID_PARAMETER /* 参数无效 */

#define                                PPLAT_SIP_OSS_ERR_COMM_SEND_SYNACK_MSG_FAILED
                                OSS_ERR_COMM_SEND_SYNACK_MSG_FAILED/* 同步应答失败 */

#define PPLAT_SIP_MAX_TMCB_NUM        MAX_TMCB_NUM

#define                                PPLAT_SIP_OSS_TIMER_ERR_KILL_NOT_EXIST_TIMER
                                OSS_TIMER_ERR_KILL_NOT_EXIST_TIMER /*杀死不存在的定时器*/
```

#### 10.4.6.2 消息宏定义封装

```
#define EV_PPLAT_SIP_API_MASTER_POWER_ON    EV_MASTER_POWER_ON/* 主板上电消息 */

#define EV_PPLAT_SIP_API_SLAVE_POWER_ON    EV_SLAVE_POWER_ON/* 备板上电消息 */
```

### 10.4.6.3 其他宏定义封装

```
#define PPLAT_SIP_OSS_STATUS    OSS_STATUS
```

```
#define COMM_RELIABLE           (BYTE)0 /* 可靠协议 */
```

```
#define COMM_UNRELIABLE         (BYTE)1 /* 不可靠协议 */
```

### 10.4.7 支撑函数/接口相关结构体封装

```
#define PPLAT_SIP_SysSoftClock_T    T_SysSoftClock
```

```
#define PPLAT_SIP_Tick_T            T_Tick
```

其中:

```
typedef struct tagSYS_CLOCK
```

```
{
```

```
    WORD16  wSysYear;        /* 年 */
```

```
    BYTE     ucSysMon;        /* 月 */
```

```
    BYTE     ucSysDay;        /* 日 */
```

```
    BYTE     ucSysHour;       /* 时 */
```

```
    BYTE     ucPadding0;      /* 填充字*/
```

```
    BYTE     ucSysMin;        /* 分 */
```

```
    BYTE     ucSysSec;        /* 秒 */
```

```
    WORD16  wMilliSec;        /* 毫秒 */
```

```
    BYTE     ucSysWeek;       /* 周 */
```

```
    BYTE     ucPadding1;      /* 填充字*/
```

```
}T_SysSoftClock; /* 系统软时钟结构，表示绝对时间 */
```

```
typedef struct tagTick
```

```
{
```

```
    WORD32  dwTickCountHigh;  /*倍数*/
```

```
    WORD32  dwTickCountLow;   /* 当前 tick 数 MOD (2 的 32 次方-1) 之值*/
```

```
}T_Tick;
```

## 10.5 IP 承载接口

### 10.5.1 IP 承载封装函数/接口列表

序号	函数/接口宏名称	类型（函数/宏）	功能描述
1	PPLAT_SIP_BRS_socket	函数	对IP承载函数brs_socket的封装
2	PPLAT_SIP_BRS_bind	函数	对IP承载函数brs_bind的封装
3	PPLAT_SIP_BRS_listen	函数	对IP承载函数brs_listen的封装
4	PPLAT_SIP_BRS_connect	函数	对IP承载函数brs_connect的封装
5	PPLAT_SIP_BRS_close	函数	对IP承载函数brs_close的封装

6	PPLAT_SIP_BRS_recv	函数	对IP承载函数brs_recv的封装
7	PPLAT_SIP_BRS_recvfrom	函数	对IP承载函数brs_recvfrom的封装
8	PPLAT_SIP_BRS_recvmsg	函数	对IP承载函数brs_recvmsg的封装
9	PPLAT_SIP_BRS_sendto	函数	对IP承载函数brs_sendto的封装
10	PPLAT_SIP_BRS_send	函数	对IP承载函数brs_send的封装
	以下3个接口是IPSEC相关接口，作为对原版本的继承，虽然用不到，建议保留		
11	PPLAT_SIP_BrsSetSPandSAentry	函数	对IP承载函数BrsSetSPandSAentry的封装
12	PPLAT_SIP_BrsSetSALifetime	函数	对IP承载函数BrsSetSALifetime的封装
13	PPLAT_SIP_BrsDelSPandSAentry	函数	对IP承载函数BrsDelSPandSAentry的封装

## 10.5.2 IP 承载封装函数/接口和参数说明

### 10.5.2.1 SD-I-PPLAT-SIP-BRS-0010 SIP 平台 SOCKET 操作相关接口

#### 10.5.2.1.1 PPLAT\_SIP\_BRS\_socket

##### 10.5.2.1.1.1 接口说明

函数名称		PPLAT_SIP_BRS_socket	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_socket的封装。创建套接字。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_socket(int ip_domain, int type, int protocol)				
序号	参数名	返回值说明		
1	无	大于0的整数 成功创建的套接字的句柄 PPLAT_SIP_ESO_INVALPARA 无效参数 PPLAT_SIP_ESO_PROTONOSUPPORT 不支持该协议或类型插口 PPLAT_SIP_ESO_SOCKETOVERFLOW SOCKET空间不够 PPLAT_SIP_ESO_ISCONN 已经连接 PPLAT_SIP_ESO_PCBALLOC PCB分配错误或PCB已分配 PPLAT_SIP_ESO_PROTOREG 协议已注册 注：仅适用于RAW Socket PPLAT_SIP_ESO_SENDMSGFAIL OSS发送同步或异步消息失败 PPLAT_SIP_BRS_SOCKET_ERROR Socket错误或无效Socket		
序号	参数名	输入参数说明		
1	domain	指定套接字的协议族。通过宏指定： PLAT_SIP_BRS_AF_INET /* =2, IPv4。以前版本中=1 */ PPLAT_SIP_BRS_AF_INET6 /* =24, IPv6 */		
2	type	指定套接字类型，目前支持流式、数据报、原始套接字，分别通过宏 PPLAT_SIP_BRS SOCK_STREAM、PPLAT_SIP_BRS SOCK_DGRAM、 PPLAT_SIP_BRS SOCK_RAW指定		

3	protocol	指定协议类型，只对PPLAT_SIP_BRS SOCK_RAW类型的套接字有效，取值范围0到255。
序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.5.2.1.1.2 参数详细说明

参数名	类型	取值范围	解释
domain	int		PLAT_SIP_BRS_AF_INET /* =2, IPv4。以前版本中=1 */ PPLAT_SIP_BRS_AF_INET6 /* =24, IPv6 */
type	int		目前支持流式、数据报、原始套接字，分别通过宏PPLAT_SIP_BRS SOCK_STREAM、PPLAT_SIP_BRS SOCK_DGRAM、PPLAT_SIP_BRS SOCK_RAW指定
protocol	int	0到255。	只对PPLAT_SIP_BRS SOCK_RAW类型的套接字有效

## 10.5.2.1.2 PPLAT\_SIP\_BRS\_bind

## 10.5.2.1.2.1 接口说明

函数名称		PPLAT_SIP_BRS_bind	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_bind的封装。绑定套接字，即设定本地地址和端口。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_bind(int sockfd, struct brs_sockaddr *local_addr, int addrlen)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_BRS_SOCKET_OK 成功，数值为0。 PPLAT_SIP_ESO_INVALPARA 无效参数 PPLAT_SIP_ESO_INVALTCB TCP中的TCB无效 注：仅适用于TCP PPLAT_SIP_ESO_ADDRINUSE 地址或端口已被使用 注：仅适用于TCP, UDP PPLAT_SIP_ESO_ADDRNOTAVAIL 非本地地址 PPLAT_SIP_ESO_SENDMSGFAIL OSS发送同步或异步消息失败 PPLAT_SIP_BRS_SOCKET_ERROR Socket错误或无效Socket		
序号	参数名	输入参数说明		

1	sockfd	套接字句柄
2	local_addr	本地地址和端口
3	addrlen	地址结构长度
序号	参数名	输出参数说明
1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.5.2.1.2.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		
local_addr	struct brs_sockaddr*		结构指针: typedef struct brs_sockaddr { unsigned char sa_len; /* total length */ unsigned char sa_family; /* address family */ char sa_data[14]; /* actually longer; address value */ } brs_sockaddr;
addrlen	int		

## 10.5.2.1.3 PPLAT\_SIP\_BRS\_listen

## 10.5.2.1.3.1 接口说明

函数名称		PPLAT_SIP_BRS_listen	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_listen的封装。侦听套接字。只对PPLAT_SIP_BRS_SOCKET_STREAM类型的套接字有效。使套接字处于等待连接状态。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_listen(int sockfd, int backlog)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_BRS_SOCKET_OK 成功，数值为0。 PPLAT_SIP_ESO_INVALPARA 无效参数 PPLAT_SIP_ESO_TCP_START_ERROR TCP启动错误 注：仅适用于TCP PPLAT_SIP_ESO_SENDMSGFAIL OSS发送同步或异步消息失败 PPLAT_SIP_BRS_SOCKET_ERROR Socket错误或无效Socket		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2	backlog	等待连接的请求数		
3				
序号	参数名	输出参数说明		

1		
2		
3		
算法及流程说明		
N/A		
其他限制或说明		
N/A		

## 10.5.2.1.3.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		
backlog	int		

## 10.5.2.1.4 PPLAT\_SIP\_BRS\_connect

## 10.5.2.1.4.1 接口说明

函数名称		PPLAT_SIP_BRS_connect	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_connect的封装。连接套接字。对于PPLAT_SIP_BRS_SOCKET_RAW和PPLAT_SIP_BRS_SOCKET_DGRAM套接字，就是设定对端地址和端口。操作结果立即返回。对于PPLAT_SIP_BRS_SOCKET_STREAM套接字，就是向对端发起主动连接，属于异步操作。即使返回成功，并不表明TCP连接成功。当连接成功或失败时，协议栈进程都会以异步消息通知应用进程。详细说明见异步I/O。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_connect(int sockfd, struct brs_sockaddr *peer_addr, int addrlen)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_BRS_SOCKET_OK 成功，数值为0。 PPLAT_SIP_ESO_INVALIDPARAM 无效参数 PPLAT_SIP_ESO_TCP_START_ERROR TCP启动错误 注：仅适用于TCP PPLAT_SIP_ESO_ADDRNOTAVAIL 非本地地址 注：仅适用于UDP PPLAT_SIP_ESO_SENDMSGFAIL OSS发送同步或异步消息失败 PPLAT_SIP_BRS_SOCKET_ERROR Socket错误或无效Socket		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2	peer_addr	对端地址和端口		
3	addrlen	地址结构长度		
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				



N/A
其他限制或说明
N/A

## 10.5.2.1.4.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		
peer_addr	struct brs_sockaddr*		结构指针： typedef struct brs_sockaddr { unsigned char sa_len; /* total length */ unsigned char sa_family; /* address family */ char sa_data[14]; /* actually longer; address value */ } brs_sockaddr;
addrlen	int		

## 10.5.2.1.5 PPLAT\_SIP\_BRS\_close

## 10.5.2.1.5.1 接口说明

函数名称		PPLAT_SIP_BRS_close	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_close的封装。关闭套接字。对于所有结束不再使用的套接字都要求调用此函数，以释放相关资源。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_close(int sockfd)				
序号	参数名	返回值说明		
1	无	PPLAT_SIP_BRS_SOCKET_OK 成功，数值为0。 PPLAT_SIP_ESO_INVALIDPARA 无效参数 PPLAT_SIP_ESO_SENDMSGFAIL OSS发送同步或异步消息失败 PPLAT_SIP_BRS_SOCKET_ERROR Socket错误或无效Socket		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2				
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.5.2.1.5.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		

## 10.5.2.2 SD-I-PPLAT-SIP-BRS-0010 SIP 平台上行 SIP 报文接收相关接口

## 10.5.2.2.1 PPLAT\_SIP\_BRS\_recv

## 10.5.2.2.1.1 接口说明

函数名称		PPLAT_SIP_BRS_recv	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_recv的封装。接收数据。只支持非阻塞调用，对流式套接字、数据报套接字和原始套接字都适用。套接字收到数据后先发送消息通知应用进程，应用进程再调用brs_recv()接收数据。 Flags = PPLAT_SIP_BRS_SO_PEEK时得到的是缓冲区的数据长度，data,datalen在此没有意义，这时函数不会清除缓冲区的数据。如果flags != PPLAT_SIP_BRS_SO_PEEK,则从缓冲区中读取数据。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_recv(int sockfd, char* data, int datalen, int flags)				
序号	参数名	返回值说明		
1	无	1.返回值的最高位置1 则调用brs_recv（）失败，原因码值见附录： Socket错误码定义 2. 如果flags= PPLAT_SIP_BRS_SO_PEEK 返回的值为缓冲区的数据长度，否则， 返回的值为实际接收到的数据的长度		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2	datalen	接收数据缓存长度		
3	flags	发送数据标识		
序号	参数名	输出参数说明		
1	data	接收数据缓存		
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.5.2.2.1.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		

datalen	int		
flags	int		
data	char*		出参，接收数据指针

### 10.5.2.2.2 PPLAT\_SIP\_BRS\_rcvfrom

#### 10.5.2.2.2.1 接口说明

函数名称		PPLAT_SIP_BRS_rcvfrom	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_rcvfrom的封装。接收数据并返回数据的源地址。只支持非阻塞调用，对流式套接字、数据报套接字和原始套接字都适用。套接字收到数据后先发送消息通知应用进程，应用进程再调用brs_rcvfrom()接收数据。 Flags = PPLAT_SIP_BRS_SO_PEEK 时得到的是缓冲区的当前待读取数据包长度，data,datalen, from, fromlen在此没有意义，这时函数不会清除缓冲区的数据。如果flags != PPLAT_SIP_BRS_SO_PEEK,则从缓冲区中读取数据。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_rcvfrom(int sockfd, char* data, int datalen, int flags, struct brs_sockaddr * from, int * fromlen)				
序号	参数名	返回值说明		
1	无	1.返回值的最高位置1 则调用brs_rcvfrom（）失败，原因码值见附录： Socket 错误码定义 2. 如果flags= PPLAT_SIP_BRS_SO_PEEK 返回的值为缓冲区的当前待读取数据包长度，否则， 返回的值为实际接收到的数据的长度		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2	datalen	接收数据缓存长度		
3	flags	接收数据标识		
序号	参数名	输出参数说明		
1	data	接收数据缓存		
2	from	数据的源地址		
3	fromlen	源地址长度		
算法及流程说明				
N/A				
其他限制或说明				
N/A				

#### 10.5.2.2.2.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		
datalen	int		
flags	int		
data	char*		出参，接收数据指针
from	struct brs_sockaddr*		出参，结构指针： typedef struct brs_sockaddr { unsigned char sa_len; /* total length */

			unsigned char sa_family; /* address family */ char sa_data[14]; /* actually longer; address value */ } brs_sockaddr;
fromlen	int		出参，源地址长度

## 10.5.2.2.3 PPLAT\_SIP\_BRS\_rcvmsg

## 10.5.2.2.3.1 接口说明

函数名称		PPLAT_SIP_BRS_rcvmsg	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_rcvmsg的封装。用于带参数选项的接收，主要针对那些需要在每个接收包中带不同参数信息的应用。只支持非阻塞调用，对流式套接字、数据报套接字和原始套接字都适用。套接字收到数据后先发送消息通知应用进程，应用进程再调用brs_rcvmsg()接收数据。 Flags = PPLAT_SIP_BRS_SO_PEEK 时得到的是缓冲区的当前待读取数据包长度，输入参数msg在此没有意义，这时函数不会清除缓冲区的数据。如果flags != PPLAT_SIP_BRS_SO_PEEK,则从缓冲区中读取数据。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_rcvmsg(int sockfd, char *msg, int flags)				
序号	参数名	返回值说明		
1	无	1.返回值的最高位置1 则调用brs_rcvmsg（）失败，原因码值见附录： Socket 错误码定义 2. 如果flags= PPLAT_SIP_BRS_SO_PEEK 返回的值为缓冲区的当前待读取数据包长度，否则，返回的值为实际接收到的数据的长度		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2	flags	接收数据标识		
序号	参数名	输出参数说明		
1	msg	接收参数。		
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.5.2.2.3.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		
flags	int		
msg	char*		出参，接收参数指针

## 10.5.2.3 SD-I-PPLAT-SIP-BRS-0010 SIP 平台下行 SIP 报文发送相关接口

## 10.5.2.3.1 PPLAT\_SIP\_BRS\_sendto

## 10.5.2.3.1.1 接口说明

函数名称		PPLAT_SIP_BRS_sendto	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_sendto的封装。发送数据到指定的地址。对于流式套接字忽略地址参数。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_sendto(int sockfd, char *data, int datalen, int flags, struct brs_sockaddr *to, int tolen)				
序号	参数名	返回值说明		
1	无	非负的整数      已发送数据的长度 PPLAT_SIP_ESO_INVALIDPARAM    无效参数 PPLAT_SIP_ESO_WOULDBLOCK      TCP栈中包积压过多。 注：仅适用于TCP PPLAT_SIP_ESO_BUFFERNOTAVAIL    内存不够 PPLAT_SIP_ESO_SENDMSGFAIL      OSS发送同步或异步消息失败 PPLAT_SIP_BRS_SOCKET_ERROR    Socket错误或无效Socket		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2	data	发送数据缓存		
3	datalen	发送数据长度		
4	flags	发送数据标识		
5	to	发送的目的地址		
6	tolen	目的地址结构长度		
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.5.2.3.1.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		
data	char *		发送数据指针
datalen	int		
flags	int		
to	struct brs_sockaddr *		结构指针：

			typedef struct brs_sockaddr { unsigned char sa_len; /* total length */ unsigned char sa_family; /* address family */ char sa_data[14]; /* actually longer; address value */ } brs_sockaddr;
tolen	int		

## 10.5.2.3.2 PPLAT\_SIP\_BRS\_send

## 10.5.2.3.2.1 接口说明

函数名称		PPLAT_SIP_BRS_send	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数brs_send的封装。发送数据。目前只支持非阻塞、异步发送数据，即不保证数据肯定能发送成功。对于数据报和原始套接字，已经调用brs_connect接口指定对端地址信息也可以此函数发送数据。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BRS_send(int sockfd, char *data, int datalen, int flags)				
序号	参数名	返回值说明		
1	无	非负的整数 已发送数据的长度 PPLAT_SIP_ESO_INVALIDPARA 无效参数 PPLAT_SIP_ESO_WOULDBLOCK TCP栈中包积压过多。 注：仅适用于TCP PPLAT_SIP_ESO_BUFFERNOTAVAIL 内存不够 PPLAT_SIP_ESO_SENDMSGFAIL OSS发送同步或异步消息失败 PPLAT_SIP_BRS_SOCKET_ERROR Socket错误或无效Socket		
序号	参数名	输入参数说明		
1	sockfd	套接字句柄		
2	data	发送数据缓存		
3	datalen	发送数据长度		
4	flags	发送数据标识，紧急或普通。		
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.5.2.3.2.2 参数详细说明

参数名	类型	取值范围	解释
sockfd	int		
data	char *		发送数据指针
datalen	int		
flags	int		

## 10.5.2.4 SD-I-PPLAT-SIP-BRS-0010 SIP 平台 IPSEC 相关接口

## 10.5.2.4.1 PPLAT\_SIP\_BrsSetSPandSAentry

## 10.5.2.4.1.1 接口说明

函数名称		PPLAT_SIP_BrsSetSPandSAentry	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数BrsSetSPandSAentry的封装。用于创建IPSEC SA。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BrsSetSPandSAentry(T_SP_SELECTOR *pSpSelector,T_SA_ENTRY *pSAEntry )				
序号	参数名	返回值说明		
1	无	0        创建成功 -1       创建失败		
序号	参数名	输入参数说明		
1	pSpSelector	创建SA所需要的SP入参		
2	pSAEntry	创建SA所需要的SA入参		
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.5.2.4.1.2 参数详细说明

参数名	类型	取值范围	解释
pSpSelector	T_SP_SELECTOR*		结构指针 typedef struct _sp_selector { BYTE direction; WORD16 upper_layer_protocol; struct brs_rc_addrlist src; struct brs_rc_addrlist dst; } T_SP_SELECTOR;
pSAEntry	T_SA_ENTRY*		结构指针 typedef struct _sa_entry { unsigned int spi_peer;       /*对端spi,只有出向才有意义*/ unsigned int spi_local;      /*本端spi,只有入向才有意义*/ }

			WORD16 nat_srcport; /*nat_t 源端口*/对NAT入向为公网*/ WORD16 nat_dstport; /*nat_t 目的端口*/对NAT出向为公网*/ u_int wsize; /* window size, i.g. 4 bytes */ long lifetime; BYTE mode; /* 封装模式*/ BYTE security_protocol; /*安全协议*/ BYTE encry_alg; /*加密算法*/ BYTE integ_alg; /* 认证算法*/ BYTE ivlen; /* length of IV */ BYTE encry_key_len; /*加密 密钥的长度 */ BYTE integ_key_len; /*认证 密钥的长度 */ BYTE pad; /*为了4字节对齐的填充为*/ BYTE encry_key[32]; BYTE integ_key[32]; } T_SA_ENTRY;
--	--	--	--

#### 10.5.2.4.2 PPLAT\_SIP\_BrsSetSALifetime

##### 10.5.2.4.2.1 接口说明

函数名称		PPLAT_SIP_BrsSetSALifetime	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数BrsSetSALifetime的封装。用于更新IPSEC SA的生命周期。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int PPLAT_SIP_BrsSetSALifetime(T_SP_SELECTOR *pSpSelector,int lifetime )				
序号	参数名	返回值说明		
1	无	0        更新成功 -1       更新失败		
序号	参数名	输入参数说明		
1	pSpSelector	更新SA所需要的SP入参		
2	lifetime	更新SA所需要的生命周期入参		
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				



## 10.5.2.4.2.2 参数详细说明

参数名	类型	取值范围	解释
pSpSelector	T_SP_SELECTOR*		结构指针 typedef struct _sp_selector { BYTE direction; BYTE policy; long lifetime; WORD16 upper_layer_protocol; struct brs_rc_addrlist src; struct brs_rc_addrlist dst; } T_SP_SELECTOR;
lifetime	int		

## 10.5.2.4.3 PPLAT\_SIP\_BrsDelSPandSAentry

## 10.5.2.4.3.1 接口说明

函数名称		PPLAT_SIP_BrsDelSPandSAentry	类型（函数/宏）	函数
功能详细说明		SIP平台对IP承载函数BrsDelSPandSAentry的封装。用于更新IPSEC SA的生命周期。		
性能		N/A		
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
N/A				
函数声明				
int BrsDelSPandSAentry(T_SP_SELECTOR *pSpSelector)				
序号	参数名	返回值说明		
1	无	0	更新成功	
		-1	更新失败	
序号	参数名	输入参数说明		
1	pSpSelector	删除SA所需要的SP入参		
3				
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
N/A				
其他限制或说明				
N/A				

## 10.5.2.4.3.2 参数详细说明

参数名	类型	取值范围	解释
pSpSelector	T_SP_SELECTOR*		结构指针 typedef struct _sp_selector {

			BYTE direction; BYTE policy; long lifetime; WORD16 upper_layer_protocol; struct brs_rc_addrlist src; struct brs_rc_addrlist dst; } T_SP_SELECTOR;

### 10.5.3 IP 承载函数/接口相关宏定义封装

#### 10.5.3.1 返回值宏定义封装

```
#define PPLAT_SIP_BRS_SOCKET_OK      BRS_SOCKET_OK /* 成功, 数值为 0 */。
#define PPLAT_SIP_ESO_INVALIDPARAM  ESO_INVALIDPARAM /* 参数 */
#define PPLAT_SIP_ESO_PROTONOSUPPORT ESO_PROTONOSUPPORT /* 不支持该协议或类型插
    口 */
#define PPLAT_SIP_ESO_SOCKETOVERFLOW ESO_SOCKETOVERFLOW /* SOCKET 空间不够 */
#define PPLAT_SIP_ESO_ISCONN         ESO_ISCONN /* 已经连接 */
#define PPLAT_SIP_ESO_PCBBALLOC      ESO_PCBBALLOC /* PCB 分配错误或 PCB 已分配 */
#define PPLAT_SIP_ESO_PROTOREG       ESO_PROTOREG /* 协议已注册 注: 仅适用于 RAW Socket */
#define PPLAT_SIP_ESO_SENDMSGFAIL    ESO_SENDMSGFAIL /* OSS 发送同步或异步消息失败 */
#define PPLAT_SIP_BRS_SOCKET_ERROR   BRS_SOCKET_ERROR /* Socket 错误或无效 Socket */
#define PPLAT_SIP_ESO_INVALIDTCB     ESO_INVALIDTCB /* TCP 中的 TCB 无效 注: 仅适用于 TCP */
#define PPLAT_SIP_ESO_ADDRINUSE      ESO_ADDRINUSE /* 地址或端口已被使用 注: 仅适用
    于 TCP, UDP */
#define PPLAT_SIP_ESO_ADDRNOTAVAIL   ESO_ADDRNOTAVAIL /* 非本地地址 */
#define PPLAT_SIP_ESO_TCP_START_ERROR ESO_TCP_START_ERROR /* TCP 启动错误 注: 仅适
    用于 TCP */
#define PPLAT_SIP_ESO_WOULDBLOCK     ESO_WOULDBLOCK /* TCP 栈中包积压过多。注: 仅适
    用于 TCP */
#define PPLAT_SIP_ESO_BUFFERNOTAVAIL ESO_BUFFERNOTAVAIL /* 内存不够 */
```

#### 10.5.3.2 类型和标识宏定义封装

```
/* 域类型 */
#define PPLAT_SIP_BRS_AF_INET        BRS_AF_INET /* =2, IPv4。以前版本中=1 */
#define PPLAT_SIP_BRS_AF_INET6      BRS_AF_INET6 /* =24, IPv6 */
/* 数据类型 */
#define PPLAT_SIP_BRS SOCK_STREAM    BRS SOCK_STREAM
#define PPLAT_SIP_BRS SOCK_DGRAM    BRS SOCK_DGRAM
#define PPLAT_SIP_BRS SOCK_RAW      BRS SOCK_RAW
/* 发送数据标识 */
#define PPLAT_SIP_BRS SO_PEEK        BRS SO_PEEK
```

#### 10.5.3.3 消息宏定义封装

```
#define PPLAT_SIP_MSG_BRSSOCKET_MSG MSG_BRSSOCKET_MSG
```

### 10.5.4 IP 承载函数/接口相关结构体封装

```
#define PPLAT_SIP_brs_sockaddr      brs_sockaddr
#define PPLAT_SIP_BRS_SOCKET_IO_MSG BRS_SOCKET_IO_MSG
```

其中：

```
typedef struct brs_sockaddr {
    unsigned char  sa_len;    /* total length */
    unsigned char sa_family; /* address family */
    char sa_data[14]; /* actually longer; address value */
} brs_sockaddr;

typedef struct brs_socket_io_msg
{
    unsigned long type;      /* 见前面消息类型定义 */
    unsigned long s;         /* 发送消息的套接字 */
    unsigned long cookie;    /* 留给应用进程扩展用 */
} BRS_SOCKET_IO_MSG;
```

## 10.6 SIP 平台内部接口

### 10.6.1 SLB 与 TP 层间的用户接口（SLBUI）定义

注：当前设计不支持项目在 SLB 层跨层指定处理消息的 SIP TU 处理进程实例。

#### 10.6.1.1 SD-F-PPLAT-SIP-SLBUI-0010 SLB 模块把 SIP 消息发到 TP EV\_PPLAT\_SIP\_MSG\_IND

消息名称		方向	
EV_PPLAT_SIP_MSG_IND		SLB-> TP 或者 SIP TP 管理进程->SIP TP 处理进程	
消息功能描述			
SLB 模块发送 SIP 消息到 TP 或者 SIP TP 管理进程发送上行消息到 SIP TP 处理进程			
应用网元			
消息结构（详细信息见下方参数说明表）			
<pre>typedef struct {     PPLAT_SIP_IPADDR_T  tSrcIpPort;     PPLAT_SIP_IPADDR_T  tDstIpPort;     BYTE                 bProtocol;     PPLAT_SIP_IPSEC_HANDLE_T tHandle;     PPLAT_SIP_EXTERNAL_INFO_T tExternalInfo;     WORD16               wMsgBufLen;     BYTE                 abMsgBuffer[1]; } PPLAT_SIP_PACK  PPLAT_SIP_MSG_IND_T;</pre>			
其他限制或说明			
具体定义见《NGN PPLAT V1.00.10_接口说明书》。			
参数名	类型	取值范围	解释
tSrcIpPort	PPLAT_SIP_IPADDR_T		收到的 SIP 消息的源地址端口
tDstIpPort	PPLAT_SIP_IPADDR_T		收到的 SIP 消息的目的地址端口

bProtocol	BYTE		承载协议类型
tHandle	PPLAT_SIP_IPSEC_HANDLE_T		接收到消息的 IPsecHandle 的句柄，具体定义见《NGN PPLAT V1.01.10_接口说明书》
tExternalInfo	PPLAT_SIP_EXTERNAL_INFO_T		扩展信息，具体定义见《NGN PPLAT V1.01.10_接口说明书》
wMsgBufLen	WORD16		SIP 消息长度
abMsgBuffer	BYTE		存储 SIP 消息的首地址，SIP 消息为文本格式

注：SLB 是 SSS2.0 中提出的一个概念，表示流量均衡模块，其负责接收初始请求消息，变更将该消息分配到各个处理板进行处理。详见《ZXUN SSS V2.00.10\_系统方案.doc》

#### 10.6.1.2 SD-F-PPLAT-SIP-SLBUI-0020 TP 把 SIP 消息发送到 SLB EV\_PPLAT\_SIP\_MSG\_REQ

消息名称		方向	
EV_PPLAT_SIP_MSG_REQ		TP-> SLB 或者 SIP TP 处理进程 ->SIP TP 管理进程	
消息功能描述			
TP 发送 SIP 消息到 SLB 或者 SIP TP 处理进程发送下行消息到 SIP TP 管理进程			
应用网元			
消息结构（详细信息见下方参数说明表）			
typedef struct { PPLAT_SIP_SAP_ID_T  tSuld; /* TR 的索引 */ PPLAT_SIP_IPADDR_T  tSrcIpPort; PPLAT_SIP_IPADDR_T  tDstIpPort; BYTE                 bProtocol; PPLAT_SIP_IPSEC_HANDLE_T tHandle; PPLAT_SIP_EXTERNAL_INFO_T tExternalInfo; WORD16              wMsgBufLen; BYTE                 abMsgBuffer[1]; } PPLAT_SIP_PACK  PPLAT_SIP_MSG_REQ_T;			
其他限制或说明			
具体定义见《NGN PPLAT V1.00.10_接口说明书》。			
参数名	类型	取值范围	解释
tSuld	PPLAT_SIP_SAP_ID_T		TR 的索引。
tSrcIpPort	PPLAT_SIP_IPADDR_T		收到的 SIP 消息的源地址端口
tDstIpPort	PPLAT_SIP_IPADDR_T		收到的 SIP 消息的目的地址端口
bProtocol	BYTE		承载协议类型
tHandle	PPLAT_SIP_IPSEC_HANDLE_T		发送消息的 IPsecHandle 的句柄。具体定义见《NGN PPLAT V1.01.10_接口说明书》
tExternalInfo	PPLAT_SIP_EXTERNAL_INFO_T		扩展信息，具体定义见《NGN PPLAT V1.01.10_接口说明书》
wMsgBufLen	WORD16		SIP 消息长度
abMsgBuffer	BYTE		存储 SIP 消息的首地址，SIP 消息为文本

## 10.6.2 内存分配器接口

## 10.6.2.1 SD-F-PPLAT-SIP-ALLOCATOR-0010 内存分配器创建接口

## 10.6.2.1.1 P\_SIP\_ALLOCATOR\_create

函数名称		P_SIP_ALLOCATOR_create	类型（函数/宏）	功能函数
功能详细说明		内存分配器创建接口，由外部指示内存分配器管理的内存资源是由外部传入还是由内存分配器自行申请。在该接口中队内存分配器结构进行初始化，建立对内存资源的管理队列。当内存由外部传入时，如果传入的内存不足，内存分配器创建失败；当内存由内存分配器自行申请时，如果申请失败，内存分配器创建失败		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
BOOL8 P_SIP_ALLOCATOR_create(IN PPLAT_SIP_ALLOCATOR_CREATE_IN_T* ptCreateIn, OUT PPLAT_SIP_ALLOCATOR_T* ptAllocator);				
序号	参数类型	返回值说明		
1	BOOL8	成功返回 TRUE；失败返回 FALSE		
序号	参数名	输入参数说明		
1	ptCreateln	创建内存分配器入参结构指针,结构定义参见《NGN PPLAT SIP V1.01.10_接口说明书.doc》		
序号	参数名	输出参数说明		
1	ptAllocator	内存分配器结构指针,结构定义参见《NGN PPLAT SIP V1.01.10_接口说明书.doc》		
算法及流程说明				
其他限制或说明				

## 10.6.2.2 SD-F-PPLAT-SIP-ALLOCATOR-0020 内存分配器销毁接口

## 10.6.2.2.1 P\_SIP\_ALLOCATOR\_destroy

函数名称		P_SIP_ALLOCATOR_destroy	类型（函数/宏）	功能函数
功能详细说明		内存分配器销毁接口，如果内存分配器管理的内资源由内存分配器自身申请的，则在本接口中释放内存分配器管理的内存资源；如果内存分配器管理的内资源由外部传入，则在本接口中不进行释放。		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				

函数声明		
VOID P_SIP_ALLOCATOR_destroy (IN PPLAT_SIP_ALLOCATOR_T* ptAllocator);		
序号	参数类型	返回值说明
1	无	
序号	参数名	输入参数说明
1	ptAllocator	内存分配器结构指针,结构定义参见《NGN PPLAT SIP V1.01.10_接口说明书.doc》
序号	参数名	输出参数说明
算法及流程说明		
其他限制或说明		

### 10.6.2.3 SD-F-PPLAT-SIP-ALLOCATOR-0030 内存分配器分配内存接口

#### 10.6.2.3.1 P\_SIP\_ALLOCATOR\_allocate

函数名称		P_SIP_ALLOCATOR_allocate	类型（函数/宏）	功能函数
功能详细说明		内存分配器分配函数，从内存分配器管理的内存池中分配需要大小的内存，分配成功返回指向相应内存的指针，失败返回 NULL。		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
VOID* P_SIP_ALLOCATOR_allocate (IN PPLAT_SIP_ALLOCATOR_T *ptAllocate, IN WORD32 dwSize);				
序号	参数类型	返回值说明		
1	VOID*	成功，为分配的内存地址；失败，为 NULL；		
序号	参数名	输入参数说明		
1	ptAllocate	内存分配器结构指针,结构定义参见《NGN PPLAT SIP V1.01.10_接口说明书.doc》		
2	dwSize	需要分配的内存大小		
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
其他限制或说明				

#### 10.6.2.4 SD-F-PPLAT-SIP-ALLOCATOR-0040 内存分配器释放内存接口

##### 10.6.2.4.1 P\_SIP\_ALLOCATOR\_deallocate

函数名称		P_SIP_ALLOCATOR_deallocate	类型（函数/宏）	功能函数
功能详细说明		内存分配器内存释放函数，将从内存分配器中申请的内存释放，使对应的内存可以重新被分配。		
性能				
序号	分类	使用的全局及静态变量		
1	全局	无		
2	静态	无		
资源限制				
函数声明				
VOID P_SIP_ALLOCATOR_deallocate(IN PPLAT_SIP_ALLOCATOR_T *ptAllocator,IN VOID* ptr);				
序号	参数类型	返回值说明		
1	VOID	无		
序号	参数名	输入参数说明		
1	ptAllocator	内存分配器结构指针,结构定义参见《NGN PPLAT SIP V1.01.10_接口说明书.doc》		
2	ptr	指向需要分配的内存指针		
序号	参数名	输出参数说明		
1				
2				
3				
算法及流程说明				
其他限制或说明				

## 11 接口变更说明

注：修订记录编号 0007

本章说明 SIP 平台相关操作维护及 DBA 缺省实现，相关接口的变化及其对原有版本的兼容性考虑。

### 11.1 PPLAT\_SIP\_DB\_getLocalIPByDstIp 接口

该接口扩展一个参数 INOUT PPLAT\_SIP\_EXTERNAL\_INFO\_T \*ptExternalInfo，具体定义见接口说明。

对于 CSCF V4.03 项目需要将本端的 IP 地址，通过扩展信息携带下来，并指示此 IP 地址是否有效，在无效时，不根据此 IP 地址获取本端 IP-PORT。

对于 SSS2.0 项目及原有 CSCF 40 版本，由于该接口中没有使用扩展信息，因此对兼  
本文中的所有信息均为中兴通讯股份有限公司内部信息，不得向外传播。

容性没有影响。

## 11.2 PPLAT\_SIP\_TP\_UpdateExternalInfo 接口

新增接口，用于 TP 收到 TPM 或者其他接入设备发送过来的异步消息，进行扩展信息的更新，具体定义说明见接口说明。

对于 CSCF V4.03 项目，TP 实例收到消息后，需要更新扩展信息，将 TP 实例所在的 IP-PORT 添加到扩展信息中。

对于 SSS2.0 项目，由于不需要进行扩展信息的更新，此接口可以采用缺省实现，直接返回成功。

对于原有 CSCF40 版本，由于处理 TPM 发送的消息与收到 BRS 消息是不同的流程，因此对兼容性没有影响。

## 11.3 PPLAT\_SIP\_DB\_GetSubAccessAddr 接口

新增接口，用于获取后续请求消息的接入地址信息（注：发送地址与后续接入地址可以不同）。具体定义见接口说明。

对于 CSCF V4.03 项目，需要指定后续请求所要发送的端口（即本端的接入端口），通过该接口获取后续所要发送的端口，根据该信息进行添加、更改 record-route 等操作。

是否添加、更改 record-route 中的端口信息，可以通过在 PPLATAPP 中增加项目开关项进行控制，因此对 SSS2.0 项目及原有 CSCF 40 版本兼容性没有影响。

## 12 可挖掘的专利点

本次方案设计重点在于 SIP 平台移植的设计工作，基本没有新的需求和新增流程和技术，所以没有专利可挖掘；

## 13 参考文献

《NGN PPLAT V1.00.10\_系统方案》

《NGN PPLAT SIP V1.01.10\_接口说明书.doc》

《NGN PPLAT SIP V1.01.10\_软件需求说明书.doc》