协议栈百问

**更改历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **状态** | **日期** | **责任人** | **更改原因** |
| 1.0 | 草稿 | 2022-05-2 | liuyongqi (刘永琪) | 创建 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[Modem Develop Silver Bullet 1](#_Toc157516076)

[目录 2](#_Toc157516077)

[1 介绍 8](#_Toc157516078)

[1.1 文档目的 8](#_Toc157516079)

[1.2 术语和缩写 8](#_Toc157516080)

[1.2.1 术语 8](#_Toc157516081)

[1.3 参考文档 8](#_Toc157516082)

[2 ECM9100常见问题记录 9](#_Toc157516083)

[2.1 NAS 9](#_Toc157516084)

[2.1.1 手动搜网独占AT通道无法关机 9](#_Toc157516085)

[2.1.2 IPV6激活DNS未获取 9](#_Toc157516086)

[2.1.3 搜网过程 10](#_Toc157516087)

[2.1.4 AP频繁自动加载 10](#_Toc157516088)

[2.1.5 Service Request流程的结束 11](#_Toc157516089)

[2.2 RRC 11](#_Toc157516090)

[2.2.1 重选失败未将重选失败的小区BAR 300s 11](#_Toc157516091)

[3 协议流程学习 11](#_Toc157516092)

[3.1 ESM 11](#_Toc157516093)

[3.1.1 概述 11](#_Toc157516094)

[3.1.2 ESP流程的种类 12](#_Toc157516095)

[3.1.3 ESM子层的状态 12](#_Toc157516096)

[3.2 IP地址分配 13](#_Toc157516097)

[3.2.1 通过NAS信令分配IP地址 13](#_Toc157516098)

[3.2.2 网络分配IP的处理过程 13](#_Toc157516099)

[3.2.3 IP头压缩 14](#_Toc157516100)

[3.2.4 ESM流程的地址处理原则 14](#_Toc157516101)

[3.3 PLMN Related 16](#_Toc157516102)

[3.3.1 高优先级PLMN搜索 16](#_Toc157516103)

[3.3.2 FPLMN处理 17](#_Toc157516104)

[3.4 RRC 17](#_Toc157516105)

[3.4.1 RRC Connection Request 17](#_Toc157516106)

[4 编解码 17](#_Toc157516107)

[5 科普类 17](#_Toc157516108)

[5.1.1 什么是MS、UE、MT、TE、TA 17](#_Toc157516109)

[5.1.2 Feature Phone和Smart Phone 17](#_Toc157516110)

[5.1.3 什么是Non-IP 18](#_Toc157516111)

[5.1.4 NIDD的实现 18](#_Toc157516112)

[5.1.5 CIoT EPS optimizations 19](#_Toc157516113)

[5.1.6 传输模式 19](#_Toc157516114)

[5.1.7 RNTI 19](#_Toc157516115)

[5.1.8 什么是DCI 20](#_Toc157516116)

[5.2 借助tshark解码任何码流 20](#_Toc157516117)

[6 语句积累 21](#_Toc157516118)

[7 PDCP代码学习 21](#_Toc157516119)

[7.1 协议内容学习 21](#_Toc157516120)

[7.1.1 UPDCP处理流程 21](#_Toc157516121)

[7.1.2 DPDCP处理流程 22](#_Toc157516122)

[7.2 代码学习 22](#_Toc157516123)

[7.2.1 判断IP包为TCP的ACK包 22](#_Toc157516124)

[7.2.2 检测Ping包 23](#_Toc157516125)

[7.2.3 PDCP关键函数 24](#_Toc157516126)

[7.2.4 PDCP的配置过程 25](#_Toc157516127)

[7.2.5 内存相关的函数操作 26](#_Toc157516128)

[7.2.6 PDCP关键Log 26](#_Toc157516129)

[8 RLC 26](#_Toc157516130)

[8.1 RLC TM模式 26](#_Toc157516131)

[9 MAC 26](#_Toc157516132)

[9.1 MAC PDU 26](#_Toc157516133)

[9.1.1 MAC PDU的组成部分 26](#_Toc157516134)

[9.1.2 不同的MAC子头 27](#_Toc157516135)

[9.1.3 MAC PDU的结构 27](#_Toc157516136)

[9.1.4 参数 27](#_Toc157516137)

[9.2 MAC控制单元 28](#_Toc157516138)

[9.2.1 BSR 28](#_Toc157516139)

[9.2.2 C-RNTI MAC Control Element 29](#_Toc157516140)

[9.2.3 DRX Command MAC Control Element 29](#_Toc157516141)

[9.2.4 UE Contention Resolution Identity MAC Control Element 29](#_Toc157516142)

[9.2.5 Timing Advance Command MAC Control Element 29](#_Toc157516143)

[9.2.6 Power Headroom MAC Control Element 29](#_Toc157516144)

[9.2.7 MCH Scheduling Information MAC Control Element 29](#_Toc157516145)

[10 RRC 29](#_Toc157516146)

[10.1 关于SRB 29](#_Toc157516147)

[10.2 测量控制 29](#_Toc157516148)

[11 IPRELAY 30](#_Toc157516149)

[11.1 发送数据过程 30](#_Toc157516150)

[11.1.1 IPC发送数据处理 30](#_Toc157516151)

[11.1.2 AP\_DATA\_TASK的创建 30](#_Toc157516152)

[11.2 接收数据过程 30](#_Toc157516153)

[11.2.1 AP接收数据后的内存释放 30](#_Toc157516154)

[12 eCat专题 30](#_Toc157516155)

[12.1.1 Layer\_Event eCat Scan 30](#_Toc157516156)

[12.1.2 多行文本显示 31](#_Toc157516157)

[13 ECM9500代码结构 31](#_Toc157516158)

[13.1 基本的概念、缩写 31](#_Toc157516159)

[13.2 入口函数 31](#_Toc157516160)

[13.3 ECM9500视频学习 31](#_Toc157516161)

[13.4 Platform\_overview录屏学习 31](#_Toc157516162)

[13.4.1 ESWIN交付/工具/PCTOOL/MORU-PCtool-GUI.exe 32](#_Toc157516163)

[13.4.2 armshrine 32](#_Toc157516164)

[13.4.3 省电相关 33](#_Toc157516165)

[13.4.4 如何编译kernel 33](#_Toc157516166)

[13.5 Innofidei\_P4\_platform\_SW\_Interface文档 34](#_Toc157516167)

[13.5.1 MODULE MANAGEMENT 34](#_Toc157516168)

[13.5.2 MEMORY MANAGEMENT 34](#_Toc157516169)

[13.5.3 Logging 35](#_Toc157516170)

[13.5.4 Platform\_Overview 35](#_Toc157516171)

[13.6 DSP\_Overview录屏学习 LTE\_DSP\_ARCH\_Design 36](#_Toc157516172)

[13.6.1 各个目录的作用 36](#_Toc157516173)

[13.6.2 代码工程 36](#_Toc157516174)

[13.6.3 LTE PHY state 38](#_Toc157516175)

[13.6.4 Tsk\_PSS\_SSS.c 38](#_Toc157516176)

[13.6.5 Tsk\_PBCH.c 38](#_Toc157516177)

[13.6.6 DSP和L1C的通信两种方式 38](#_Toc157516178)

[13.6.7 DSP-ARM Interface 39](#_Toc157516179)

[13.6.8 DSP-LTE\_HW interface 39](#_Toc157516180)

[13.7 LTE\_DSP\_HW\_Interface\_P4A 40](#_Toc157516181)

[13.7.1 系统模块图 40](#_Toc157516182)

[13.7.2 DSP-FPGA之间的接口 40](#_Toc157516183)

[13.7.3 中断和总线连接 40](#_Toc157516184)

[13.7.4 寄存器定义 40](#_Toc157516185)

[13.8 LTE UE L1 Interface Design Specification for P4(v2) 41](#_Toc157516186)

[13.8.1 System Overview 41](#_Toc157516187)

[13.8.2 L1 Interface 42](#_Toc157516188)

[13.8.3 L1 Interface Procedure 42](#_Toc157516189)

[13.9 FS&CS 45](#_Toc157516190)

[13.10 Platform\_MemInfo\_Dump\_etc 45](#_Toc157516191)

[13.10.1 平台内存分配相关的函数接口 45](#_Toc157516192)

[13.10.2 内存/程序运营情况 47](#_Toc157516193)

[13.10.3 Note 48](#_Toc157516194)

[13.11 PS\_AT&USIM&SMS培训视频 49](#_Toc157516195)

[13.12 PS+PHY软件编译环境演示 50](#_Toc157516196)

[13.13 代码中的module 50](#_Toc157516197)

[13.13.1 模块的定义 50](#_Toc157516198)

[13.13.2 Module.h文件中的枚举变量 50](#_Toc157516199)

[13.14 流程整理 51](#_Toc157516200)

[13.15 NAS代码整理 51](#_Toc157516201)

[13.15.1 EMmSession 51](#_Toc157516202)

[14 LTE协议 51](#_Toc157516203)

[14.1 基本的概念 51](#_Toc157516204)

[14.1.1 UE mode of operation 51](#_Toc157516205)

[14.2 EMM 52](#_Toc157516206)

[14.2.1 概述 52](#_Toc157516207)

[14.2.2 术语记录 52](#_Toc157516208)

[14.2.3 EMM流程的类型 52](#_Toc157516209)

[14.2.4 EMM状态机 53](#_Toc157516210)

[14.2.5 EMM子层状态 53](#_Toc157516211)

[14.2.6 EPS 54](#_Toc157516212)

[14.3 Cell Search 55](#_Toc157516213)

[14.4 LTE PHY 55](#_Toc157516214)

[15 C/C++常见操作记录 55](#_Toc157516215)

[15.1 C++基础 55](#_Toc157516216)

[15.1.1 如何设置枚举类型的整型类型 55](#_Toc157516217)

[15.2 库函数 55](#_Toc157516218)

[15.2.1 realloc 55](#_Toc157516219)

[15.2.2 malloc\_usable\_size 56](#_Toc157516220)

[15.2.3 calloc 56](#_Toc157516221)

[15.2.4 srand/rand 56](#_Toc157516222)

[15.2.5 pcap\_loop 56](#_Toc157516223)

[15.2.6 getopt 56](#_Toc157516224)

[15.2.7 calloc 57](#_Toc157516225)

[15.2.8 strdup 57](#_Toc157516226)

[15.2.9 sprintf 57](#_Toc157516227)

[15.3 STL 57](#_Toc157516228)

[15.3.1 map 57](#_Toc157516229)

[15.3.2 hash\_map和map 57](#_Toc157516230)

[15.4 操作符相关 57](#_Toc157516231)

[15.4.1 #与## 57](#_Toc157516232)

[15.4.2 Enum 58](#_Toc157516233)

[15.4.3 类默认的属性类型 58](#_Toc157516234)

[15.4.4 operator 58](#_Toc157516235)

[15.4.5 sizeof 59](#_Toc157516236)

[15.4.6 do{}while(0) 59](#_Toc157516237)

[15.4.7 MemCheck实现的几个例子 59](#_Toc157516238)

[15.5 类相关 59](#_Toc157516239)

[15.5.1 C++关键字friend 59](#_Toc157516240)

[15.5.2 C++虚函数回顾 59](#_Toc157516241)

[15.5.3 类模板 60](#_Toc157516242)

[15.5.4 inline 60](#_Toc157516243)

[15.5.5 const 60](#_Toc157516244)

[15.6 内存操作 60](#_Toc157516245)

[15.7 线程 60](#_Toc157516246)

[15.7.1 线程锁 60](#_Toc157516247)

[15.8 文件系统 61](#_Toc157516248)

[15.8.1 FILE 61](#_Toc157516249)

[15.9 Makefile 62](#_Toc157516250)

[15.9.1 常见的变量 62](#_Toc157516251)

[15.9.2 链接符号 62](#_Toc157516252)

[15.9.3 make 62](#_Toc157516253)

[16 基本工具操作 62](#_Toc157516254)

[16.1 Win10常见操作 62](#_Toc157516255)

[16.1.1 快捷键 62](#_Toc157516256)

[16.1.2 Win10常见问题和解决方法 63](#_Toc157516257)

[16.2 VS Code操作 63](#_Toc157516258)

[16.2.1 VS Code快速添加注释 63](#_Toc157516259)

[16.2.2 快速打开终端 64](#_Toc157516260)

[16.2.3 快速格式化代码 64](#_Toc157516261)

[16.3 eCat工具 64](#_Toc157516262)

[16.3.1 eCat工程编译 64](#_Toc157516263)

[16.4 Eclipse/ZSP 64](#_Toc157516264)

[16.4.1 Eclipse中显示编译problem的方法： 64](#_Toc157516265)

[16.4.2 设置为UTF-8编码 64](#_Toc157516266)

[16.4.3 "IOConsole updater" has encounter a problem 64](#_Toc157516267)

[16.4.4 如何将多个.o文件连接为一个 65](#_Toc157516268)

[16.5 ZSP调试工具 65](#_Toc157516269)

[16.5.1 安装 65](#_Toc157516270)

[16.5.2 ZSP调试项目代码 66](#_Toc157516271)

[16.5.3 根据地址确定代码所在的行 67](#_Toc157516272)

[16.5.4 地址写入断点的设置 67](#_Toc157516273)

[16.5.5 启动调试停在\_start位置 68](#_Toc157516274)

[16.5.6 BSS段设置为0的操作 68](#_Toc157516275)

[16.5.7 ZSP打断点的方法 68](#_Toc157516276)

[16.6 梳理FreeRTOS内存管理流程 70](#_Toc157516277)

[16.6.1 关键的类型/变量 70](#_Toc157516278)

[16.7 GCC 73](#_Toc157516279)

[16.7.1 常见的报错 73](#_Toc157516280)

[16.8 Visual Studio 74](#_Toc157516281)

[16.8.1 Visual Studio预定义宏 74](#_Toc157516282)

[16.8.2 Visual Studio常见报错 76](#_Toc157516283)

[16.8.3 Visual Studio常见问题解决 77](#_Toc157516284)

[16.9 Source Insight 77](#_Toc157516285)

[16.9.1 常用快捷键 77](#_Toc157516286)

[16.9.2 将函数名的字体设置为大号字体，加粗，醒目 77](#_Toc157516287)

[16.9.3 如何在SI IDE中设置一个全局变量 77](#_Toc157516288)

[16.9.4 指定目录执行搜索 78](#_Toc157516289)

[16.10 Excel 78](#_Toc157516290)

[16.10.1 常见问题及对应的解决方法 78](#_Toc157516291)

[16.11 Git 79](#_Toc157516292)

[16.11.1 Git原理 79](#_Toc157516293)

[16.11.2 Git基本操作 79](#_Toc157516294)

[16.11.3 常见问题 79](#_Toc157516295)

[16.12 Jenkins 81](#_Toc157516296)

[16.12.1 Jenkins安装注意事项 81](#_Toc157516297)

[16.12.2 Jenkins邮件无法发送 83](#_Toc157516298)

[16.13 测试工具使用 85](#_Toc157516299)

[16.13.1 RIDE自动化测试工具 85](#_Toc157516300)

[16.14 MAP文件分析 86](#_Toc157516301)

[16.14.1 基本概念 86](#_Toc157516302)

[16.15 PlantUML 86](#_Toc157516303)

[16.15.1 给箭头修改颜色 86](#_Toc157516304)

[16.15.2 文本对齐 87](#_Toc157516305)

[16.15.3 改变箭头的样式 87](#_Toc157516306)

[16.15.4 消息序列的自动编号 87](#_Toc157516307)

[16.15.5 添加标题 87](#_Toc157516308)

[16.15.6 分隔示意图 87](#_Toc157516309)

[16.15.7 组合消息 87](#_Toc157516310)

[16.15.8 注释信息 88](#_Toc157516311)

[16.15.9 分隔符 88](#_Toc157516312)

[16.15.10 引用 88](#_Toc157516313)

[16.15.11 延迟 89](#_Toc157516314)

[16.15.12 文本换行 89](#_Toc157516315)

[16.15.13 空间 89](#_Toc157516316)

[16.15.14 生命线的激活与撤销 89](#_Toc157516317)

[16.15.15 锚点和持续时间 90](#_Toc157516318)

[17 芯片公司 90](#_Toc157516319)

[17.1 半导体 90](#_Toc157516320)

[17.1.1 芯科科技 Silicon Lab 90](#_Toc157516321)

[17.1.2 ADI 90](#_Toc157516322)

[17.1.3 ASTRI 90](#_Toc157516323)

[17.1.4 创毅视讯 Innofidei 90](#_Toc157516324)

[17.1.5 瀚讯 90](#_Toc157516325)

[17.1.6 芯原微电子（上海）股份有限公司 90](#_Toc157516326)

[17.1.7 移芯通信 91](#_Toc157516327)

[17.1.8 芯翼 92](#_Toc157516328)

[17.1.9 翱捷ASR 92](#_Toc157516329)

[17.1.10 智联安 92](#_Toc157516330)

[17.1.11 瑞普康 92](#_Toc157516331)

[17.1.12 创芯汇联 92](#_Toc157516332)

[17.1.13 芯晟 92](#_Toc157516333)

[17.1.14 星思 92](#_Toc157516334)

[17.1.15 宇龙 92](#_Toc157516335)

[17.2 模组厂商 92](#_Toc157516336)

[17.2.1 广和通 92](#_Toc157516337)

[17.3 竞品分析 92](#_Toc157516338)

[17.3.1 EC618供参考 92](#_Toc157516339)

[18 周报 92](#_Toc157516340)

[18.1 2022年-5月 92](#_Toc157516341)

[18.1.1 Weekly 20 – 五月第二周 92](#_Toc157516342)

[18.1.2 Weekly21 – 五月第三周 92](#_Toc157516343)

[19 MORU原始平台代码学习 93](#_Toc157516344)

[19.1 makefile 93](#_Toc157516345)

[20 PS代码平台相关 93](#_Toc157516346)

[20.1 消息和模块 93](#_Toc157516347)

[20.2 AT命令相关模块 93](#_Toc157516348)

[20.2.1 MODULE\_ATCMD 93](#_Toc157516349)

[20.2.2 MODULE\_ATOUTPUT 93](#_Toc157516350)

[20.2.3 MODULE\_ATMT 93](#_Toc157516351)

[20.2.4 MODULE\_ATCMDLINE 93](#_Toc157516352)

[21 SMS模块学习 93](#_Toc157516353)

[21.1 SMS模块中的代码文件 93](#_Toc157516354)

[21.2 关键类定义 94](#_Toc157516355)

[21.2.1 SmsCommonConfig 94](#_Toc157516356)

[21.2.2 SmsMain.cpp文件 94](#_Toc157516357)

[21.3 函数接口定义 94](#_Toc157516358)

[21.4 SMS消息处理 95](#_Toc157516359)

[21.4.1 配置类消息的处理函数 95](#_Toc157516360)

[21.4.2 处理发送接收短信相关的消息 95](#_Toc157516361)

[21.5 配置类结构体说明 95](#_Toc157516362)

[21.6 收发短信相关结构体说明 95](#_Toc157516363)

[21.7 ATMT与SMS消息交互 95](#_Toc157516364)

[21.7.1 短信格式设置：AT+CMGF 95](#_Toc157516365)

[21.7.2 SMS发送给ATMT的消息 100](#_Toc157516366)

[22 内存泄漏统计工具 101](#_Toc157516367)

[23 AT命令 101](#_Toc157516368)

[23.1 代码中的AT命令 101](#_Toc157516369)

[23.1.1 关键结构体 101](#_Toc157516370)

[23.1.2 AT命令解析 102](#_Toc157516371)

[23.1.3 MtMain中的函数接口 103](#_Toc157516372)

[23.2 常见AT命令整理 104](#_Toc157516373)

[23.3 Weekly 22 – 五月第四周 104](#_Toc157516374)

[24 3GPP相关 106](#_Toc157516375)

[24.1 3GPP各个Release之间的功能差别 106](#_Toc157516376)

[25 RRC代码学习 106](#_Toc157516377)

[25.1 RRC信令无线承载 106](#_Toc157516378)

[25.2 各个逻辑信道上的RRC消息 107](#_Toc157516379)

[25.2.1 BCCH-BCH-Message 107](#_Toc157516380)

[*25.2.2* *BCCH-DL-SCH-Message* 107](#_Toc157516381)

[*25.2.3* *PCCH-Message* 107](#_Toc157516382)

[*25.2.4* *DL-CCCH-Message* 107](#_Toc157516383)

[25.2.5 *DL-DCCH-Message* 107](#_Toc157516384)

[25.2.6 *UL-CCCH-Message* 107](#_Toc157516385)

[*25.2.7* *UL-DCCH-Message* 107](#_Toc157516386)

[25.3 常用的常量定义 107](#_Toc157516387)

[25.3.1 RRC状态机相关 107](#_Toc157516388)

[25.4 RRC常用函数 108](#_Toc157516389)

[25.4.1 常用宏函数 108](#_Toc157516390)

[26 代码空间压缩 109](#_Toc157516391)

[26.1 各个内存分区的含义 109](#_Toc157516392)

[26.2 ESM部分 109](#_Toc157516393)

[26.2.1 定义部分 109](#_Toc157516394)

[27 Miscellaneous 109](#_Toc157516395)

[27.1 时钟同步方式 109](#_Toc157516396)

[27.2 AMT 110](#_Toc157516397)

[27.2.1 AGC 110](#_Toc157516398)

[27.2.2 AGF 110](#_Toc157516399)

[27.2.3 APC 110](#_Toc157516400)

[27.2.4 LOLeakage 110](#_Toc157516401)

[27.2.5 ACLR 110](#_Toc157516402)

[28 RTC功能 110](#_Toc157516403)

[28.1 CP侧与系统时间相关的流程 111](#_Toc157516404)

[28.2 NITZ 111](#_Toc157516405)

[28.3 NTP 111](#_Toc157516406)

[29 Linux常用操作 111](#_Toc157516407)

[29.1 命令行操作 111](#_Toc157516408)

[29.1.1 Ctrl+U 将光标前面的内容删掉。 112](#_Toc157516409)

[29.1.2 Ctrl+K 将光标后面的内容删掉。 112](#_Toc157516410)

[29.1.3 Ctrl+A 光标移动到命令行开始处。 112](#_Toc157516411)

[29.1.4 Ctrl+E 光标移动到命令行尾部。 112](#_Toc157516412)

[29.1.5 Ctrl+W 删除前面的单词 112](#_Toc157516413)

[29.1.6 Ctrl+Y 恢复删除前的命令 112](#_Toc157516414)

[29.1.7 Ctrl+P 上一条命令 112](#_Toc157516415)

[29.1.8 Ctrl+B 向左移动一个字符 112](#_Toc157516416)

[29.1.9 Ctrl+F 向右移动一个字符 112](#_Toc157516417)

[29.1.10 Ctrl+R 历史命令 112](#_Toc157516418)

[29.1.11 Ctrl+D 删除光标右方的字符 112](#_Toc157516419)

[29.1.12 Ctrl+L 清屏 112](#_Toc157516420)

[29.1.13 Ctrl+M Enter 112](#_Toc157516421)

[29.1.14 Ctrl+ 左箭头 向左移动一个单词 112](#_Toc157516422)

[29.1.15 Ctrl+ 右箭头 向右移动一个单词 112](#_Toc157516423)

[29.2 常用命令 112](#_Toc157516424)

[29.2.1 tee 112](#_Toc157516425)

[29.2.2 script 113](#_Toc157516426)

[29.2.3 ldd 113](#_Toc157516427)

[29.2.4 readelf 113](#_Toc157516428)

[29.2.5 file 113](#_Toc157516429)

[29.2.6 cat 113](#_Toc157516430)

[29.2.7 tac 113](#_Toc157516431)

[29.2.8 more 113](#_Toc157516432)

[29.2.9 less 113](#_Toc157516433)

[29.2.10 scp 113](#_Toc157516434)

[29.3 Linux常见问题 114](#_Toc157516435)

[29.3.1 Ubuntu中的时区不对，显示时间与北京时间相差8小时 114](#_Toc157516436)

[30 python常见操作 114](#_Toc157516437)

[30.1 常见函数使用 114](#_Toc157516438)

[30.1.1 range()函数的使用 114](#_Toc157516439)

[30.1.2 round() 114](#_Toc157516440)

[30.2 列表 114](#_Toc157516441)

[30.2.1 列表的遍历 114](#_Toc157516442)

[30.2.2 列表的增删 115](#_Toc157516443)

[30.3 时间 115](#_Toc157516444)

[30.4 文件 115](#_Toc157516445)

[30.4.1 获取指定文件的绝对路径 115](#_Toc157516446)

[30.4.2 当前目录切换 115](#_Toc157516447)

[30.4.3 文件重命名 115](#_Toc157516448)

[31 协议重点内容 115](#_Toc157516449)

[32 Web学习 115](#_Toc157516450)

[32.1 ChatGPT的相关提问回答 115](#_Toc157516451)

[32.1.1 关于JavaScript 115](#_Toc157516452)

[32.1.2 Web前端和后端在Web应用程序中承担不同的任务 116](#_Toc157516453)

[32.1.3 Web后端技术 116](#_Toc157516454)

[32.1.4 Web相关的技术有哪些 116](#_Toc157516455)

[32.1.5 Web学习步骤 116](#_Toc157516456)

[32.2 HTML常见问题 117](#_Toc157516457)

[33 工具设计 117](#_Toc157516458)

[33.1 Map分析工具 117](#_Toc157516459)

[33.1.1 需求 117](#_Toc157516460)

[33.2 eCat Log工具分析 117](#_Toc157516461)

[33.2.1 DisplayText 117](#_Toc157516462)

[33.3 3gppdecoder 117](#_Toc157516463)

[33.4 SDR LTE 117](#_Toc157516464)

# 介绍

## 文档目的

本文档用于记录新人入职奕斯伟期间各种账号申请的方法和流程，以及一些常用资料的获取说明、常用工具的安装和配置，旨在加快新人的办公环境和开发环境的搭建，避开一些常见的问题，提升效率。

## 术语和缩写

### 术语

1. ESWIN：E-solution to win
2. OA：Office Automation，办公自动化
3. ETX：Exceed TurboX，一款远程连接到windows/Linux的工具。
4. 3GPP：3rd Generation Partnership Project，第三代合作伙伴计划
5. ETSI：European Telecommunications Standards Institute，欧洲电信标准化协会
6. JIRA：JIRA是Atlassian公司出品的项目与事务跟踪工具，被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域。
7. Confluence：一个专业的企业知识管理与协同软件，也可以**用于构建企业**[**wiki**](https://baike.baidu.com/item/wiki/97755)。使用简单，但它强大的编辑和站点管理特征能够帮助团队成员之间共享信息、文档协作、集体讨论，信息推送。

## 参考文档

1.《上海新员工Note.docx》康国法 丁会彦

# ECM9100常见问题记录

# ECM9100常见问题记录

## NAS

### 手动搜网独占AT通道无法关机

问题：[ECM9100-2446](http://jira.eswincomputing.com/browse/ECM9100-2446)【Modem测试】【MPW\_功能\_协议栈】cops=?期间，所有AT命令不能处理，需要支持关机命令

分析：AT是串行执行，前一个命令不返回，后一个命令无法执行。设计原因，暂不修改。

建议：可以把AT+COPS=?做成一个不阻塞的命令。这个期间可以执行一些与驻网无关的命令，如关机等。

### IPV6激活DNS未获取

参考对比机高通平台的PDN Connectivity Request消息设置不同IP类型的PCO即可。具体设置参考如下章节。

目前设置请求MTU网络并不会返回，因此，可以不设置此IE。

#### IPV4

IPCP、DNS Server IPv4 Address、IP Address allocation via NAS signaling、NWK Req Bearer Control indicator、IPv4 Link MTU Request（网络不回复此请求，可以忽略）

Note：

1. 在IPV4/IPV4V6类型的请求中，一定要带IPCP。
2. IP Address allocation via NAS signaling、NWK Req Bearer Control indicator这两个域也是必带的。

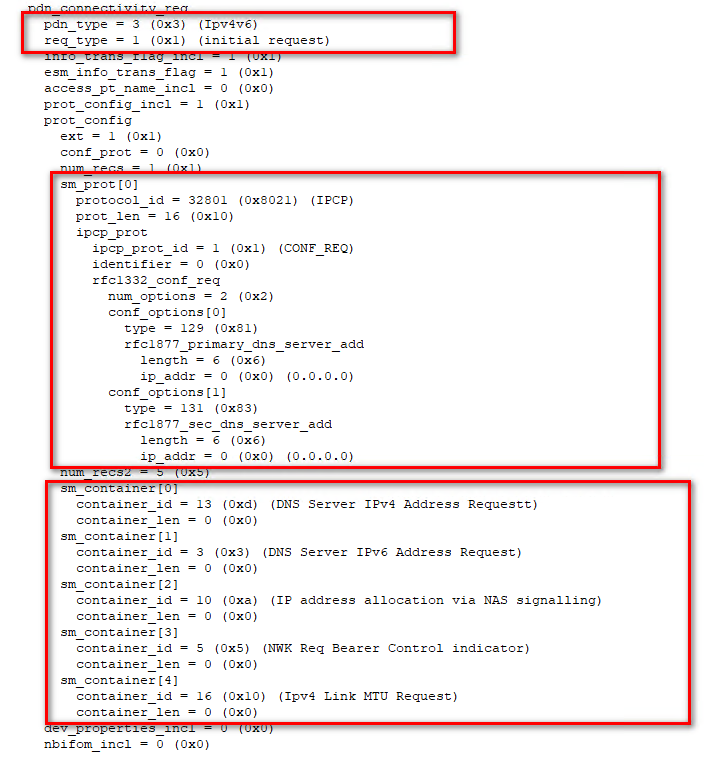


#### IPV4V6

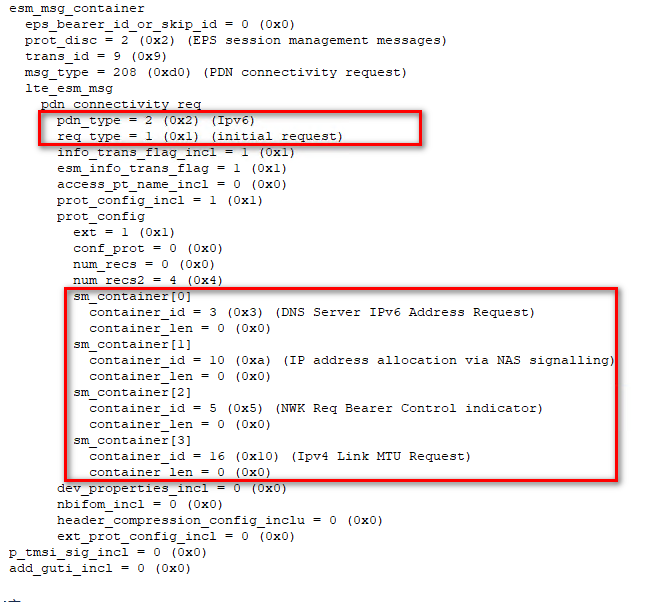
IPCP、DNS Server IPv4 Address、IP Address allocation via NAS signaling、NWK Req Bearer Control indicator、IPv4 Link MTU Request（网络不回复此请求，可以忽略）

Note：

同IPV4中的说明。



#### IPV6



### 搜网过程

#### 自动搜网过程

1. RPLMN、EHPLMN、HPLMN、EPLMN搜索，执行全BandScan
2. Step1失败，Search Available PLMN，执行各个自带的搜索，搜到了就上报，EMM根据PLMN的优先顺序选择驻留

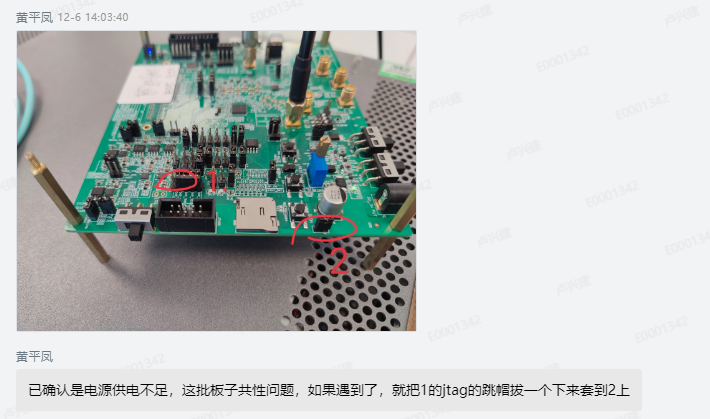
#### 手动搜网过程

Search Available PLMN，设置搜网类型为MANUAL，所有频点搜完后上报所有搜到的PLMN。

### AP频繁自动加载

问题原因：供电不足

处理方法：在如下图中接上跳线帽



### Service Request流程的结束

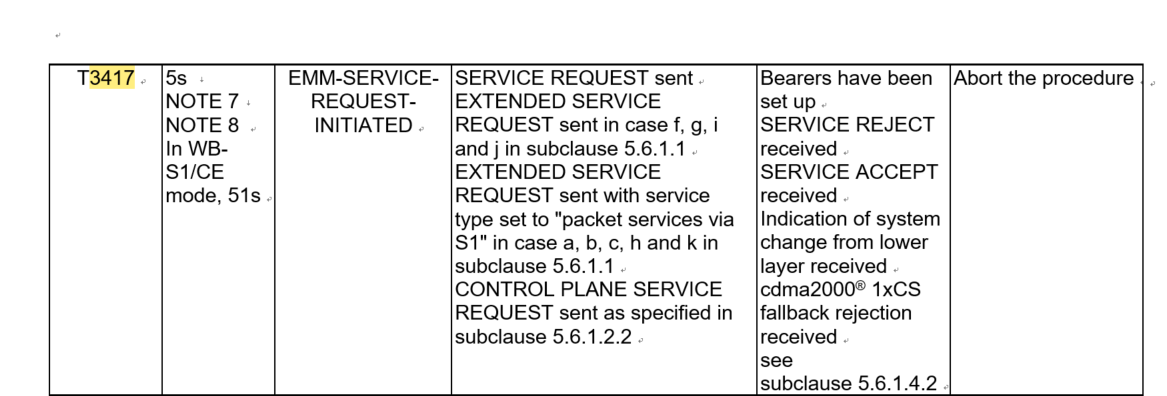
EMM发起Service Request流程后，启动T3417定时器（5s）

发起的原因：

1. 网络有下行信令需要发送（同过Paging消息触发SR）
2. UE有上行信令发送
3. UE或网络有用户数据发送并且当前处于EMM-IDLE状态

Service Request流程的结束：

1. T3417定时器超时，中止Service Request流程
2. Bearers更新了（通过RRCConnectionReconfiguration更新）
3. 收到SERVICE REJECT消息
4. 收到SERVICE ACCEPT消息



## RRC

### 重选失败未将重选失败的小区BAR 300s

[ECM9100-2497](http://jira.eswincomputing.com/browse/ECM9100-2497) 【Modem测试】【MPW\_功能\_协议栈】重选到非HPMN/EHPLMN的小区，驻留失败，没有将该频点加入到bar list中300s

根据协议36304 5.2.4.4节如下字段：

If the highest ranked cell or best cell according to absolute priority reselection rules is an intra-frequency or interfrequency cell which is not suitable for a CN type due to being part of the "list of forbidden TAs for roaming" or belonging to a PLMN which is not indicated as being equivalent to the registered PLMN, the UE shall not consider this cell and other cells on the same frequency, as candidates for reselection for the CN type for a maximum of 300s. If the UE enters into state any cell selection, any limitation shall be removed. If the UE is redirected under E-UTRAN control to a frequency for which the timer is running, any limitation on that frequency shall be removed.

### 在什么情况下，RRC将一个频点添加到BAR列表中

1. SIB1中设置了CellBarred，且设置了同频重选不允许，则将整个频点Bar掉；
2. 重选中选择的小区的PLMN与当前注册的PLMN不是EPLMN的关系，则Bar 300s；

### Modification period

SIB2消息中的RadioResourceConfigCommonSIB（*RadioResourceConfigCommon类型*）

|  |
| --- |
| BCCH-Config ::= SEQUENCE {  modificationPeriodCoeff ENUMERATED {n2, n4, n8, n16}  }  PCCH-Config ::= SEQUENCE {  defaultPagingCycle ENUMERATED {  rf32, rf64, rf128, rf256},  nB ENUMERATED {  fourT, twoT, oneT, halfT, quarterT, oneEighthT,  oneSixteenthT, oneThirtySecondT}  } |

修改周期的计算为：

|  |
| --- |
| ***modificationPeriodCoeff***  Actual modification period, expressed in number of radio frames= *modificationPeriodCoeff* \* *defaultPagingCycle*. n2 corresponds to value 2, n4 corresponds to value 4, n8 corresponds to value 8 and n16 corresponds to value 16. |

modification period = *modificationPeriodCoeff* \* *defaultPagingCycle，单位为无线帧（10ms）*

**问题：如果下发Paging指示系统消息修改，则UE至多在一个修改周期之后重新接收新的系统消息！**



图 1 系统消息的修改

重点内容：

1. 修改周期（modification period）：在一个修改周期内，系统消息可以多次发送，但内容一致，下一个修改周期，网络可以修改系统消息的内容；
2. 修改周期边界（modification period boundaries）：SFN mod m = 0，其中m为修改周期的长度；
3. UE收到Paging中的系统消息修改通知（*systemInfoModification*）后，将在下一个修改周期开始时立即获取系统消息；
4. UE在获取到新的系统消息前使用之前获取到的系统消息；
5. 修改周期的计算方法：*modification period = modificationPeriodCoeff* \* *defaultPagingCycle*，单位为无线帧（10ms），均在SIB2中；
6. SIB1消息中的*systemInfoValueTag指示系统消息中是否发生了变化；*
7. UE在确认系统消息有效3小时后，应认为存储的系统消息无效；

### 加解密

#### 加解密的线程安全问题

目前RRC和EMM都使用硬件执行消息的完保和加解密，RRC和EMM属于两个线程，调用的函数为同一个，且使用了同一个全局变量。

S8 SetKey\_And\_Clock\_Switch(U32 algorithmId, U32 keyTypeIndex, U8\* key, U32 keyLength)

void cipher\_open\_cipher\_clock(void)

U16 cipher\_disable\_clock\_timer\_exp(void)

需要优化的点：

1. 多线程访问同一个全局变量可能遇到的问题

## 数据业务

### IPRELAY有上行数据，DRB状态为激活状态，但数据发送一直不成功，频繁触发RRC建链

结论：

IPRELAY发送RB\_CONN\_REQ请求恢复用户名的DRB资源时，若收到了EMM返回的结果为FAIL，则启动一个5s的定时器，5s内不再尝试，5s后若存在pending的上行数据则再发起Service Request流程。

[ECM9100-3554](http://jira.eswincomputing.com/browse/ECM9100-3554)【Modem测试】【FMASK\_功能\_协议栈】21:34:29左右，CP一直在反复RRC REQUEST

|  |
| --- |
| 灌包过程中IPRELAY持续不断地接收AP的上行数据，若此时DRB不处于激活状态，则IPRELAY将触发EMM发起Service Request流程。  这个过程中EMM触发RRC发起RRC CONNECTION REQUEST,但是此时  **459670 1 IPRELAY EMM 1-6-528-4 24.06.23-21:34:22.603 IPRELAY\_EMM\_DATA\_RB\_CONN\_REQ** 459681 1 EMM ERRC 1-6-528-4 24.06.23-21:34:22.604 EMM\_ERRC\_CONNECTION\_REQ 459684 9892 INFO EMM 1-6-528-4 24.06.23-21:34:22.604 EMM State Changed, **EMM\_REGISTERED->EMM\_SERVICE\_REQUEST\_INITIATED** 460379 1 EL1C ERRC 1-6-543-0 24.06.23-21:34:22.715 L1C\_RRC\_CAMP\_ON\_CELL\_CONFIRM **460447 1 ERRC ERRC 1-6-543-2 24.06.23-21:34:22.715 EMM\_ERRC\_CONNECTION\_REQ** 460561 2 UE ENB 1-6-543-4 24.06.23-21:34:22.758 (1550,18) RRC CONNECTION REQUEST 460562 1 ERRC EURLC 1-6-543-4 24.06.23-21:34:22.758 RRC\_URLC\_TM\_AIR\_SIG\_IND 460622 1 EL1C PAL 1-6-543-6 24.06.23-21:34:22.759 L1C\_DSP\_PRACH\_CONTROL\_REQUEST **461554 1 EL1C EUMAC 1-6-557-9 24.06.23-21:34:22.948 L1C\_UMAC\_RACH\_FAILURE\_IND** 461635 1 EL1C ERRC 1-6-558-2 24.06.23-21:34:22.948 L1C\_RRC\_RELEASE\_CONFIRM    **// RRC本地释放连接，通知EMM进入空闲态，EMM中止当前的Service Request流程** **461666 1 ERRC EMM 1-6-558-2 24.06.23-21:34:22.949 ERRC\_EMM\_RRC\_CONNECTION\_STATUS\_UPD** 461671 9450 INFO EMM 1-6-558-2 24.06.23-21:34:22.949 EMM:MSG\_ID=ERRC\_EMM\_RRC\_CONNECTION\_STATUS\_UPD, CURR\_STATE=EMM\_SERVICE\_REQUEST\_INITIATED, CURR\_SUBSTATE=EMM\_REGISTERED\_NORMAL\_SERVICE, PLMNSelectMode=AUTOMATIC, EUStatus=EU1\_UPDATED, EmmMode=EMM\_IDLE 461673 9365 INFO EMM 1-6-558-2 24.06.23-21:34:22.949 EMM Connect Mode change, EMM\_IDLE->EMM\_IDLE. 461674 9797 INFO EMM 1-6-558-2 24.06.23-21:34:22.949 Lower layer failure while in Service Request, abort Service Request **461679 9796 INFO EMM 1-6-558-2 24.06.23-21:34:22.949 Service Request Procedure has been aborted!**    **// IPRELAY收到RB\_CONN\_REQ的响应后，清除了Pending的状态，此时有数据下发将再次触发Service Request，在低层看到的现象就是不停地发送RRCConnectionRequest和尝试接入。**  461680 1 EMM IPRELAY 1-6-558-2 24.06.23-21:34:22.949 EMM\_IPRELAY\_DATA\_RB\_CONN\_CFM **461683 1 IPRELAY EMM 1-6-558-2 24.06.23-21:34:22.949 IPRELAY\_EMM\_DATA\_RB\_CONN\_REQ** 461684 9450 INFO EMM 1-6-558-2 24.06.23-21:34:22.949 EMM:MSG\_ID=IPRELAY\_EMM\_DATA\_RB\_CONN\_REQ, CURR\_STATE=EMM\_REGISTERED, CURR\_SUBSTATE=EMM\_REGISTERED\_NORMAL\_SERVICE, PLMNSelectMode=AUTOMATIC, EUStatus=EU1\_UPDATED, EmmMode=EMM\_IDLE 461694 1 EMM ERRC 1-6-558-2 24.06.23-21:34:22.949 EMM\_ERRC\_CONNECTION\_REQ    修改方案：  **IPRELAY发送RB\_CONN\_REQ请求恢复用户名的DRB资源时，若收到了EMM返回的结果为FAIL，则启动一个5s的定时器，5s内不再尝试，5s后若存在pending的上行数据则再发起Service Request流程。** |

# 协议流程学习

以问答的形式来做协议流程学习

## 基本的概念

Combined attach

### UE mode of operation

参考：TS 24.301 4.3 UE mode of operation

1. PS mode 1 of operation: the UE registers only to EPS services, and UE's usage setting is "**voice centric**";
2. PS mode 2 of operation: the UE registers only to EPS services, and UE's usage setting is "**data centric**";
3. CS/PS mode 1 of operation: the UE registers to both EPS and **non-EPS services**, and UE's usage setting is "**voice centric**"; and
4. CS/PS mode 2 of operation: the UE registers to both EPS and **non-EPS services**, and UE's usage setting is "**data centric**".

配置为CS fallback的UE应当运行在CS/PS mode1或者CS/PS mode2，这类的UE也可以配置为使用IMS，在这种情况下将会使用到TS 24.167中定义的**语音域优先级**用于**选择发起语音通信服务的域选择**。

Note：尝试CS紧急呼叫时将忽略发起语音通信服务的域选择。

一旦上层请求建立电路域紧急呼叫：

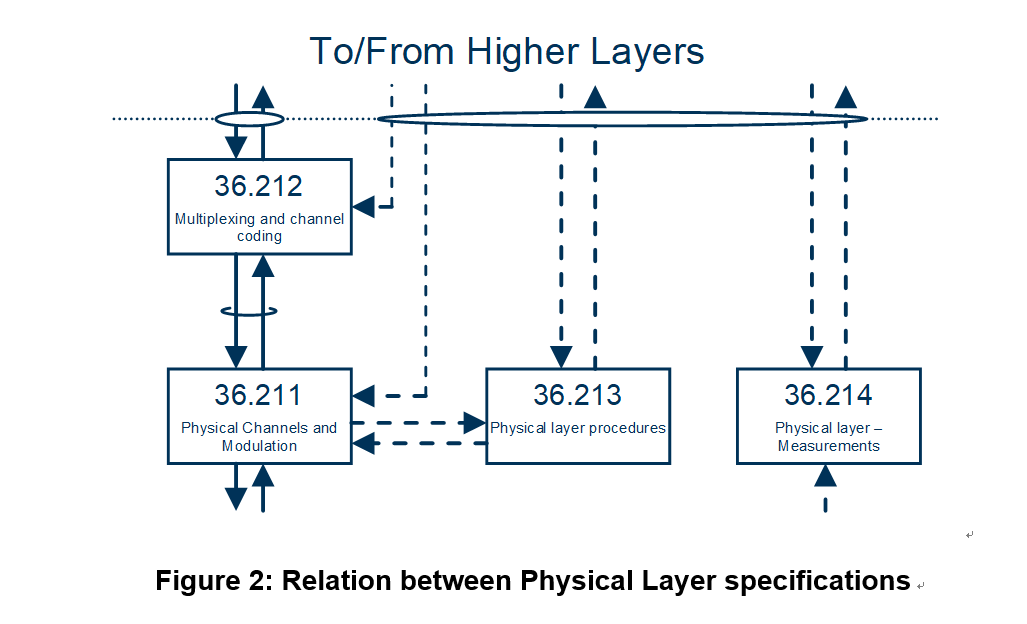
1. 如果UE需要发起一个CSFB紧急呼叫，但是无法执行CSFB，则UE应当尝试选择GERAN或者UTRAN RAT，并且“IMS voice not available”的UE应当禁止E-UTRA能力来允许潜在的会掉（应该是防止在CSFB期间返回到LTE），并处理CS紧急呼叫的建立。
2. 如果UE需要发起1XCS FB紧急呼叫，但是无法执行1xCSFB，则UE应当尝试选择CDMA2000 1X RAT来建立呼叫。

配置SMS over SGs的UE应当运行在CS/PS mode 1或者CS/PS mode 2

5.5.1.3 Combined attach procedure for EPS services and non-EPS services (S1 mode only)

## LTE PHY

参考文档：TS 36.201



### Cell Search

LTE小区搜索过程

PSS Peak Search， Timing Location

LTE的帧结构

## EMM

### 概述

EMM：EPS Mobility Management，EPS移动性管理。

EMM是LTE NAS中的一个子层，**用于支持UE的移动性**，例如通知网络其当前位置并提供用户身份机密性。

EMM的另外的功能：提供SM子层与连接管理子层（CM）的SMS实体之间的**连接管理**。

EMM流程**只能在NAS信令连接建立的基础上执行**，否则EMM需要发起NAS信令连接的建立。

EMM-IDLE mode：UE与网络之间不存在NAS信令连接，或者收到了低层RRC连接中止的指示。

EMM-CONNECTED mode：UE与网络之间已经建立了NAS信令连接。

### 术语记录

#### S101 mode

Qouted from 24.301 5.3.1.1

In S101 mode, when the cdma2000® HRPD access network resources are available for tunnelled NAS signalling, the UE shall enter EMM-CONNECTED mode and consider the S101 mode NAS signalling connection established.

UE通过CDMA 2000 HRPD接入网络通过隧道传输NAS信令接入到EPC的模式。

### EMM流程的类型

基于流程的发起方式，EMM流程分为如下三种类型：

1. EMM common procedure
2. EMM specific procedure
3. EMM connection management procedures（S1 mode only）

#### EMM common procedure

始终可以在存在NAS信令连接的同时**启动EMM通用过程**。这类程序包括:

由网络发起的流程：

1. GUTI reallocation
2. authentication
3. security mode control
4. identification
5. EMM information

#### EMM specific procedure

任何时候**只能运行一个终端启动的EMM特定流程**。这类程序包括:

1. 为了获取EPS服务/非EPS服务，UE在网络中发起IMSI附着，并根据UE请求建立EMM上下文和一个默认承载。

attach and combined attach

1. 为了获取紧急服务，UE发起IMSI或者IMEI附着，建立一个EMM上下文和默认的PDN承载用于提供紧急承载服务。

attach

1. 由UE或者网络发起的，用于去附着网络中EPS服务或者非EPS服务的IMSI，释放EMM上下文和所有的承载（若存在）。

detach and combined detach

1. UE发起的，用于去附着网络中EPS服务或者非EPS服务的IMSI，释放EMM上下文和所有的承载（若存在）。

eCall inactivity procedure

1. EMM上下文存在的情况下UE发起的流程

normal tracking area updating and combined tracking area updating(S1 mode only)

periodic tracking area updating(S1 mode only)

TAU流程可以用于请求发送用户数据的资源预留。

#### EMM connection management procedures（S1 mode only）

1. 由UE发起的用于建立与网络之间的安全连接或者请求发送数据的资源预留，或者二者同时兼备：

service request

**service ruqust流程只能在没有UE发起的EMM specific procedure进行中时，才可以发起！🡪 优先级低于EMM特定流程！**

1. 由网络发起用于请求建立一个NAS信令连接或者促使UE重新附着（若网络失败导致）

paging procedure

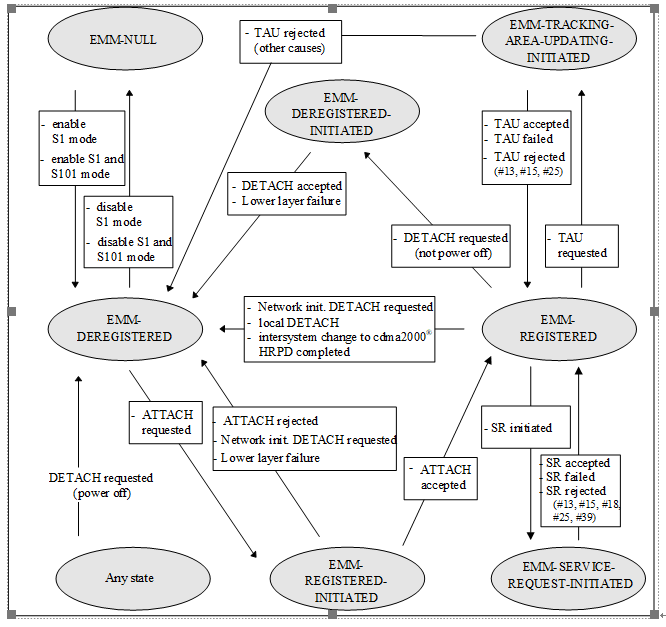
1. 由UE或者网络发起用于传输NAS消息

transport of NAS messages

generic transport of NAS messages

**NAS消息的传输和NAS消息的通用传输无法在EMM specific procedure或者service request流程进行中执行！ 🡪 优先级最低！**

### EMM状态机

****

### EMM子层状态

UE的行为定义在5.2，异常情况定义在5.4,5.5,5.6和5.7

#### 主状态

##### EMM-NULL

EPS服务在UE中被禁止。在此状态下，没有EPS移动性管理功能执行。

##### EMM-DEREGISTERED

此状态下，没有EMM上下文建立并且MME对于UE的位置无法获知因为UE不可达。为了建立EMM上下文，UE应当发起attach或者combined attach流程。

##### EMM-REGISTERED-INITIATED

当UE发起attach或者combined attach流程并且**处于等待MME的响应消息过程中**时，UE进入到此状态。

##### EMM-REGISTERED

在此状态下，UE的EMM上下文已经建立成功，并且默认承载已经激活：

-若UE或者MME不支持EMM-REGISTERED without PDN connection，或者UE或者MME支持EMM-REGISTERED without PDN connection，UE已经请求PDN连接并且建立了默认的EPS承载上下文。

EMM-REGISTERED状态中包含两个模式：EMM-IDLE和EMM-CONNECTED。

在EMM-IDLE mode，MME能够知道UE的位置，精度为包含了某些数量跟踪区的跟踪区列表。

在EMM-CONNECTED mode，MME获取UE的位置，精度为当前服务的eNodeB。

在EMM-REGISTERED状态下，UE可以发起发送和接收用户数据、信令信息或者响应Paging。另外，也可执行TAU或者combined TAU流程。

##### EMM-DEREGISTERED-INITIATED

UE请求释放EMM上下文发起detach或者combined detach流程并且处于等待网络MME响应的过程中时，UE进入到此状态。

##### EMM-TRACKING-AREA-UPDATING-INITIATED

UE发起TAU或者combined TAU流程并且处于等待MME响应的过程中时，UE进入到此状态。

##### EMM-SERVICE-REQUEST-INITIATED

UE发起service request流程并且处于等待MME响应的过程时，UE进入到此状态。

##### 各个EMM子状态的状态流转图



#### EMM-DEREGISTERED的子状态

EMM-DEREGISTERED状态中有分为多个子状态，除了子状态EMM-DEREGISTERED.NO-IMSI外，其他子状态都可以获取到有效的签约数据。

##### EMM-DEREGISTERED.NORMAL-SERVICE

当UE找到一个suitable小区并且PLMN或者TA不在禁止的列表中时，进入到此子状态。

##### EMM-DEREGISTERED.LIMITED-SERVICE

当UE知道当前选择的小区无法提供正常服务时，进入到此子状态。无法提供正常服务的原因示例如下：

1. 选择的小区在禁止的PLMN中
2. 选择的小区在禁止的TA中
3. 选择的小区是一个CSG小区，该CSG ID和关联的PLMN ID不在UE允许的CSG列表或者UE的运营商CSG列表中。

##### EMM-DEREGISTERED.ATTEMPTING-TO-ATTACH

当UE在attach或者combined attach流程中由于网络丢失响应或者环境xx导致的失败而进入到的子状态。

1. Attach not accepted by network
2. abnormal case in UE
3. combined attach not accepted by network.

eCall Only support

EF(FDN) EF(SDN) EF(FDNURI) EF(SDNURI)

### EPS update status

参考协议：《TS 24.301》5.1.3.3 EPS update status

为了描述详细的UE行为，定义了与特定签约者相关的EPS update（EU）状态。EPS update status存储在USIM中的NV（若USIM中存在相应的文件）或者在ME中得NV中。

EPS update status更新的条件为如下流程的执行：

1. attach or combined attach
2. network initiated detach
3. authentication
4. tracking area update or combined tracking area update
5. service request
6. paging for EPS service using IMSI procedure > 这会触发UE重新发起Attach
7. change in TAI which is not part of TAI list while T3346 is running. 移动出了TAI list，触发TAU

三种状态：

1. EU1：UPDATED。最近一次的attach或者TAU尝试成功。
2. EU2：NOT UPDATED。最近的一次attach，service request或者TAU尝试在程序上失败了，例如没有网络侧响应或者收到了MME的reject消息。
3. EU3：ROAMING NOT ALLOWED。最近的一次attach，service request或者TAU尝试成功执行了，但是从MME返回的响应是否定的（例如因为漫游或者签约信息限制）。

## ESM

### 概述

1. ESM子层的主要作用是支持UE与MME之间的EPS承载上下文处理。
2. ESM包括的流程有：
   1. EPS承载上下文的激活、去激活和修改
   2. UE请求到PDN的连接或者专有承载资源
   3. 通过UE与MME的控制面通道传输用户面数据（用于IoT）
3. 每个EPS承载上下文代表UE和PDN之间的一个EPS承载。即使UE和MME之间构成相应EPS承载的空口和S1承载被临时释放，EPS承载上下文仍可保持激活状态。
4. EPS 承载上下文既可以是默认承载上下文，也可以是专用承载上下文。
5. 当 UE 请求与 PDN 连接时，将激活默认 EPS 承载上下文。
6. 一般来说，只有在 UE 和 MME 之间建立了 EMM 上下文，并且 MME 通过使用第 5 条所述的 EMM 程序启动了 NAS 信息的安全交换时，才能执行 ESM 程序。然而，第一个默认 EPS 承载上下文可在 EPS 附加过程中激活。一旦 UE 成功连接，并且第一个默认 EPS 承载上下文已在连接过程中或之后激活，UE 可请求 MME 建立与其他 PDN 的连接。对于每个附加连接，MME 将激活单独的默认 EPS 承载上下文。默认 EPS 承载上下文在与 PDN 连接的整个生命周期内保持激活状态。
7. 专用 EPS 承载上下文始终与默认 EPS 承载上下文相关联，代表 UE 和 PDN 之间的额外 EPS 承载资源。只要默认 EPS 承载上下文仍处于激活状态，网络可在激活默认 EPS 承载上下文的同时或之后的任何时间启动专用 EPS 承载上下文的激活。但是，网络不得为已建立的 "非 IP "PDN 类型的 PDN 连接启动专用承载上下文激活程序。
8. 可修改默认和专用 EPS 承载上下文。专用 EPS 承载上下文可以在不影响默认 EPS 承载上下文的情况下被释放。默认 EPS 承载上下文被释放后，与之链接的所有专用 EPS 承载上下文也会被释放。
9. UE 可请求网络分配、修改或释放 EPS 承载资源。网络可通过激活新的专用 EPS 承载上下文、修改 EPS 承载上下文或停用 EPS 承载上下文来满足 UE 的此类请求。

### ESP流程的种类

ESM流程的种类可以分为两种

#### 与EPS承载上下文相关的流程

如下流程由网络发起，用于操作EPS承载上下文。

1. 默认EPS承载上下文激活
2. 专用EPS承载上下文激活
3. EPS承载上下文修改
4. EPS承载上下文去激活

如下流程由UE或网络发起用于通过控制面传输用户名数据：

1. 通过控制面传输用户数据

#### 事务相关的流程

如下流程由UE发起用于请求资源，如一个新的PDN连接/专有承载资源，或释放这些资源。

1. PDN连接流程
2. PDN断开流程
3. 承载资源分配流程
4. 承载资源修改流程

该过程由 ProSe UE 到网络中继发起，用于操作 EPS 承载上下文：

1. 远端UE报告（remote UE report）

当与Attach流程结合使用时，PDN 连接流程可触发网络执行以下与事务相关的程序：

1. ESM information request procedure

若UE和网络支持EMM-REGISERED无PDN连接并且在Attach流程中没有发起PDN连接流程，当与Attach流程关联时，UE或者网络执行如下事务相关的流程

1. ESM dummy message procedure

由 UE 成功启动的交易相关程序会触发网络执行其中一个与 EPS 承载上下文相关的程序。UE 将 EPS 承载上下文相关程序的启动视为交易相关程序的完成。

UE发起事务相关的流程会触发网络执行一个EPS承载上下文相关的一个流程。UE将EPS承载上下文相关的流程开始视为事务相关流程的结束。

Note：UE的事务相关的流程就是用于触发网络发起EPS承载上下文的流程。

在与 EPS 承载上下文相关的流程期间，MME 和 UE 不得通过控制平面程序启动用户数据的传输，直到正在进行的程序完成为止。

Note：这个要求类似于EMM流程期间不能发起ESM流程！

除了remote UE report流程和ESM information request流程，在事务相关流程期间，UE和MME不可发起通过控制面传输用户面数据直到当前流程结束

Note：也就是remote UE report流程和ESM information request流程中间是可以user data via control plane的。

如下流程可以与EPS承载上下文或者流程事务关联：

1. ESM status procedure
2. notification procedure

### ESM子层的状态

#### 承载上下文状态

BEARER CONTEXT INACTIVE：无EPS承载上下文存在

BEARER CONTEXT ACTIVE：UE中的EPS承载上下文已经激活

图 2 EPS承载上下文状态处理

总结：

1. EPS承载上下文激活状态的切换只有两条消息：ACTIVATE DEFAULT EPS BEARER CONTEXT ACCEPT和ACTIVATE DEDICATED EPS BEARER CONTEXT ACCEPT。
2. 在BEARER CONTEXT ACTIVE状态下，承载修改失败后断开失败，都将维持在此状态。
3. 注意Cause:43和49。

#### 流程事务状态

PROCEDURE TRANSACTION INACTIVE：无流程事务存在

PROCEDURE TRANSACTION PENDING：UE已经发起了流程事务



图 3 UE中的流程事务状态

### TFT

参考文档：TS 24.301

TFT：traffic flow template，业务流模板

#### Traffic flow aggregate：业务流聚合

流量集合： 包含在 UE 请求的承载资源分配流程（UE requested bearer resource allocation procedure）或 UE 请求的承载资源修改流程（UE requested bearer resource modification procedure）中的数据包过滤器的临时聚合，一旦完成 UE 请求的承载资源分配流程或 UE 请求的承载资源修改流程，网络就会将其插入 EPS 承载上下文的业务流模板 (TFT)。

#### 《6.4.1 Default EPS bearer context activation procedure》

在激活流程中，默认 EPS 承载上下文不分配任何 TFT。这相当于使用了全匹配数据包过滤器。网络可在建立该承载后随时为 EPS 默认承载分配一个 TFT，并可随后修改该默认承载的 TFT 或数据包过滤器。

#### 《6.4.2 Dedicated EPS bearer context activation procedure》

专用 EPS 承载上下文激活流程的目的是在 UE 和 EPC 之间建立具有特定 QoS 和 TFT 的 EPS 承载上下文。

#### 《6.4.2.3 Dedicated EPS bearer context activation accepted by the UE》

收到 ACTIVATE DEDICATED EPS BEARER CONTEXT REQUEST 消息后，UE 应首先检查收到的 TFT，然后再将其投入使用。

如果 TFT 包含上行链路方向的数据包过滤器，则 UE 应使用接收到的 TFT 将上行链路流量映射到无线承载。

#### 《6.4.3 EPS bearer context modification procedure》

EPS 承载上下文修改流程的目的是修改具有特定 QoS 和 TFT 的 EPS 承载上下文。

#### 《6.5.3.2 UE requested bearer resource allocation procedure initiation》

UE 应在Linked EPS bearer identity IE 中包含与请求的承载资源相关联的默认 EPS 承载的 EPS 承载标识。UE 应将Traffic flow aggregate IE 中的 TFT operation code设为 " Create new TFT "。在Required traffic flow QoS IE 中，UE 应指示 QCI，如果 UE 还包括 GBR，则指示流量聚合所需的额外 GBR。

参考文档：TS 24.008

#### 《10.5.6.12 Traffic Flow Template》关于TFT的编码

业务流模板信息元素的目的是为 PDP 上下文指定 TFT 参数和操作。此外，该信息元素还可用于向网络传输额外参数（如授权令牌；见 3GPP TS 24.229 [95]）。TFT 可包含下行链路方向、上行链路方向的数据包过滤器或同时适用于两个方向的数据包过滤器。分组过滤器决定到 PDP 上下文的流量映射。下行链路数据包过滤器应由网络使用，上行链路数据包过滤器应由 MS 使用。适用于两个方向的数据包过滤器应由网络用作下行链路数据包过滤器，由 MS 用作上行链路数据包过滤器。

流量模板是type 4 信息元素，最小长度为 3 个八位字节。IE 的最大长度为 257 个八位字节。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 8 | 7 | 6 | 5 | | 4 | | 3 | 2 | 1 | |  | |
|  | | Traffic flow template IEI | | | | | | | | | | | Octet 1 | |
|  | | Length of traffic flow template IE | | | | | | | | | | | Octet 2 | |
|  | | TFT operation code | | | | E bit | | Number of packet filters | | | | | Octet 3 | |
|  | | Packet filter list | | | | | | | | | | | Octet 4  Octet z | |
|  | | Parameters list | | | | | | | | | | | Octet z+1  Octet v | |

Figure 10.5.144/3GPP TS 24.008: *Traffic flow template* information element

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |  |
|  | 0 | 0 | 0 | 0 | Packet filter identifier 1 | | | | Octet 4 |
| Spare | | | |
|  | 0 | 0 | 0 | 0 | Packet filter identifier 2 | | | | Octet 5 |
| Spare | | | |
|  | … | | | | | | | |  |
|  | 0 | 0 | 0 | 0 | Packet filter identifier N | | | | Octet N+3 |
| Spare | | | |

Figure 10.5.144a/3GPP TS 24.008: *Packet filter list* when the TFT operation is "delete packet filters from existing TFT" (z=N+3)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |  |
|  | 0 | 0 | Packet filter direction 1 | | Packet filter identifier 1 | | | | Octet 4 |
| Spare | |
|  | Packet filter evaluation precedence 1 | | | | | | | | Octet 5 |
|  | Length of Packet filter contents 1 | | | | | | | | Octet 6 |
|  | Packet filter contents 1 | | | | | | | | Octet 7  Octet m |
|  | 0 | 0 | Packet filter direction 2 | | Packet filter identifier 2 | | | | Octet m+1 |
| Spare | |
|  | Packet filter evaluation precedence 2 | | | | | | | | Octet m+2 |
|  | Length of Packet filter contents 2 | | | | | | | | Octet m+3 |
|  | Packet filter contents 2 | | | | | | | | Octet m+4  Octet n |
|  | … | | | | | | | | Octet n+1  Octet y |
|  | 0 | 0 | Packet filter direction N | | Packet filter identifier N | | | | Octet y+1 |
| Spare | |
|  | Packet filter evaluation precedence N | | | | | | | | Octet y+2 |
|  | Length of Packet filter contents N | | | | | | | | Octet y+3 |
|  | Packet filter contents N | | | | | | | | Octet y+4  Octet z |

Figure 10.5.144b/3GPP TS 24.008: *Packet filter list* when the TFT operation is "create new TFT", or "add packet filters to existing TFT" or "replace packet filters in existing TFT"

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |  |
|  | Parameter identifier 1 | | | | | | | | Octet z+1 |
|  | Length of Parameter contents 1 | | | | | | | | Octet z+2 |
|  | Parameter contents 1 | | | | | | | | Octet z+3  Octet k |
|  | Parameter identifier 2 | | | | | | | | Octet k+1 |
|  | Length of Parameter contents 2 | | | | | | | | Octet k+2 |
|  | Parameter contents 2 | | | | | | | | Octet k+3  Octet p |
|  | … | | | | | | | | Octet p+1  Octet q |
|  | Parameter identifier N | | | | | | | | Octet q+1 |
|  | Length of Parameter contents N | | | | | | | | Octet q+2 |
|  | Parameter contents N | | | | | | | | Octet q+3  Octet v |

Figure 10.5.144c/3GPP TS 24.008: *Parameters list*

Table 10.5.162/3GPP TS 24.008: *Traffic flow template* information element

|  |
| --- |
| TFT operation code (octet 3) Bits 8 7 6  0 0 0 Ignore this IE 0 0 1 Create new TFT  0 1 0 Delete existing TFT  0 1 1 Add packet filters to existing TFT  1 0 0 Replace packet filters in existing TFT  1 0 1 Delete packet filters from existing TFT  1 1 0 No TFT operation  1 1 1 Reserved  The TFT operation code "No TFT operation" shall be used if a *parameters list* is included but no *packet filter list* is included in the *traffic flow template* information element.  The TFT operation code "Ignore this IE" shall be used by the MS if the Traffic flow aggregate information element has presence requirement "M" in a message, but the information element does not serve any useful purpose in the specific procedure for which the message is sent (see 3GPP TS 24.301 [120], subclauses 6.5.3.2 and 6.5.4.2). If the TFT operation code indicates "Ignore this IE", the MS shall also set the E bit and the number of packet filters to zero.  If the TFT operation code is set to "Ignore this IE" and the the E bit and the number of packet filters to zero, then the network shall ignore the contents of the traffic flow template information element.  E bit (bit 5 of octet 3)  The *E bit* indicates if a *parameters list* is included in the TFT IE and it is encoded as follows:  0 *parameters list* is not included  1 *parameters list* is included  Number of packet filters (octet 3)  The *number of packet filters* contains the binary coding for the number of packet filters in the *packet filter list*. The *number of packet filters* field is encoded in bits 4 through 1 of octet 3 where bit 4 is the most significant and bit 1 is the least significant bit. For the "delete existing TFT" operation and for the "no TFT operation", the *number of packet filters* shall be coded as 0. For all other operations, the number of packet filters shall be greater than 0 and less than or equal to 15.  Packet filter list (octets 4 to z)  The *packet filter list* contains a variable number of packet filters. For the "delete existing TFT" operation and the "no TFT operation", the *packet filter list* shall be empty.  For the "delete packet filters from existing TFT" operation, the *packet filter list* shall contain a variable number of packet filter identifiers. This number shall be derived from the coding of the *number of packet filters* field in octet 3.  For the "create new TFT", "add packet filters to existing TFT" and "replace packet filters in existing TFT" operations, the *packet filter list* shall contain a variable number of packet filters. This number shall be derived from the coding of the *number of packet filters* field in octet 3.  Each packet filter is of variable length and consists of  - a packet filter identifier and direction (1 octet);  - a packet filter evaluation precedence (1 octet);  - the length of the packet filter contents (1 octet); and - the packet filter contents itself (v octets).  The *packet filter identifier* field is used to identify each packet filter in a TFT. The least significant 4 bits are used.  The *packet filter direction* is used to indicate, in bits 5 and 6, for what traffic direction the filter applies:  00 - pre Rel-7 TFT filter 01 - downlink only 10 - uplink only 11 - bidirectional  Bits 8 through 7 are spare bits.  The *packet filter evaluation precedence* field is used to specify the precedence for the packet filter among all packet filters in all TFTs associated with this PDP address. Higher the value of the *packet filter evaluation precedence* field, lower the precedence of that packet filter is. The first bit in transmission order is the most significant bit.  The *length of the packet filter contents* field contains the binary coded representation of the length of the *packet filter contents* field of a packet filter. The first bit in transmission order is the most significant bit.  The *packet filter contents* field is of variable size and contains a variable number (at least one) of *packet filter components*. Each *packet filter component* shall be encoded as a sequence of a one octet *packet filter component type identifier* and a fixed length *packet filter component value* field. The *packet filter component type identifier* shall be transmitted first.  In each packet filter, there shall not be more than one occurrence of each packet filter component type. Among the "IPv4 remote address type" and "IPv6 remote address type" packet filter components, only one shall be present in one packet filter. Among the "single local port type" and "local port range type" packet filter components, only one shall be present in one packet filter. Among the "single remote port type" and "remote port range type" packet filter components, only one shall be present in one packet filter.  The term *local* refers to the MS and the term *remote* refers to an external network entity.  Packet filter component type identifier Bits 8 7 6 5 4 3 2 1  0 0 0 1 0 0 0 0 IPv4 remote address type 0 0 0 1 0 0 0 1 IPv4 local address type  0 0 1 0 0 0 0 0 IPv6 remote address type 0 0 1 0 0 0 0 1 IPv6 remote address/prefix length type 0 0 1 0 0 0 1 1 IPv6 local address/prefix length type 0 0 1 1 0 0 0 0 Protocol identifier/Next header type 0 1 0 0 0 0 0 0 Single local port type 0 1 0 0 0 0 0 1 Local port range type 0 1 0 1 0 0 0 0 Single remote port type  0 1 0 1 0 0 0 1 Remote port range type 0 1 1 0 0 0 0 0 Security parameter index type 0 1 1 1 0 0 0 0 Type of service/Traffic class type 1 0 0 0 0 0 0 0 Flow label type  All other values are reserved.  The description and valid combinations of packet filter component type identifiers in a packet filter are defined in 3GPP TS 23.060 [74] subclause 15.3.2.  For "IPv4 remote address type", the *packet filter component value* field shall be encoded as a sequence of a four octet *IPv4 address* field and a four octet *IPv4 address mask* field. The *IPv4 address* field shall be transmitted first.  For "IPv4 local address type", the *packet filter component value* field shall be encoded as defined for "IPv4 remote address type". Both the MS and network indication for support of the Local address in TFTs are required to use this packet filter component.  For "IPv6 remote address type", the *packet filter component value* field shall be encoded as a sequence of a sixteen octet *IPv6 address* field and a sixteen octet *IPv6 address mask* field. The *IPv6 address* field shall be transmitted first.  For "IPv6 remote address/prefix length type", the packet filter component value field shall be encoded as a sequence of a sixteen octet IPv6 address field and one octet prefix length field. The IPv6 address field shall be transmitted first. This parameter shall be used, instead of IPv6 remote address type, when both the MS and network indication for support of the Local address in TFT are present.  For "IPv6 local address/prefix length type", the packet filter component value field shall be encoded as defined for "IPv6 remote address /prefix length".  Both the MS and network indication for support of the Local address in TFTs are required to use this packet filter component.  NOTE: Local IP address and mask can be used when IPv6 prefix delegation is used (see 3GPP TS 23.060 [74] subclause  9.2.1.2).  For "Protocol identifier/Next header type", the *packet filter component value* field shall be encoded as one octet which specifies the IPv4 protocol identifier or IPv6 next header.  For "Single local port type" and "Single remote port type", the *packet filter component value* field shall be encoded as two octet which specifies a port number.  For "Local port range type" and "Remote port range type", the *packet filter component value* field shall be encoded as a sequence of a two octet *port range low limit* field and a two octet *port range high limit* field. The *port range low limit* field shall be transmitted first.  For "Security parameter index", the *packet filter component value* field shall be encoded as four octet which specifies the IPSec security parameter index.  For "Type of service/Traffic class type", the *packet filter component value* field shall be encoded as a sequence of a one octet *Type-of-Service/Traffic Class* field and a one octet *Type-of-Service/Traffic Class* *mask* field. The *Type-of-Service/Traffic Class* field shall be transmitted first.  For "Flow label type", the *packet filter component value* field shall be encoded as three octet which specifies the IPv6 flow label. The bits 8 through 5 of the first octet shall be spare whereas the remaining 20 bits shall contain the IPv6 flow label.  Parameters list (octets z+1 to v)  The *parameters list* contains a variable number of parameters that may be transferred. If the *parameters list* is included, the *E bit* is set to 1; otherwise, the *E bit* is set to 0.  Each parameter included in the *parameters list* is of variable length and consists of:  - a parameter identifier (1 octet);  - the length of the parameter contents (1 octet); and - the parameter contents itself (v octets).  The *parameter identifier* field is used to identify each parameter included in the *parameters list* and it contains the hexadecimal coding of the parameter identifier. Bit 8 of the *parameter identifier* field contains the most significant bit and bit 1 contains the least significant bit. In this version of the protocol, the following parameter identifiers are specified:  - 01H (Authorization Token);  - 02H (Flow Identifier); and - 03H (Packet Filter Identifier).  If the *parameters list* contains a parameter identifier that is not supported by the receiving entity the corresponding parameter shall be discarded.  The *length of parameter contents* field contains the binary coded representation of the length of the *parameter contents* field. The first bit in transmission order is the most significant bit.  When the *parameter identifier* indicates Authorization Token, the *parameter contents* field contains an authorization token, as specified in 3GPP TS 29.207 [100]. The first octet is the most significant octet of the authorization token and the last octet is the least significant octet of the authorization token.  The *parameters list* shall be coded in a way that an Authorization Token (i.e. a parameter with identifier 01H) is always followed by one or more Flow Identifiers (i.e. one or more parameters with identifier 02H).  If the *parameters list* contains two or more consecutive Authorization Tokens without any Flow Identifiers in between, the receiver shall treat this as a semantical TFT error.  When the *parameter identifier* indicates Flow Identifier, the *parameter contents* field contains the binary representation of a flow identifier. The Flow Identifier consists of four octets. Octets 1 and 2 contains the Media Component number as specified in 3GPP TS 29.207 [100]. Bit 1 of octet 2 is the least significant bit, and bit 8 of octet 1 is the most significant bit. Octets 3 and 4 contains the IP flow number as specified in 3GPP TS 29.207 [100]. Bit 1 of octet 4 is the least significant bit, and bit 8 of octet 3 is the most significant bit.  When the *parameter identifier* indicates Packet Filter Identifier, the parameter contents field contains the binary representation of one or more packet filter identifiers. Each packet filter identifier is encoded in one octet, in the 4 least significant bits. This parameter is used by the MS and the network to identify one or more packet filters in a TFT when modifying the QoS of a PDP context without modifying the packet filter itself. |

## IP地址分配

**问题：IP地址的分配过程**

UE 可在建立 EPS 默认承载上下文时配置 IPv4 地址。默认承载建立后，UE 可通过基于 IETF 的 IP 地址分配机制获取 IPv4 地址或 IPv6 前缀，或两者兼得。

IPCP/CHAP/PAP/LCP 参考TS 29.061

如下基于IETF的IP地址/前缀分配方法用于EPS系统。

1. 通过IPv6无状态地址自动分配/64 IPv6默认前缀。可选的方法，通过有状态的DHCPv6前缀授权分配额外的IPv6前缀，长度为64或更短。
2. 通过DHCPv6分配IPv4地址和IPv4参数。
3. 通过无状态的DHCPv6分配IPv6参数配置

一旦PDN连接的默认承载去激活，UE应本地释放与此PDN连接相关的任何IPv4地址和IPv6前缀。

### 通过NAS信令分配IP地址

**问题：UE在激活默认承载时请求IP类型的策略**

基于IP协议栈的配置，UE应当设置PDN类型IE在PDN CONNECTIVITY REQUEST消息中，设置如下。

1. UE同时具备IPV4和IPV6的能力时的处理方法：
   1. 针对当前APN未分配IP地址，则应当设置PDN类型为IPv4v6。
   2. 针对当前APN分配了IPv4地址并收到了原因值#52 "single address bearers only allowed"，当前请求IPv6地址时，应设置PDN类型为IPv6
   3. 针对当前APN分配了IPv6地址并收到了原因值#52 "single address bearers only allowed"，当前请求IPv4地址时，应设置PDN类型为IPv4

Note：若没分配过IP地址，则PDN类型默认为IPv4v6，若分配过且网络指示只能单地址，则将PD类型设置为需要请求的IP类型。

1. 仅支持IPv4的UE，设置PDN类型为IPv4
2. 仅支持IPv6的UE，设置PDN类型为IPv6
3. 当IP版本号支持能力未知，则UE设置PDN类型为IPv4v6

若UE期望使用DHCPv4请求IPv4地址分配，则需要通过在PDN CONNECTIVITY REQUEST消息中携带PCO指示。

若UE期望请求一个无IP的PDN连接，则需要通过在PDN CONNECTIVITY REQUEST消息中设置PDN类型为“Non IP”

### 网络分配IP的处理过程

TS 24.301 6.2.2 带梳理！

网络分配IP地址需要考虑的因素：PDN类型，归属和拜访运营商的策略、用户的签约数据。

1. 若UE请求PDN类型为IPv4v6，但是签约数据限制为请求ANP的类型为仅IPv4或仅IPv6，那么网络将覆盖UE请求的PDN类型，使其仅限于单地址的PDN类型（IPv4或IPv6）。在发送给 UE 的 ACTIVATE DEFAULT EPS BEARER CONTEXT REQUEST 消息中，网络应将 PDN 类型值设置为 "IPv4 "或 "IPv6"，并将 ESM 原因值分别设置为 **#50 "PDN 类型仅允许 IPv4 "或 #51 "PDN 类型仅允许 IPv6"**。UE 随后不得启动与同一 APN 的另一个 UE 请求 PDN 连接程序，以获得与网络允许的 PDN 类型不同的 PDN 类型，直至：
   1. 由于以下原因，给定 APN 的所有 EPS 承载上下文在 UE 被去激活：
      1. 在TAU或Service Request过程中 EPS 承载上下文同步；
      2. 网络发起的EPS承载上下文去激活流程；
      3. 如 6.4.4.6 子条款规定的无 NAS 信令的本地 EPS 承载上下文去激活；
      4. Detach流程
      5. TAU或Service Request流程被拒绝，携带的原因值导致UE进入到EMM-DEREGISTERED状态。
   2. 用于访问 APN 的 PDN 类型发生变化；
2. 如果 UE 请求 PDN 类型 IPv4v6，但 PDN GW 配置规定此 APN 仅使用 IPv4 寻址或仅使用 IPv6 寻址，则网络应覆盖 UE 请求的 PDN 类型，将其限制为单一地址 PDN 类型（IPv4 或 IPv6）。在发送给 UE 的 ACTIVATE DEFAULT EPS BEARER CONTEXT REQUEST 消息中，网络应将 PDN 类型值设置为 "IPv4 "或 "IPv6"，并将 ESM 原因值分别设置为 #50 "PDN 类型仅允许 IPv4 "或 #51 "PDN 类型仅允许 IPv6"。UE 随后不得启动与同一 APN 的另一个 UE 请求 PDN 连接程序，以获得与网络允许的 PDN 类型不同的 PDN 类型，直至：Step1中的a)部分流程发生。

#50 "PDN type IPv4 only allowed"

#51 "PDN type IPv6 only allowed"

1. 如果 UE 请求 PDN 类型 IPv4v6，但运营商对每个承载使用单一寻址（例如，由于与早期版本的节点互通），则网络应将 UE 请求的 PDN 类型覆盖为单一 IP 版本。在发送给 UE 的 ACTIVATE DEFAULT EPS BEARER CONTEXT REQUEST 消息中，网络应将 PDN 类型值设置为 "IPv4 "或 "IPv6"，并将 ESM 原因值设置为 #52 "仅允许单地址承载"。UE 随后应使用 UE 请求的 PDN 连接程序，使用单地址 PDN 类型（IPv4 或 IPv6）向同一 APN 请求另一个 IP 版本的 PDN 连接，而不是已激活的 PDN 连接；

#52 "single address bearers only allowed"

1. 如果网络将 PDN 类型设置为 IPv4 或 IPv4v6，则网络应在 PDN 地址信息中包含 IPv4 地址。在这种情况下，如果要使用 DHCPv4 配置 IPv4 地址，网络应将 IPv4 地址设置为 0.0.0.0；以及
2. 如果网络将 PDN 类型设置为 IPv6 或 IPv4v6，则网络应在 PDN 地址信息中包含 UE 用于链路本地地址的**接口标识符**。

网络应在发送给 UE 的 ACTIVATE DEFAULT EPS BEARER CONTEXT REQUEST 消息中的 PDN 地址 IE 中包含 PDN 类型和 PDN 地址信息。

### IP头压缩

目前仅用于控制面CIoT EPS优化场景！

如果 IP PDN 类型的 PDN 连接支持控制平面 CIoT EPS 优化，则 UE 和 MME 可支持用于 IP 报头压缩的RObust Header Compression(ROHC) 框架（见 IETF RFC 5795 [37]）。

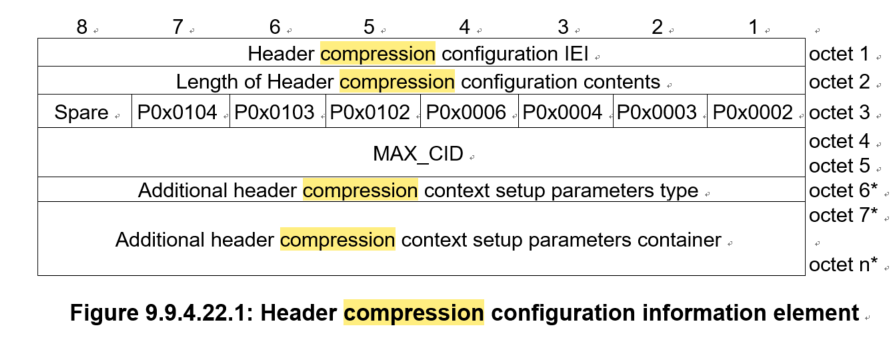
如果支持用于控制平面 CIoT EPS 优化的 IP 报头压缩，则可支持 3GPP TS 36.323 [38] 中定义的 ROHC 配置文件。

ROHC 配置是在 **UE 请求 PDN 连接过程中协商和建立的**，如 6.5.1 子条款所规定。

在Attach和TAU过程中，UE 和 MME 都要指明是否支持用于控制平面 CIoT EPS 优化的 IP 标头压缩（见 5.5.1 和 5.5.3 子条款）。ROHC 配置可通过使用子条款 6.4.3 和 6.5.4 中规定的 UE 请求的承载资源修改程序或 EPS 承载上下文修改程序重新协商。

#### 头压缩的协商配置

Header compression configuration



承载的消息列表如下：

1. PDN CONNECTIVITY REQUEST(UE2NW)
2. BEARER RESOURCE MODIFICATION REQUEST(UE2NW)
3. MODIFY EPS BEARER CONTEXT REQUEST(NW2UE)
4. ACTIVATE DEFAULT EPS BEARER CONTEXT REQUEST(NW2UE)

### ESM流程的地址处理原则

#### 事务相关流程地址参数PTI

PTI：Procedure Transaction Identity，流程事务标识

EBI：EPS Bearer Identity，EPS承载标识

1. 事务相关流程使用PTI作为 ESM 消息头中的地址参数。
2. 当 UE 或网络启动事务相关流程时，应在消息头部中包含有效的PTI值，并将 EBI设置为 " no EPS bearer identity assigned "。
3. 事务相关的请求对应的响应如果还是事务相关的，则响应消息中设置与请求中相同的PTI，并将EBI设为 " no EPS bearer identity assigned "。
4. 事务相关的响应消息有：
   1. PDN CONNECTIVITY REJECT
   2. PDN DISCONNECT REJECT
   3. BEARER RESOURCE ALLOCATION REJECT
   4. BEARER RESOURCE MODIFICATION REJECT
   5. ESM INFORMATION REQUEST
   6. ESM DUMMY MESSAGE（网络到UE）
   7. ESM INFORMATION RESPONSE（UE到网络）
   8. ESM DUMMY MESSAGE（UE到网络）
5. 如果发送的 ESM DUMMY MESSAGE 是对收到的 ESM DUMMY MESSAGE 的响应，则发送实体应将收到的PTI值包含在消息头中，并将 EBI设置为 " no EPS bearer identity assigned "。
6. 几种事务发起的场景以及相应的PTI和EBI设置。



图 4 UE发起事务相关流程并被网络拒绝



图 5 UE发起事务相关流程并被网络发起的事务请求响应



图 6 网络发起的事务相关流程



图 7 UE发起的事务相关流程

#### EPS承载上下文相关流程地址参数EBI

EPS 承载上下文相关流程使用EBI作为 ESM 报文头的地址参数。当网络启动 EPS 承载上下文相关流程时，应在报文头中包含有效的EBI。程序事务标识值应设置如下：

1. 如果 EPS 承载上下文相关流程是通过收到 UE 的事务相关请求消息触发的，网络应在 EPS 承载上下文相关请求消息的消息头中包含与事务相关请求消息一起收到的PTI（见图 6.3.2.3 示例）。



图 8 EPS承载上下文相关流程由事务相关请求触发

1. 如果程序是由网络内部触发的，网络应将 EPS 承载上下文相关请求报文报文头中的PTI设置为 " no procedure transaction identity assigned "（见图 6.3.2.4 示例）。



图 9 由网络内部触发的EPS承载上下文相关流程

1. 如果流程是通过控制平面传输用户数据触发的，网络应将 EPS 承载上下文相关请求报文头中的PTI设置为 " no procedure transaction identity assigned "（见图 6.3.2.5 示例）。



图 10 控制面传输用户数据触发的EPS承载上下文相关流程

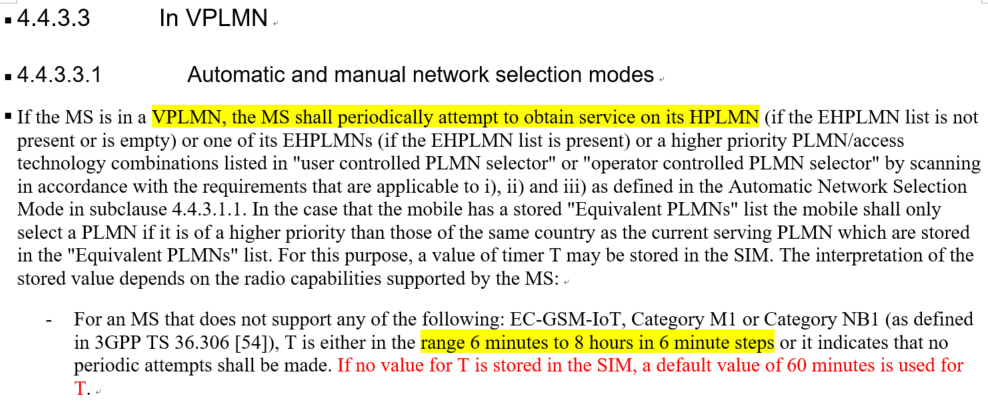
总结：

1. 事务相关的

## PLMN Related

### 高优先级PLMN搜索

参考文档：TS 23.122 4.4.3.3.1 Automatic and manual network selection modes



1. 若MS驻留在**VPLMN**且搜网模式为**自动网络选择模式**，则MS应当周期性的尝试获取在HPLMN/EHPLMN，或者在UPLMN/OPLMN中的一个较高优先级的PLMN的服务
2. 若MS中存储了与当前VPLMN等效的EPLMN，则MS只能选择优先级高于EPLMN的，且与当前服务的PLMN相同国家码的PLMN。
3. 重复高优先级的时间间隔：Timer T，存储在SIM卡中。
4. 对于不支持EC-GSM-IoT, Category M1 or Category NB1其中任何一种的MS，Timer T要么在 6 分钟至 8 小时的范围内，以 6 分钟为单位，要么表示不进行定期尝试。如果 SIM 中未存储 T 值，则 T 默认为 60 分钟。
5. 如果MS根据TS 31.102或TS 24.368配置了MinimumPeriodicSearchTimer（最新周期搜网定时器），则MS不应当使用比MinimumPeriodicSearchTimer还小的Timer T，当Timer T小于MinimumPeriodicSearchTimer，则使用MinimumPeriodicSearchTimer。
6. 激活PSM期间，Timer T不应当停止。参考TS 24.008和TS 24.301。
7. MS可以设置Fast Higher Priority PLMN search（快速高优先级PLMN搜索）。此设置需要根据TS 31.102或TS 24.368进行参数设置。

HPPLMN的处理流程：

1. 周期的尝试发生在自动搜网模式和处于漫游中的MS。另，MS附着到紧急承载、注册到紧急服务、有一个紧急服务的PDU会话或有一个紧急承载服务的PDN连接时不可执行HPPLMN搜索。
2. MS首次执行HPPLMN搜索的周期为2分钟~T分钟。场景如下：
   1. Fast First Higher Priority PLMN search未使能且处于首次开机
   2. Fast First Higher Priority PLMN search使能，开机或选择到了VPLMN。

#### 代码处理

定时器：EMM\_TeFHPLMN

时长：TeFHPLMN\_duration

如果是第一次尝试回到HPLMN，则将Timer T的定时器设置为120s，用于快速回到HPLMN，后续按照协议规定的时间处理。

定时器超时处理函数：void EMmSession::EmmTryPeriodicBackHome()

处理检查：

1. EMM处于EMM\_IDLE且子状态为EMM\_REGISTERED\_NORMAL\_SERVICE
2. 当前驻留的PLMN为VPLMN
3. 网络选择模式为自动选网

几个标识符：

1. this->periodicGoingHPendingForStateMismatch需要执行周期性回HPPLMN但是EMM子状态不匹配
2. this->periodicGoingHPendingForRrcConn需要执行周期性回HPPLMN但是RRC处于连接态
3. this->periodicGoingHomeInProgress回HPPLMN过程中

### FPLMN处理

消息：EMM\_ERRC\_FORBIDDEN\_PLMN\_CONFIG

发送场景：

1. EMM读取USIM卡时将FPLMN列表发送个ERRC
2. Atttach Reject和TAU Reject携带Cause#14 EPSServicesNotAllowedInThisPLMN
3. Atttach Reject和TAU Reject携带Cause#15 NoSuitableCellsInTrackingArea且携带了EMM Cause为EUTRAN Not Allowed
4. 手动搜网模式下，Attach Accept和TAU Accept消息处理时将当前选择的PLMN从FPLMN列表删除时
5. 手动搜网模式下收到Attach Accept消息时
6. T3245超时，清空FPLMN列表时通知ERRC。

EMM\_ERRC\_MOBILITY\_CONFIG

## RRC

### RRC Connection Request

TS 36.331, clause 5.3.3.3 “Actions related to transmission of RRCConnectionRequest message”

Upper layers provide the S-TMSI if the UE is registered in the TA of the current cell

RRC发起RRCConnectionRequest时，若UE当前小区所属的TA是注册的TA，则上层需要提供S-TMSI。

处理：RRC发起前需要检查当前的TAI是否位于Registered TAI List中，位于则触发RRCConnectionRequest时带上S-TMSI，否则不带。

# 编解码

# 科普类

### 什么是MS、UE、MT、TE、TA

首先，参考TR 21.905

参考链接：[移动台 - 维基百科，自由的百科全书 (wikipedia.org)](https://zh.wikipedia.org/wiki/%E7%A7%BB%E5%8A%A8%E5%8F%B0)

1. Terminal Equipment (TE): Equipment that provides the functions necessary for the operation of the access protocols by the user. A functional group on the user side of a user-network interface (source: ITU-T I.112).

Note：提供了用户接入到网络的功能，也就是应用层设备。

1. Mobile Equipment (ME): The Mobile Equipment is functionally divided into several entities, i.e.one or more Mobile Terminations (MT) and one or more Terminal Equipments (TE).

参考文档：GSM 11.11: "Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module ‑ Mobile Equipment (SIM ‑ ME) interface".

Note：ME的介绍没有详细说明具体的组成，但是从上面参考文档的写法推测出，ME包括了MT和TE，但是不包括SIM。可以理解为不插卡的“手机”形态。

1. Mobile Termination (MT): The Mobile Termination is the component of the Mobile Equipment (ME) which supports functions specific to management of the PLMN access interface (3GPP or non-3GPP). The MT is realized as a single functional entity.

Note：MT提供了接入到PLMN的功能，作为一个单功能实体实现，也就是常说的Modem、CP模块。

1. Mobile Station (MS): A Mobile Station (MS) corresponds to a User Equipment (UE). See 3GPP TS 24.002.

Note：MS和UE(User Equipment)一样，就是移动电话，包含了TE、MT。

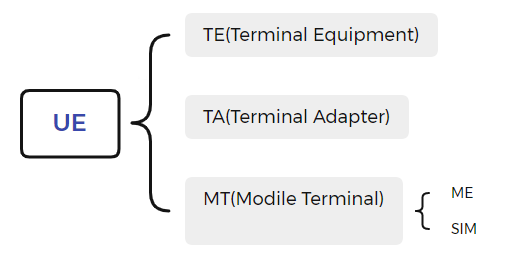


图 11 UE的构成图

TE、TA、MT封装在一个模块中，构成一部移动电话，但是TE和MT的功能由截然不同的处理器承担，TE运行在应用处理器上，也就是常说的AP；MT运行于基带处理器(BaseBand Processor)，也就是常说的BP，或者协处理器(Co-Processor，基带芯片+协处理器)。

MT中若未插入SIM卡，则称为ME。

### Feature Phone和Smart Phone

参考链接：[AP与CP介绍【转】 - 请给我倒杯茶 - 博客园 (cnblogs.com)](https://www.cnblogs.com/zzb-Dream-90Time/p/9992271.html)

Feature Phone，更通俗的说法为功能机，提供基本的通讯服务，行业属于称为移动台（Mobile Station）。

Smart Phone，智能手机，CP+AP，是增加了无线通信功能的手持式电脑。

在智能手机中，手机功能的实现以应用处理器（AP）为主，基带芯片提供通信功能。可以把AP看作计算机，把基带芯片看作AP的无线modem。这个无线modem通过AT接口（相当于计算机和调制解调器之间的接口，但各厂家都有扩展命令）提供通话、短消息、上网、UIM卡等功能。

二者区别

智能手机和FeaturePhone究竟有什么区别？其实，FeaturePhone可以实现智能手机的大多数功能。两者最本质的区别就是不同的出发点。FeaturePhone是在不断扩充应用功能的无线通信终端（行业术语叫移动台），而智能手机是增加了无线通信功能的手持式电脑。

FeaturePhone的软件基本上都运行在CPU的特权模式，在PC程序员看来，FeaturePhone的软件就是一个做了UI的宏内核。而智能手机的软件体系基本上照搬了PC的软件体系，将内核、驱动（可以编入内核，也可以独立）和应用分开。

### 什么是Non-IP

ISG：Industry Specification Group，行业规范小组

NIN：Non-IP Networking

我们的愿景是建立一个更高效、更能满足用户需求的系统。

虚拟化时代的网络

TCP/IP 网络协议套件已有 40 多年的历史，其设计初衷与 2020 年代的网络需求不同。这就产生了寻址、移动性、性能和安全问题；而要解决这些问题，需要付出巨大的努力、精力和成本，并增加了极大的复杂性。

随着现代网络在支持新用例（其中一些要求超低延迟）和更大连接性方面面临的挑战越来越多，服务提供商正在寻找比当前系统中使用的基于 TCP/IP 的网络更能满足其需求的候选技术。

ISG NIN 正在对网络技术进行标准化，以满足移动性、安全性、隐私性、有效利用频谱和超低延迟等 21 世纪要求。非 IP 网络这一名称强调了该技术并不依赖于 IP 数据包格式或协议；但它支持 TCP/IP 套件以及其他系统，如信息中心网络（Information Centric Networking）。希望访问由 URL 或虚拟化网络服务标识的服务（或某些内容）的应用程序可以直接提出请求，而不必首先查找标识提供服务位置的 IP 地址。

软件定义网络（SDN）可为 IP 网络提供流量控制覆盖。与 SDN 类似，ISG NIN 的 Flexilink 技术也认识到，大多数数据包都是 "流 "的一部分，系统必须做出的大多数决定都是按 "流 "而不是按数据包做出的。例如，流量中的每个数据包从交换机到目的地的 "下一跳 "都是相同的；携带实时音频或视频流的数据包会定期到达，需要及时转发，而对延迟要求不那么严格的其他应用的数据包则可以排队。当出现故障或移动设备移动到不同的小区时，可以很容易地对流量进行重定向。

通过www.DeepL.com/Translator（免费版）翻译

参考链接：

[ETSI - Next Generation Mobile Networks | Next Generation Protocols](https://www.etsi.org/technologies/non-ip-networking)

### NIDD的实现

参考链接：[NON IP Data delivery (NIDD) (iotcreators.com)](https://docs.iotcreators.com/docs/non-ip-data-delivery-nidd)

NIDD：Non-IP Data Delivery，非IP数据传输

用途：物联网设备

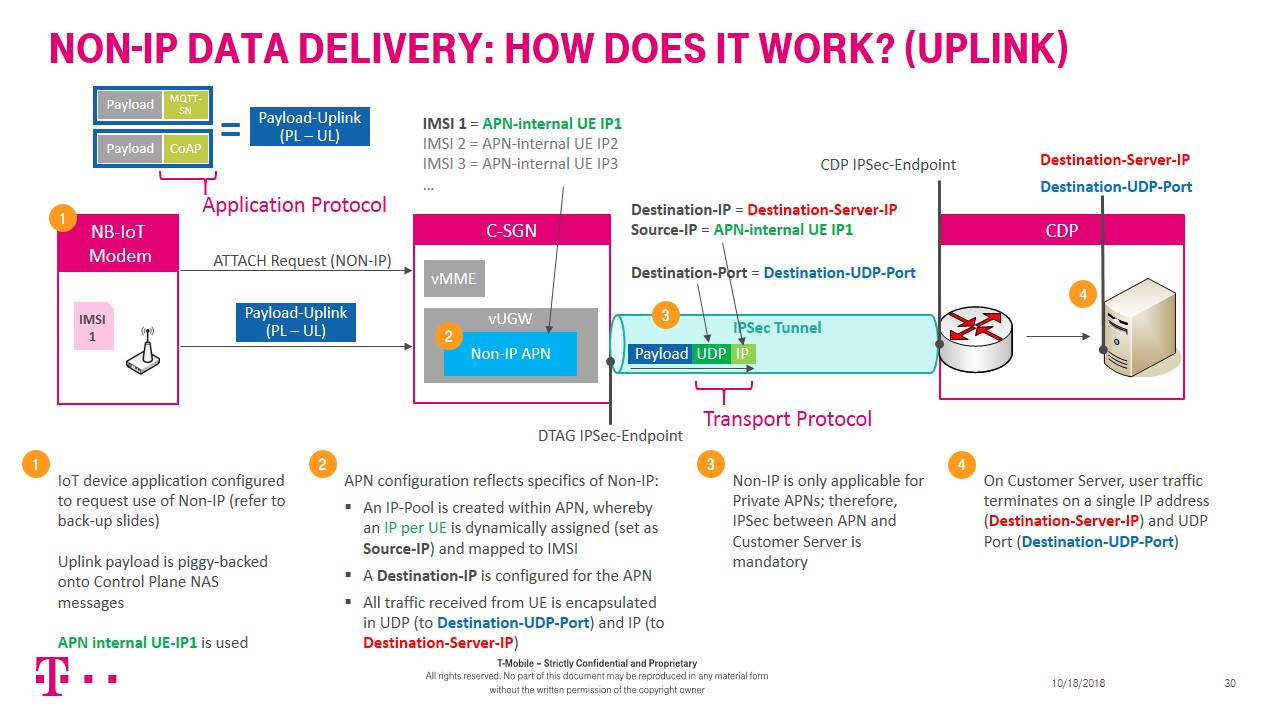
#### 什么是NON-IP

蜂窝连接有两层连接，即管理层（控制平面）和数据层（用户平面）。对于偶尔传输少量数据的服务，由于所需的信令量和 "通话时间 "减少，控制平面的使用将优化功耗。使用非 IP UDP 可以优化功耗。非 IP 允许使用针对特定用途进行了优化的协议。UDP 是异步协议，可缩短连接时间，而 TCP 协议则会保持连接开放，直到收到确认为止。

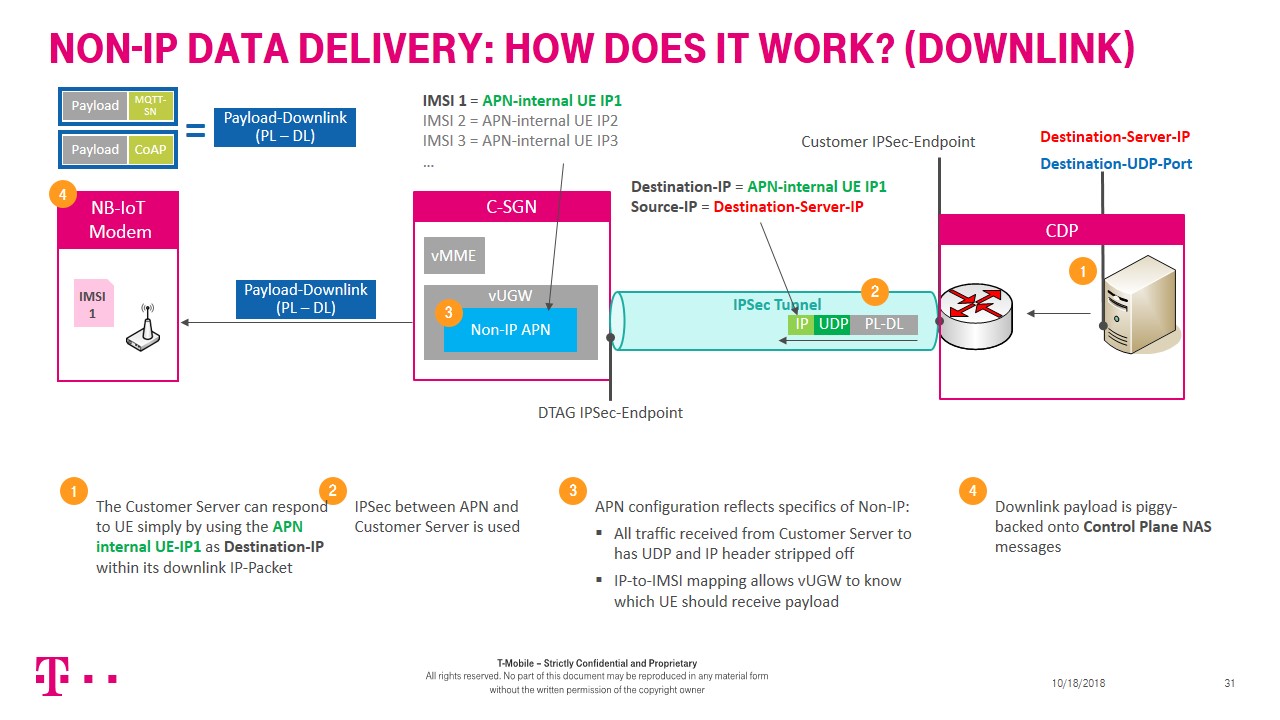
总结：偶尔传输的少量数据服务使用控制面信令携带，优点：信令量少，功耗小，连接时间短。

#### 运行机制

1. 上行



1. 下行



### CIoT EPS optimizations

参考文档：TS 24.301，此功能由R13引入。

CIoT的EPS优化，关于CIoT EPS Optimizatons要点如下:

1. 主要作用：改进了小数据和短信的传输
2. 能力指示：Attach流程或TAU流程中指示UE支持的CIoT网络行为和偏向
3. UE可以指示对控制面CIoT EPS Optimization、用户面CIoT EPS Optimization、EMM-REGISTERED without PDN connection、S1-U data transfer和header compression的支持。
4. UE可以在Attach流程中使用无Combined Attach流程的SMS传输。
5. UE 可以在表示支持的同时，单独表示对控制平面 CIoT EPS 优化或用户平面 CIoT EPS 优化的preference，Preference指示也被视为**使用请求。**

### 传输模式

### RNTI

参考文档：TS 36.321

RNTI：Radio Network Temporary Identifier，无线网络临时标识符。

RA-RNTI：Random Access RNTI，RA-RNTI 用于传输随机接入响应消息时的 PDCCH。它能明确识别 UE 利用哪个时频资源传输随机接入前导码。

C-RNTI Cell RNTI

SI-RNTI System Information RNTI

P-RNTI Paging RNTI

M-RNTI MBMS RNTI

表格 1 RNTI的用途

|  |  |  |  |
| --- | --- | --- | --- |
| RNTI | Usage | Transport Channel | Logical Channel |
| P-RNTI | Paging and System Information change notification | PCH | PCCH |
| SI-RNTI | Broadcast of System Information | DL-SCH | BCCH |
| M-RNTI | MCCH Information change notification | N/A | N/A |
| RA-RNTI | Random Access Response | DL-SCH | N/A |
| Temporary C-RNTI | Contention Resolution (when no valid C-RNTI is available) | DL-SCH | CCCH |
| Temporary C-RNTI | Msg3 transmission | UL-SCH | CCCH, DCCH, DTCH |
| C-RNTI | Dynamically scheduled unicast transmission | UL-SCH | DCCH, DTCH |
| C-RNTI | Dynamically scheduled unicast transmission | DL-SCH | CCCH, DCCH, DTCH |
| C-RNTI | Triggering of PDCCH ordered random access | N/A | N/A |
| Semi-Persistent Scheduling C-RNTI | Semi-Persistently scheduled unicast transmission  (activation, reactivation and retransmission) | DL-SCH, UL-SCH | DCCH, DTCH |
| Semi-Persistent Scheduling C-RNTI | Semi-Persistently scheduled unicast transmission  (deactivation) | N/A | N/A |
| TPC-PUCCH-RNTI | Physical layer Uplink power control | N/A | N/A |
| TPC-PUSCH-RNTI | Physical layer Uplink power control | N/A | N/A |

表格 2 RNTI的取值

|  |  |
| --- | --- |
| Value (hexa-decimal) | RNTI |
| 0000 | N/A |
| 0001-003C | RA-RNTI, C-RNTI, Semi-Persistent Scheduling C-RNTI, Temporary C-RNTI, TPC-PUCCH-RNTI and TPC-PUSCH-RNTI (see note) |
| 003D-FFF3 | C-RNTI, Semi-Persistent Scheduling C-RNTI, Temporary C-RNTI, TPC-PUCCH-RNTI and TPC-PUSCH-RNTI |
| FFF4-FFFC | Reserved for future use |
| FFFD | M-RNTI |
| FFFE | P-RNTI |
| FFFF | SI-RNTI |

### 什么是DCI

DCI：Downlink Control Indicator，下行控制指示

参考链接：<https://www.sharetechnote.com/html/DCI.html>

在学习物理层帧结构时，你一定会对资源分配的所有方式的灵活性印象深刻，它是时域、频域和调制方式的组合。

特别是在频域，你可以使用如此多的资源块（在 20 Mhz 带宽的情况下有 100 个资源块），如果考虑到这些变量的所有可能排列方式，可能的资源分配方式将非常多。那么你就会有这样一个问题（至少我有这个问题）：另一方（接收方）如何才能准确地知道发送方（发射方）是在哪个时隙以哪种调制方案发送数据（子帧）的？我只是捕捉到了物理信号，但我（接收方）如何解码这个信号呢？这就是所谓 "DCI（下行链路控制指标）"的由来。

DCI指示的内容：

1. 必须在哪个资源块中携带数据
2. 必须使用哪种调制方案解码数据
3. 功率控制命令（Power Control Command）
4. CSI报告请求
5. CQI报告请求

如果是下行数据，则接收方必须先解码DCI，从DCI中获得信息解码真正的数据，没有DCI就无法解码下行的实际数据内容。

如果是上行数据，则发生方需要收到基站的DCI0，才能发送数据。

#### DCI的种类

|  |  |  |
| --- | --- | --- |
| DCI Format | Usage | Major Contents |
| [Format 0](https://www.sharetechnote.com/html/DCI.html#DCI_Format_0) | UL Grant. Resource Allocation for UL Data | RB Assignment,TPC,PUSCH Hopping Flag |
| [Format 1](https://www.sharetechnote.com/html/DCI.html#DCI_Format_1) | DL Assignment for SISO | RB Assignment,TPC, HARQ |
| [Format 1A](https://www.sharetechnote.com/html/DCI.html#DCI_Format_1A) | DL Assignment for SISO (compact) | RB Assignment,TPC, HARQ |
| [Format 1A](https://www.sharetechnote.com/html/DCI.html#Format_1A_C_RNTI_PDCCH_Order) | Triggering RACH in Connected States | PRACH Sequence Index |
| [Format 1B](https://www.sharetechnote.com/html/DCI.html#DCI_Format_1B) | DL Assignment for MIMO with Rank 1 | RB Assignment,TPC, HARQ,TPMI, PMI |
| [Format 1C](https://www.sharetechnote.com/html/DCI.html#DCI_Format_1C) | DL Assignment for SISO (minimum size) | RB Assignment |
| [Format 1D](https://www.sharetechnote.com/html/DCI.html#DCI_Format_1D) | DL Assignment for Multi User MIMO | RB Assignment,TPC, HARQ,TPMI,DL Power Offset |
| [Format 2](https://www.sharetechnote.com/html/DCI.html#DCI_Format_2) | DL Assignment for Closed Loop MIMO | RB Assignment,TPC, HARQ, Precoding Information |
| [Format 2A](https://www.sharetechnote.com/html/DCI.html#DCI_Format_2A) | DL Assignment for Open Loop MIMO | RB Assignment,TPC, HARQ, Precoding Information |
| [Format 2B](https://www.sharetechnote.com/html/LTE_Advanced_#DCI_Format_2B) | DL Assignment for TM8 (Dual Layer Beamforming) | RB Assignment,TPC, HARQ, Precoding Information |
| [Format 2C](https://www.sharetechnote.com/html/LTE_Advanced_#DCI_Format_2C) | DL Assignment for TM9 | RB Assignment,TPC, HARQ, Precoding Information |
| [Format 3](https://www.sharetechnote.com/html/DCI.html#DCI_Format_3) | TPC Commands for PUCCH and PUSCH with 2 bit power adjustment | Power Control Only |
| [Format 3A](https://www.sharetechnote.com/html/DCI.html#DCI_Format_3A) | TPC Commands for PUCCH and PUSCH with 1 bit power adjustment | Power Control Only |
| Format 4 | UL Assignment for UL MIMO (up to 4 layers) | RB Assignment,TPC, HARQ, Precoding Information |

#### 特定情况下决定DCI格式的因素

1. RNTI类型
2. 传输模式

## 借助tshark解码任何码流

两个步骤

1. 使用text2pcap工具将二进制码流转换为

tshark.exe -V -o "uat:user\_dlts:\"User 0 (DLT=147)\",\"nas-eps\",\"0\",\"\",\"0\",\"\"" -r decode\_temp.pcap

# 语句积累

Unless explicitly stated otherwise, the procedures described in the following subclauses can only be executed whilst a NAS signalling exists between the UE and the MME.

翻译：

除非另有明确说明，否则只能在 UE 和 MME 之间存在 NAS 信令的情况下执行以下子条款中描述的程序。

be supposed to do,应该做什么事

You are never supposed to wake up a sleeping baby.

# PDCP代码学习

## 协议内容学习

### UPDCP处理流程

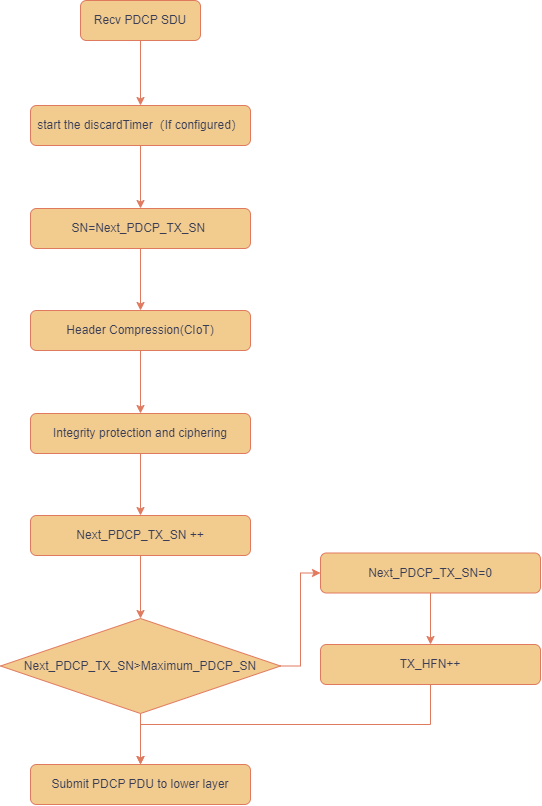


图 12 UPDCP业务处理过程

几个重要的变量：

1. SN：Sequence Number，序列号

2. RX\_HFN：Hyper Frame Number，超帧号，SN递增到最大值后RX\_HFN增加1

3. COUNT：RX\_HFN<<(SN占用的bit数)+SN

4. Next\_PDCP\_TX\_SN：下一个SDU对于的SN号

UPDCP处理过程：

1. 收到PDCP SDU后，启动改SDU对应的Discard timer（若配置了）
2. 分配SN号，Next\_PDCP\_TX\_SN为当前SDU对应的SN号。
3. 头压缩
4. 先完保后加密
5. Next\_PDCP\_TX\_SN增1，若超过最大值则置为0，TX\_HFZ增加1
6. 将编号的PDCP PDU发送个低层处理。

### DPDCP处理流程

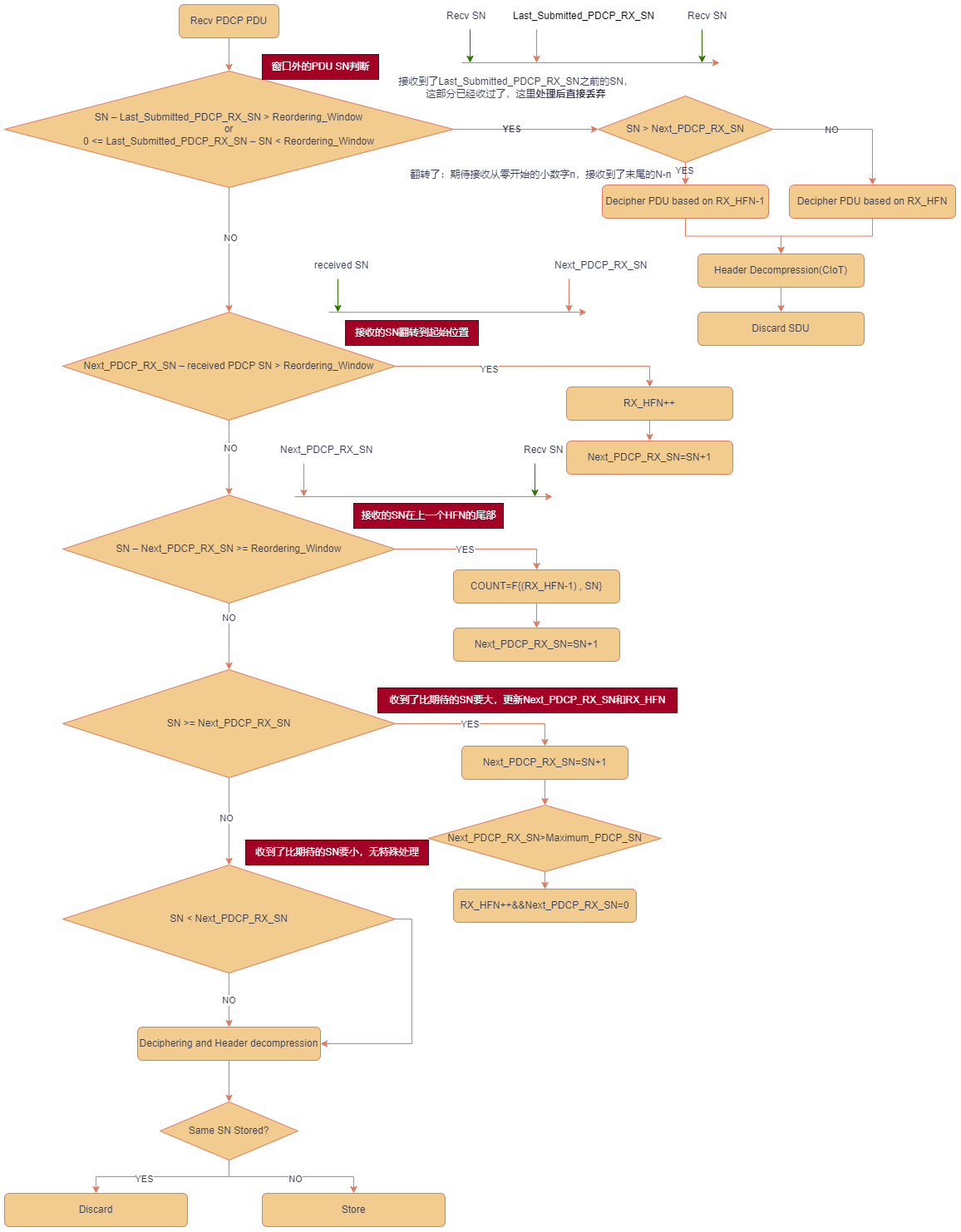


图 13 DPDCP处理流程图

DPDCP处理接收时的几个重要状态变量

1. Recv PDCP SN

当前接收到的PDCP PDU的SN号

1. Last\_Submitted\_PDCP\_RX\_SN

上次提交给上层的最后一个SDU的SN号。

1. Reordering\_Window

重排序窗口，该窗口的下边界为Last\_Submitted\_PDCP\_RX\_SN，下边界为Last\_Submitted\_PDCP\_RX\_SN+ Reordering\_Window

1. Next\_PDCP\_RX\_SN

下一次期望接收到PDU的SN，为上一次接收到PDU SN+1

1. RX\_HFN

当前PDU所在的接收超帧号，该值不在PDU中发送，由发送端和接收端分别维护，用于计算COUNT。

1. COUNT

加解密和完保过程中需要用到，值为32bit的整型变量，可能会发生翻折。取值如下，当SN去到最大值再增加时，HFN增加1，SN归0零。



## 代码学习

### 判断IP包为TCP的ACK包

检测方法：

#define TCP\_RAW\_ACK(x) (((x)[0]==0x45) && ((x)[9]==0x06) && (((x)[33]&0x10)!=0))

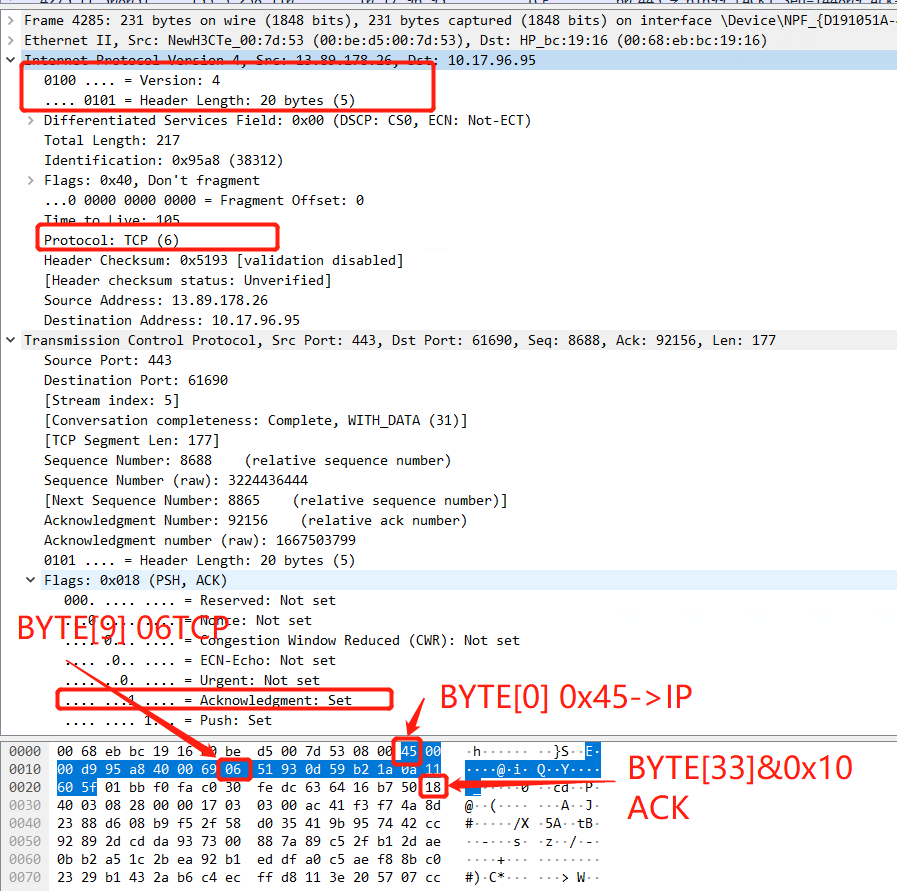


图 14 IP包中TCP.ACK

#### 相关处理

//at+ecfg=maxpritxdatalen,a,b,c,d

//a:UPDCP 囤包门限值 for mr4207 4208

//b:TCP ACK精确识别的标志，取值0,1

//c:UPDCP 从囤包BUFFER中一次递交给URLC的数据包的最大值

//d:UPDCP 一次递交延时数据包的最大个数

### 检测Ping包

检测方法：

#define TCP\_PING(x) (((x)[0]==0x45) && ((x)[9]==0x01) && (((x)[20]==0) || ((x)[20]==0x08)))

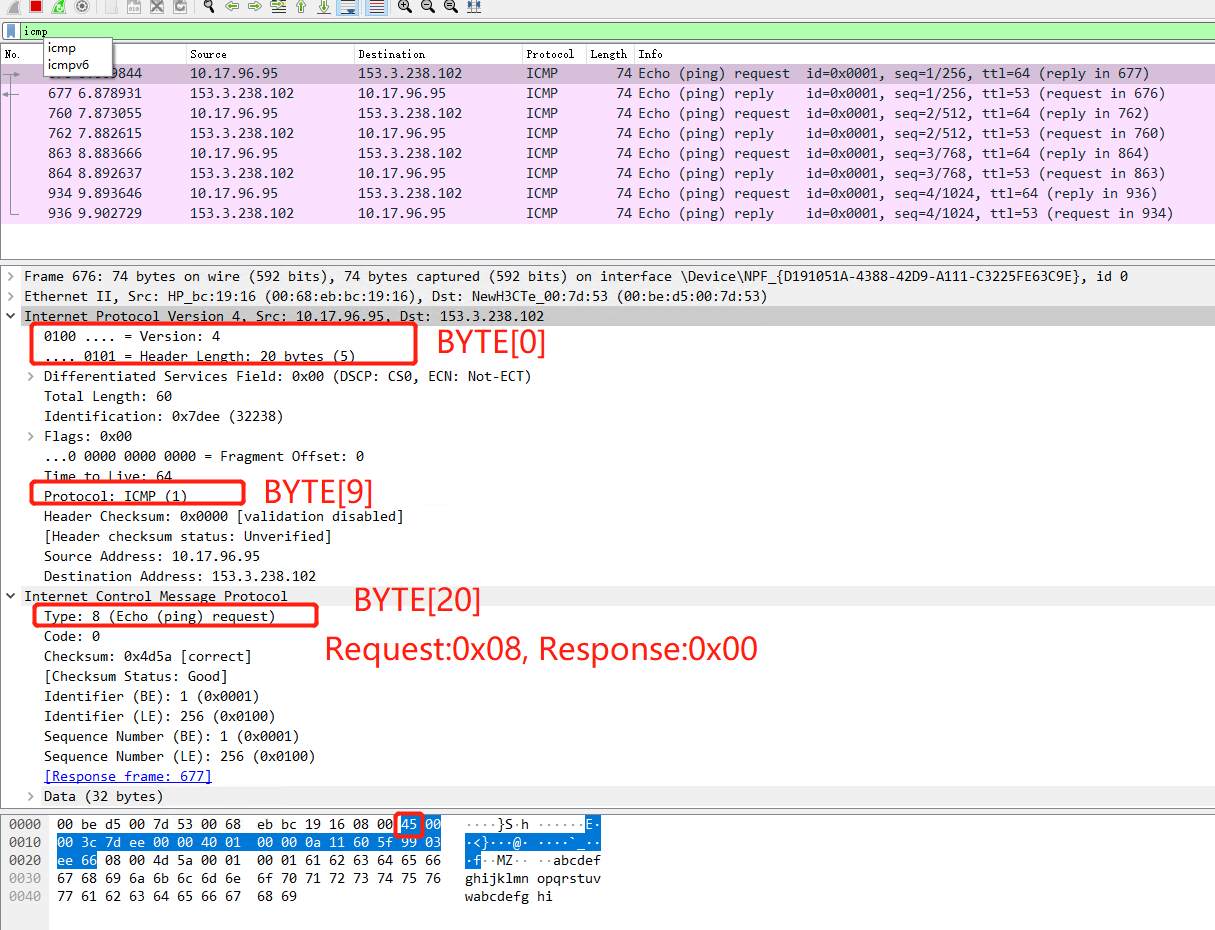


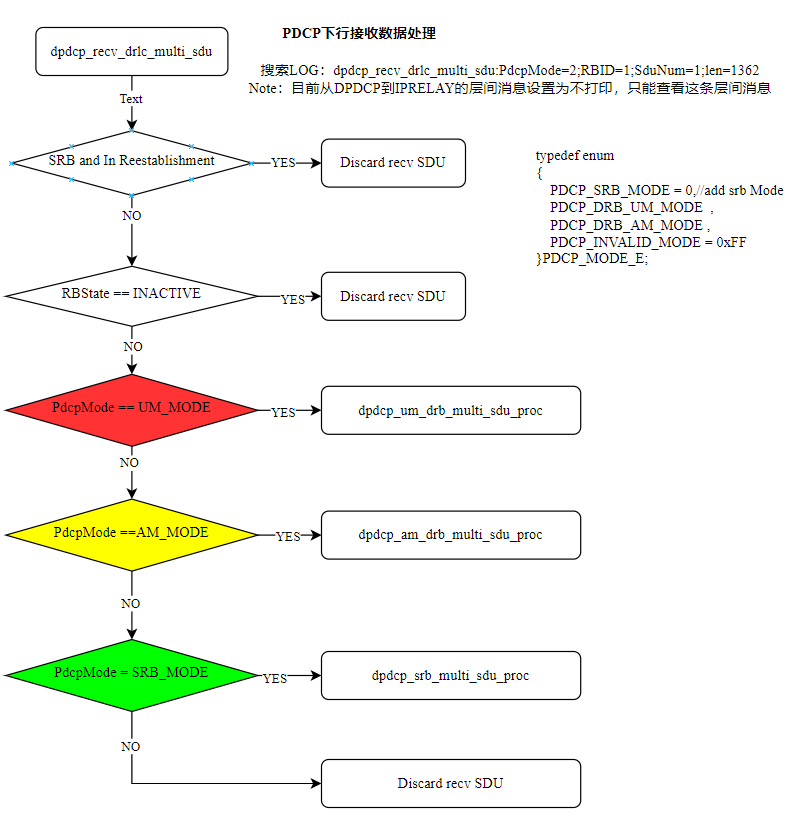
图 15 IP包中的Ping包格式

ulDataLen = ntohs(\*(U16 \*)(&ip\_data[2])) - ((ip\_data[0] & 0x0f) << 2) - ((ip\_data[32] & 0xf0) >> 2);

ulDatLen = IP包总长度-IP头部长度-TCP头部长度

如果ulDatLen为0，则表示没有用户数据。

### PDCP关键函数



#### dpdcp\_recv\_drlc\_multi\_sdu

DPDCP的所有DRB、SRB接收数据的总入口。

#### dpdcp\_srb\_data\_proc

PDCP的Mode为PDCP\_SRB\_MODE的单个包处理函数。

|  |
| --- |
| U16 dpdcp\_srb\_data\_proc(  U8 ucRBID,  U16 usRlcSduLen,  U16 usOffset,  U8 \*pucRlcSdu) |

参数说明：

* 1. ucRBID：当前SRB包对应的哪个Radio Bearer ID
  2. U16 usRlcSduLen,
  3. U16 usOffset,
  4. U8 \*pucRlcSdu)

主要的处理流程

如果收到了Security Mode Command，但是还没有收到安全配置消息RRC\_DPDCP\_SECURITY\_CFG\_REQ，则将收到的下行RRC消息暂存

#### DPDCP接收到下行数据

dpdcp\_recv\_drlc\_multi\_sdu

dpdcp\_am\_drb\_multi\_sdu\_proc

#### UDPCP收到上行包处理

U16 updcp\_recv\_data\_from\_iprelay(IpRelayEUpdcpDataDeliveryInd \*pstIpData)

核心函数

#### UDPCP缓存AP发送的上行数据

ZSP\_ITCM U16 updcp\_data\_buffer\_in(IpRelayEUpdcpDataDeliveryInd \*pstIpData)

代码中最大缓存1250个

#### UDPCP刷出 AP发送的上行数据

ZSP\_ITCM U16 updcp\_data\_buffer\_out(

U8 ucRBTblIdx,

U32 ulIndBuffOutLen,

U16 usMaxBuffOutNum)

#### UPDCP发送上行数据

ZSP\_ITCM U16 updcp\_multi\_drb\_data\_proc(L2\_UL\_PDCP\_SDU\_INFO\_S \*pstPdcpSduInd)

#### 统计函数

l2\_malloc\_outside\_ul

目前未打开统计，无作用

#### UPDCP重建处理

U16 updcp\_drb\_reestablishment(U8 ucRbTblIdx)

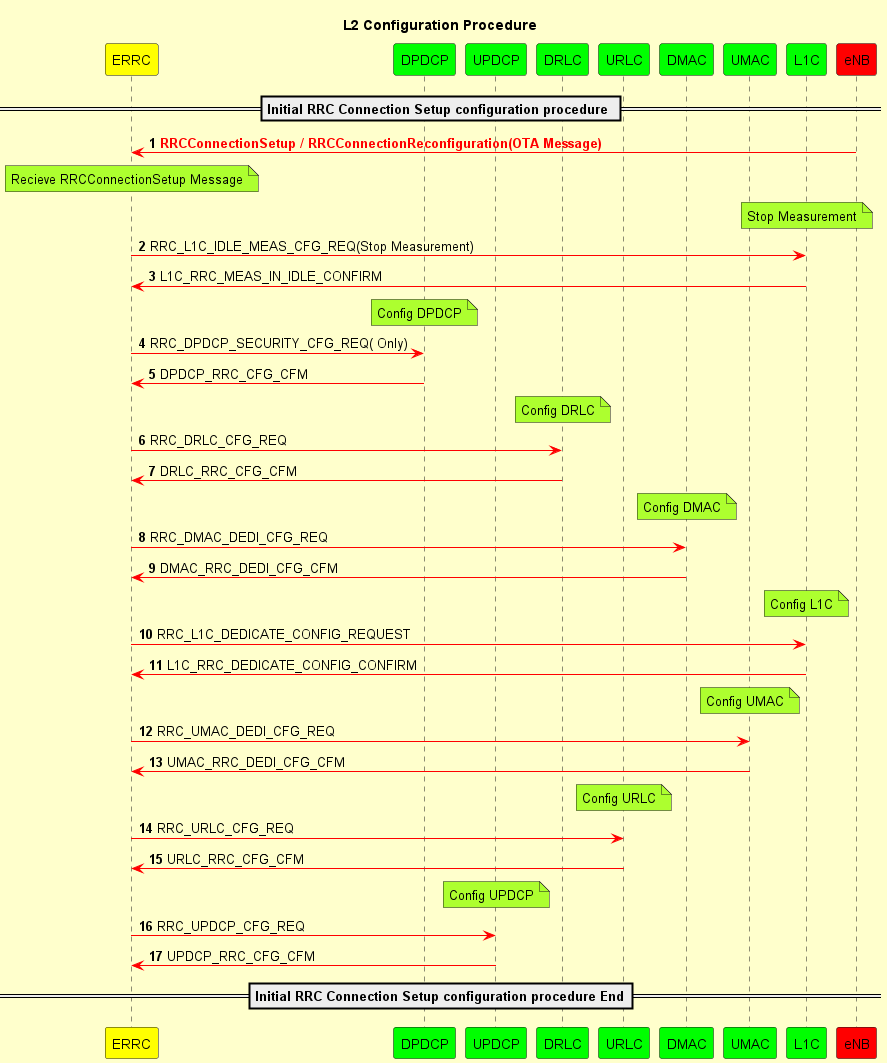
1. 如果输入的ucRbTblIdx对应的PDCP为AM模式，则将ucRbTblIdx对应的上行待发送SDU使用原密钥解密，然后重新挂入到链表中。
2. 如果是UM模式则重置TxSN和TxHFN。
3. 将ucRbTblIdx对应的SDU List遍历，对每个缓存的SDU进行加密和添加头部，发生到URLC中。
4. URLC处理错误则直接将当前节点删除，并更新前后结点指向逻辑。

使用新的

U32 updcp\_reverse\_decryption(U8 ucRbTblIdx)

如果是AM模式，则将ucRbTblIdx对应的上行待发送SDU使用原密钥解密，然后用新的密钥加密重新挂入到链表中。

### PDCP的配置过程



#### SRB配置

1． UDPCP配置

触发消息：RRC\_UPDCP\_CFG\_REQ

#### DRB配置

触发消息：RRC\_UPDCP\_CFG\_REQ

##### 初始配置

|  |
| --- |
| U16 updcp\_drb\_init\_cfg\_proc(U32 ulRBTabIdx, PDCP\_CFG\_S \*pstRrcCfgMsg)  参数：  ulRBTabIdx：配置DRB ID对应的RB context的索引号  pstRrcCfgMsg：DRB对应的PDCP配置指针 |

处理过程：

1. 重置DiscardTimer，若DiscardTimer有效则设置。
2. AM模式，则设置SN长度为12bit
3. UM模式则设置网络配置的

##### 重配

updcp\_drb\_recfg\_proc

#### DRB释放

UPDCP收到RRC\_UPDCP\_CFG\_REQ，该消息由RRCConnectionReconfiguration消息触发。

U16 updcp\_cfg\_release\_drb\_proc(RRC\_UPDCP\_CFG\_REQ\_MSG\_S \*pstRrcCfgReq)

释放步骤：

1. 每个DRB配置了一个Discard Timer，如果启动则停止。
2. 通知URLC删除DRB对应的缓存链表。
3. 删除UPDCP SDU链表。
4. 状态清空。TxSN、TxHFN、Discardtimer、SNFieldLen、RBState=INACTIVE

#### 安全配置

### 内存相关的函数操作

pstPduListNode = (DPDCP\_PDU\_S \*)pl\_l2\_malloc\_dl(sizeof(DPDCP\_PDU\_S));

pl\_l2\_malloc\_dl实际为pl\_malloc

**pl\_l2\_free\_dl**

### PDCP关键Log

g\_stDpdcpContext.ucSMCCfgFlag //

#### DPDCP接收到下行数据

|  |
| --- |
| 287632 INFO EDPDCP 0-24-172-9 23.11.23-15:57:11.258 dpdcp\_recv\_drlc\_multi\_sdu:PdcpMode=2;RBID=1;SduNum=1;len=1362  287633 INFO EDPDCP 0-24-172-9 23.11.23-15:57:11.258 dpdcp\_deliver\_sdu:RBID=1;len=1360 |

相关的参数

PdcpMode为枚举类型，具体枚举值如下。

typedef enum

{

PDCP\_SRB\_MODE = 0,//add srb Mode

PDCP\_DRB\_UM\_MODE, // 1

PDCP\_DRB\_AM\_MODE, // 2

PDCP\_INVALID\_MODE = 0xFF

}PDCP\_MODE\_E;

# RLC

## RLC TM模式

用于RRC还未建立时的RRC消息收发，使用SRB0承载发送，PDCP不参与发送和接收。

用于发送消息：

1. RRCConnectionRequest
2. RRCConnectionReestablishmentRequest

接收如下消息：

1. RRCConnectionSetup
2. RRCConnectionReject
3. RRCConnectionReestablishment
4. RRCConnectionReestablishmentReject

# MAC

参考文档：《TS 36.321 V9.2.0 Medium Access Control (MAC) protocol specification》

## MAC PDU

### MAC PDU的组成部分

1. MAC头部
2. 0或者多个MAC SDU（Service Data unit）
3. 0或者多个MAC CE（Control Element）
4. 可选的Padding

MAC PDU的头部有一个或多个MAC PDU的子头（Subheader）组成，每个子头对应一个MAC SDU/MAC CE/Padding。

### 不同的MAC子头

1. MAC PDU 子头由 R/R/E/LCID/F/L 六个标头字段组成，但 MAC PDU 中的最后一个子头和固定大小的 MAC CE除外。
2. MAC PDU 中的最后一个子头和固定大小 MAC CE的子头仅由四个标头字段 R/R/E/LCID （1字节）组成。
3. 与填充相对应的 MAC PDU 子头由四个标头字段 R/R/E/LCID 组成。



图 16 6域MAC子头



图 17 4域MAC子头

### MAC PDU的结构

DL-SCH与UL-SCH信道的MAC PDU结构，不包括透明MAC（Transparent MAC）、随机接入响应和MCH。

1. MAC PDU 子头的顺序与相应的 MAC SDU、MAC CE和Padding相同。
2. MAC CE总是放在任何 MAC SDU 之前。
3. Padding出现在 MAC PDU 的末尾，需要单字节或双字节填充时除外。填充可以是任何值，UE 应忽略它。在 MAC PDU 末尾进行填充时，允许使用零个或多个填充字节。
4. 需要单字节或双字节填充时，与填充相对应的一个或两个 MAC PDU 子头应放在 MAC PDU 的开头，任何其他 MAC PDU 子头之前。【一个或两个字节的Padding则直接的头部添加1个或2个Padding对应的子头即可，目的是不需要在最后一个Subheader中添加长度域】
5. 每个 UE 每个 TB 最多可传输一个 MAC PDU。每个 TTI 最多可传输一个 MCH MAC PDU。



图 18 MAC PDU结构（大于2个Padding的场景）

### 参数

#### LCID

LCID：Logical Channel ID，逻辑信道 ID 字段标识相应 MAC SDU 的逻辑信道实例或相应 MAC CE或填充的类型，下图中分别用于 DL-SCH、UL-SCH 和 MCH。MAC PDU 中包含的每个 MAC SDU、MAC CE或填充都有一个 LCID 字段。此外，如果需要单字节或双字节填充，但又无法通过在 MAC PDU 末尾进行填充来实现，则会在 MAC PDU 中包含一个或两个额外的 LCID 字段。LCID 字段大小为 5 比特；

表格 3 DL-SCH信道LCID取值

|  |  |
| --- | --- |
| Index | LCID values |
| 00000 | CCCH |
| 00001-01010 | Identity of the logical channel |
| 01011-11011 | Reserved |
| 11100 | UE Contention Resolution Identity |
| 11101 | Timing Advance Command |
| 11110 | DRX Command |
| 11111 | Padding |

表格 4 UL-SCH信道LCID取值

|  |  |
| --- | --- |
| Index | LCID values |
| 00000 | CCCH |
| 00001-01010 | Identity of the logical channel |
| 01011-11001 | Reserved |
| 11010 | Power Headroom Report |
| 11011 | C-RNTI |
| 11100 | Truncated BSR |
| 11101 | Short BSR |
| 11110 | Long BSR |
| 11111 | Padding |

表格 5 MCH信道LCID取值

|  |  |
| --- | --- |
| Index | LCID values |
| 00000 | MCCH (see note) |
| 00001-11100 | MTCH |
| 11101 | Reserved |
| 11110 | MCH Scheduling Information |
| 11111 | Padding |
| NOTE: If there is no MCCH on MCH, an MTCH could use this value. | |

#### L

L：Length（长度）字段表示相应 MAC SDU 或可变大小 MAC CE的长度（以字节为单位）。每个 MAC PDU 子头都有一个 L 字段，但最后一个子头和与固定大小 MAC CE相对应的子头除外。L 字段的占用的bit数由 F 字段指示。

#### F

F：Format（格式）字段表示长度字段的大小。每个 MAC PDU 子头有一个 F 字段，但最后一个子头和与固定大小的 MAC CE相对应的子头除外。F 字段的大小为 1 位。如果 MAC SDU 或大小可变的 MAC 控制元素的大小小于 128 字节，则 F 字段的值设置为 0，否则设置为 1；

#### E

E：Extension（扩展）字段是一个标志，表示 MAC 标头中是否存在更多的字段。E 字段设置为 "1 "表示至少有另一组 R/R/E/LCID 字段。E 字段设置为 "0 "表示 MAC SDU、MAC CE或填充从下一个字节开始；

通过www.DeepL.com/Translator（免费版）翻译

## MAC控制单元

MAC Control Element，简称MAC CE。

### BSR

BSR：Buffer Status Report MAC Control Elements，缓冲区状态报告MAC控制单元。由UE上报给eNodeB当前待发送的字节数。

BSR MAC CE有两种格式

1. 短 BSR 和截断 BSR 格式：一个 LCG ID 字段和一个相应的缓冲区大小字段（图 6.1.3.1-1）；或
2. 长 BSR 格式：四个缓冲区大小字段，分别对应 LCG ID #0 至 #3（图 6.1.3.1-2）。

BSR 格式由 MAC PDU 子标头标识，其 LCID 如表 6.2.1-2 所示。

LCG ID 和缓冲区大小字段定义如下：

- LCG ID： 逻辑通道组 ID 字段标识报告缓冲区状态的逻辑通道组。字段的长度为 2 位；

- 缓冲区大小： 缓冲区大小字段标识 MAC PDU 生成后逻辑通道组中所有逻辑通道的可用数据总量。数据量以字节数表示。它应包括 RLC 层和 PDCP 层可用于传输的所有数据；关于哪些数据应被视为可用于传输的定义分别在 [3] 和 [4] 中规定。 在计算缓冲区大小时，不考虑 RLC 和 MAC 标头的大小。该字段长度为 6 位。缓冲区大小字段的取值如表 6.1.3.1-1 所示。

通过www.DeepL.com/Translator（免费版）翻译

### C-RNTI MAC Control Element

### DRX Command MAC Control Element

### UE Contention Resolution Identity MAC Control Element

### Timing Advance Command MAC Control Element

### Power Headroom MAC Control Element

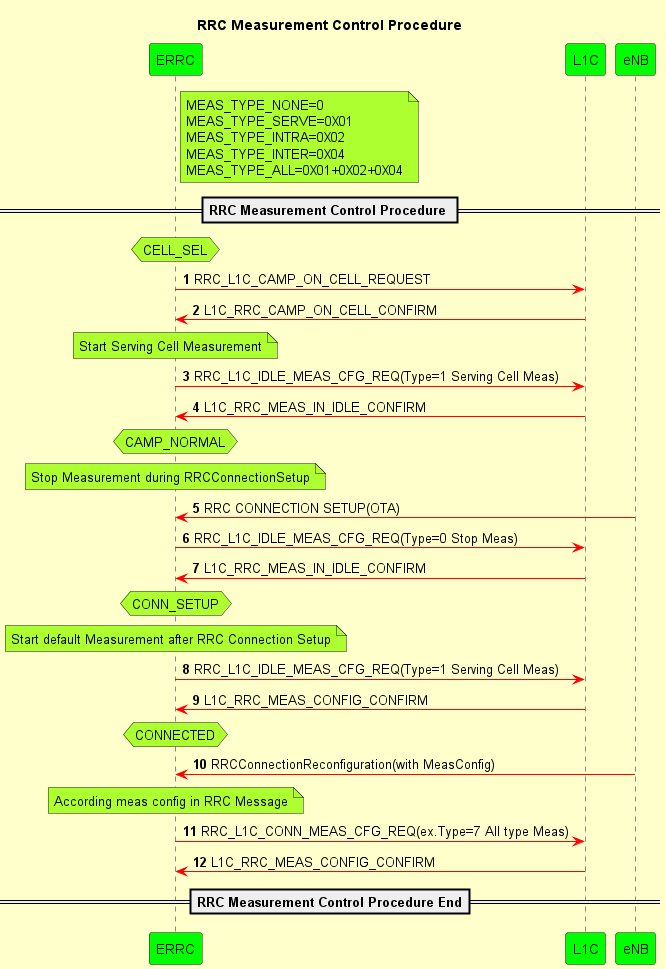
### MCH Scheduling Information MAC Control Element

# RRC

## 关于SRB

1. RRC连接建立前通道为SRB0，没有任何专用配置
2. RRC连接建立后的通道为SRB1，通过RRCConnectionSetup配置
3. NAS的Security Mode Command/Security Mode Complete流程结束后，网侧通过RRCConnectionReconfiguration配置SRB Identity为2时，就建立了SRB2，仅用于收发UL\_INFORMATION\_TRANSFER和DL\_INFORMATION\_TRANSFER

## 测量控制

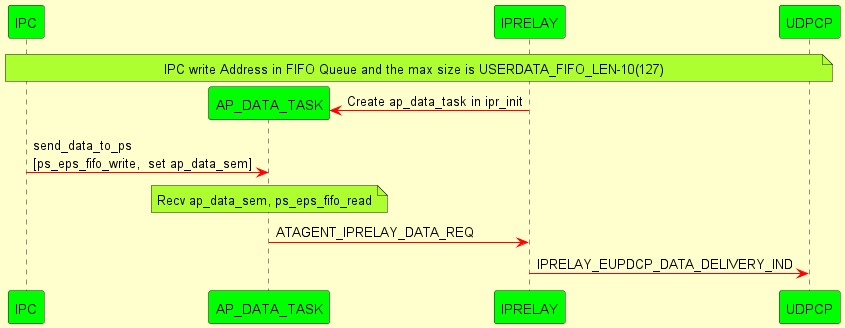


测量的启动：

1. 驻留到小区后，发起服务小区的测量启动。
2. 进入到连接态后，发起服务的小区的默认测量。
3. 在收到RRCConnectionSetup消息时停止测量。
4. 收到RRCConnectionReconfiguration中带有MeasConfig时，启动测量。

# IPRELAY

## 发送数据过程



### IPC发送数据处理

IPC将数据块的首地址写入到FIFO队列，并发送信号量ap\_data\_sem。

FIFO队列设置：ZSP\_DTCM static USER\_DATA\_FIFO\_T user\_fifo;

FIFO队列长度设置：USERDATA\_FIFO\_LEN-10 //#define USERDATA\_FIFO\_LEN (128)

写信号量函数：xSemaphoreGiveFromISR(ap\_data\_sem, &xHigherPriorityTaskWoken);

### AP\_DATA\_TASK的创建

在调用IPRELAY task的初始化函数void ipr\_init()时创建了一个ap\_data\_task。该task仅用于获取ap\_data\_sem信号量，并读取IPC设置的FIFO队列的地址，最后组合成消息ATAGENT\_IPRELAY\_DATA\_REQ并发送给IPRELAY。

获取信号量：xSemaphoreTake(ap\_data\_sem,portMAX\_DELAY);

### 上行数据的发送流程

#### 收到ATAGENT\_IPRELAY\_DATA\_REQ的处理流程

1. ZSP\_ITCM void IpRelaySession::DeliverIpPacket(AtAgentIpRelayDataSendReq \*tmpIpDataReq)
2. prepare\_updcp\_flow\_control();

作用：UPDCP流控函数，用于控制IPRELAY向UPDCP发送数据，当UPDCP还有较多的缓存空间时下发，当UPDCP缓存空间不够时停止向UPDCP下发。该函数目前生效！

1. TcpIpHeader::parseIP(U8 \*ipPacketPointer, U32 totalLength)

作用：解析当前IP包，获取原、宿IP地址、UDP/TCP端口、Identification、SPI等报文参数。

1. ZSP\_ITCM bool IpRelaySession::IpRelayTryDrvIPPktWithHdr(TcpIpHeader\* theTCPIPHeader, U32 pkSize,U32 offset,U8\* oriPacket, U8 cid )

作用：尝试发送IP包，其中第一个参数为解析后的IP/UDP/TCP头部信息

先匹配默认承载（使用IP或Cid），默认承载匹配不成功，则直接丢弃；

默认承载匹配成功，在匹配专用承载，专用承载匹配成功且DRB有效则直接发送出去；DRB无效则缓存并请求EMM发起Service Request。

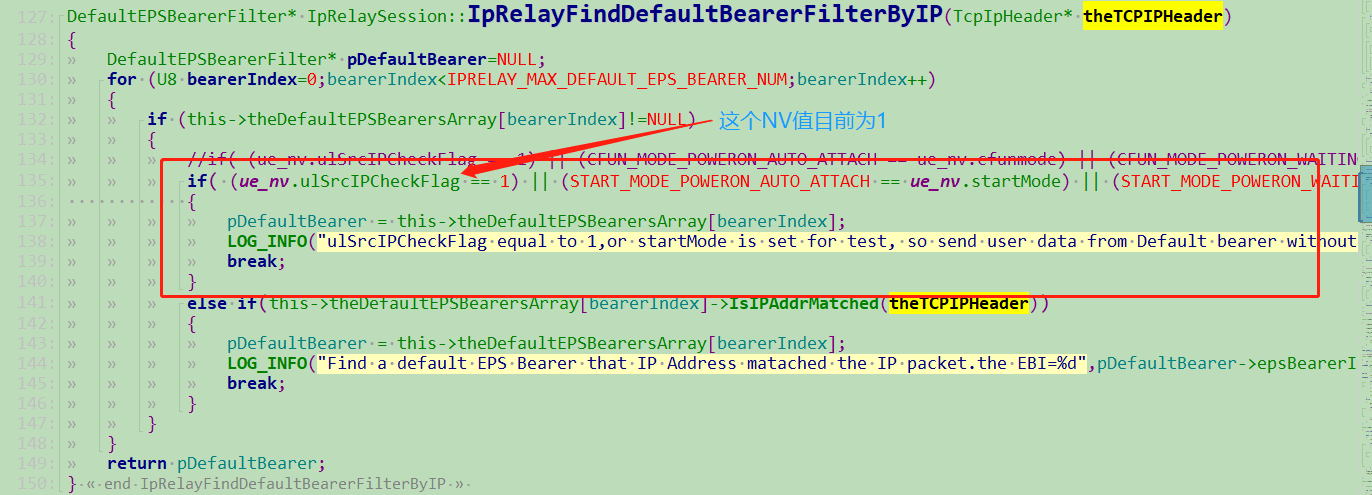
默认承载匹配成功，但专用承载匹配失败，但判断true == pDefaultEPSBearerFilter->hasTFT则直接丢弃（需要在专用承载上发送）

默认承载匹配成功，但专用承载匹配失败，但判断false== pDefaultEPSBearerFilter->hasTFT则直接发送处理。（无DRB在发起Servic Request）

1. DefaultEPSBearerFilter\* IpRelaySession::IpRelayFindDefaultBearerFilterByIP(TcpIpHeader\* theTCPIPHeader)

根据IP报文中的源IP匹配是否存在匹配的默认承载。

Note：在该函数中，默认会取第一个默认承载，而不会继续根据UE IP进行匹配。



1. ZSP\_ITCM DefaultEPSBearerFilter\* IpRelaySession::IpRelayFindDefaultBearerFilterByCid(U8 cid)

如果与报文一起传入了cid则会根据cid进行匹配，得到默认EPS承载过滤器。

1. ZSP\_ITCM DedicatedEPSBearerFilter\* DefaultEPSBearerFilter::FindMatchedEpsBearer(TcpIpHeader\* theTcpIpHeader,EpsPacketFilterDirection pkDirection)

如果根据TCPIP头部和cid都没能找到对应的默认承载过滤器DedicatedEPSBearerFilter，则使用已经存储的TFT专用承载的DedicatedEPSBearerFilter进行匹配。

Note：专用承载必须要通过TFT对上行包进行过滤确认允许后才能发送。

1. void IpRelaySession::IpRelayBufferIpPacket(U32 pkSize,U32 offset,U8\* oriPacket,U8 epsBearerId)

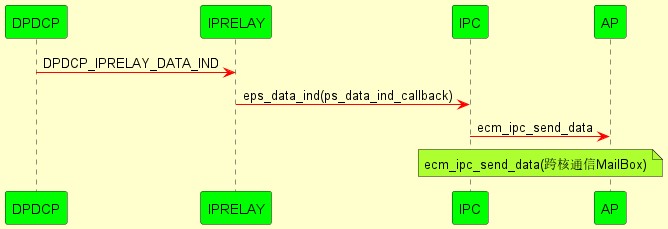
作用：缓存不能及时发送的包。缓存的总大小为MAX\_BUFFERED\_DATA\_IN\_BYTES 1000000（1M字节）

#### 收到LOOPBACK\_IPRELAY\_DATA\_IND消息的处理

### TFT检查

ZSP\_ITCM bool IpRelaySession::IpRelayTryToDeliveryIPPacket(TcpIpHeader\* theTCPIPHeader, U32 pkSize,U32 offset,U8\* oriPacket, U8 cid )

## 接收数据过程



### AP接收数据后的内存释放

在IPC task的处理中（static void ipc\_task\_entry(void \*parameters)），AP通过通知IPC释放CP接收下行数据时申请的内存。

具体方式为，AP通知IPC，将需要释放的内存地址写入到RingBuffer中，IPC读取该Buffer中的地址依次释放。具体处理如下函数处理。

