

ICS 33.060

CCS M10/29

YJ

中华人民共和国应急管理行业标准

YJ/T XXXXX—XXXX

应急指挥无线宽带自组网系统技术规范 第
3 部分：数据链路层

Technical specifications for emergency command wireless broadband ad hoc network
system: Data link layer

(征求意见稿)

XXXX - XX - XX 发布

XXXX - XX - XX 实施

中华人民共和国应急管理部 发布

目 次

前 言	VII
引 言	8
1 范围	9
2 规范性引用文件	9
3 术语和定义	9
4 缩略语	9
5 概述	10
5.1 介绍	10
5.2 总体架构	10
6 RRC	11
6.1 概述	11
6.1.1 介绍	11
6.1.2 架构	11
6.1.2.1 节点分类	11
6.1.2.2 节点状态	11
6.1.2.3 无线承载	11
6.1.2.4 复帧结构	12
6.1.3 服务	12
6.1.3.1 提供给上层的服务	12
6.1.3.2 L2 提供给 RRC 的服务	12
6.1.4 功能	12
6.2 过程	12
6.2.1 概述	12
6.2.2 系统信息	13
6.2.2.1 系统消息介绍	13
6.2.2.2 系统消息调度	13
6.2.2.3 系统消息发送与接收	13
6.2.2.4 MasterInformationBlock 消息发送与接收	13
6.2.2.5 SystemInformationBlockType1 消息发送与接收	14
6.2.2.6 SystemInformationBlockType2 消息发送与接收	14
6.2.3 网络搜索与部署	14
6.2.3.1 网络搜索	14
6.2.3.2 网络部署	14
6.2.4 连接控制	14
6.2.4.1 节点入网	15
6.2.4.2 节点退网	16

6.2.4.3 信令发送	17
6.2.5 无线资源分配	17
6.2.5.1 资源分类	17
6.2.5.2 资源分配	17
6.2.5.3 资源释放	18
6.2.6 邻节点检测	18
6.2.6.1 邻节点发现	18
6.2.6.2 邻节点删除	18
6.2.7 参数变更	19
6.2.7.1 参数广播	19
6.2.7.2 参数生效	20
6.2.8 子网分裂与合并	20
6.2.8.1 子网分裂	20
6.2.8.2 子网检测	20
6.2.8.3 子网合并	20
6.3 协议数据单元，格式以及参数	21
6.3.1 概述	21
6.3.2 RRC 信息	22
6.3.2.1 通用消息结构	22
6.3.2.2 消息定义	24
6.3.3 RRC 信息元素	29
6.3.3.1 无线资源信息元素	29
6.3.3.2 路由信息元素	31
6.3.3.3 其他信息元素	31
6.3.4 RRC 多样性和类型常量值	32
6.4 定时器	33
6.5 默认无线配置	33
6.5.1 SRB 配置	33
6.5.1.1 SRB1	33
6.5.1.2 SRB2	34
6.5.2 DRB 配置	34
6.5.2.1 DRB3	34
6.5.2.2 DRB4	34
6.5.2.3 DRB5	35
6.5.2.4 DRB7	35
7 PDCP	36
7.1 概述	36
7.1.1 PDCP 架构	36
7.1.2 PDCP 实体	36
7.1.3 业务	37
7.1.3.1 提供给上层的服务	37
7.1.3.2 从下层获得的服务	37

7.1.4 功能	37
7.1.5 可传数据	38
7.2 PDCP 过程	38
7.2.1 PDCP 数据传输过程	38
7.2.1.1 发送数据传输过程	38
7.2.1.2 接收数据传输过程	38
7.2.2 PDCP 丢弃	40
7.2.3 加密和解密	41
7.2.4 完整性保护及确认	41
7.2.5 未知的,意外的以及错误的协议数据的处理	42
7.3 协议数据单元,格式及参数	42
7.3.1 协议数据单元	42
7.3.1.1 PDCP 数据 PDU	42
7.3.1.2 PDCP 控制 PDU	42
7.3.2 格式	42
7.3.2.1 综述	42
7.3.2.2 控制平面 PDCP Data PDU	42
7.3.2.3 具有长 PDCP SN 值 (12bit) 的用户平面 PDCP Data PDU	43
7.3.2.4 具有短 PDCP SN 值 (7bit) 的用户平面 PDCP Data PDU	43
7.3.3 参数	44
7.4 变量,常量及定时器	45
7.4.1 状态变量	45
7.4.2 定时器	46
7.4.3 常量	46
8 RLC	46
8.1 概述	46
8.1.1 RLC 结构	46
8.1.1.1 UM Mode RLC 实体	46
8.1.1.2 发送 UM RLC 实体介绍	47
8.1.1.3 接收 UM RLC 实体介绍	47
8.1.1.4 AM Mode RLC 实体	47
8.1.2 服务	49
8.1.2.1 提供给上层的服务	49
8.1.2.2 期望从下层获得的服务	49
8.1.3 功能	49
8.2 过程	49
8.2.1 数据传输过程	49
8.2.1.1 UM 数据传输	49
8.2.1.2 AM 数据传输	51
8.2.2 ARQ 过程	53
8.2.2.1 重传	53
8.2.2.2 轮询	53
8.2.2.3 状态报告。	55

8.2.3 SDU 丢弃过程	55
8.2.4 Reset 过程	56
8.2.4.1 概述	56
8.2.4.2 RLC 发起的 RESET	56
8.2.5 处理未知的、未预见的及错误的协议数据	56
8.3 协议数据单元、格式及参数	56
8.3.1 协议数据单元	56
8.3.1.1 RLC 数据 PDU	56
8.3.1.2 RLC 控制 PDU	56
8.3.2 格式和参数	57
8.3.2.1 格式	57
8.3.2.2 参数	63
8.4 变量、常量和定时器	67
8.4.1 状态变量	67
8.4.2 常量	69
8.4.3 定时器	69
8.4.4 可配参数	69
9 MAC	69
9.1 概述	69
9.1.1 介绍	69
9.1.2 MAC 架构	69
9.1.2.1 MAC 实体	70
9.1.3 服务	70
9.1.3.1 为上层提供的服务	70
9.1.3.2 期待从物理层得到的服务	70
9.1.4 MAC 的功能	70
9.1.5 信道结构	71
9.1.5.1 传输信道	71
9.1.5.2 逻辑信道	71
9.1.5.3 传输信道到逻辑信道的映射	71
9.1.5.4 传输信道到物理信道的映射	72
9.2 MAC 过程	72
9.2.1 随机接入过程	72
9.2.1.1 随机接入过程初始化	72
9.2.1.2 随机接入资源选择	73
9.2.1.3 随机接入前导码传输	73
9.2.1.4 随机接入响应接收	73
9.2.1.5 随机接入过程完成	74
9.2.2 BCCH 数据传输	74
9.2.3 TCH 数据传输	74
9.2.3.1 HARQ 操作	74
9.2.3.2 逻辑信道优先级	75

9.2.4 BCH 的接收.....	75
9.2.5 MAC reset.....	75
9.2.6 对于未知, 预料以外以及错误数据的处理	75
9.3 协议数据单元, 格式和参数	76
9.3.1 协议数据单元	76
9.3.1.1 概述	76
9.3.1.2 MAC PDU (TCH, 除了随机接入响应)	76
9.3.1.3 MAC PDU (随机接入响应)	77
9.3.2 格式和参数	78
9.3.2.1 TCH (除 RAR)	78
9.3.2.2 随机接入响应的 MAC 头	78
9.3.2.3 随机接入响应 MAC 净荷	79
9.4 变量和常量.....	79
9.4.1 RNTI 值	79
参 考 文 献.....	80

前 言

本文件是应急指挥无线宽带自组网系统技术规范系列标准之一。该系列标准文件的结构及名称预计如下：

- 应急指挥无线宽带自组网系统技术规范 第1部分：总体要求；
- 应急指挥无线宽带自组网系统技术规范 第2部分：物理层；
- 应急指挥无线宽带自组网系统技术规范 第3部分：数据链路层；
- 应急指挥无线宽带自组网系统技术规范 第4部分：网络层。

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由应急管理部科技和信息化司提出。

本文件由全国应急管理 with 减灾救灾标准化技术委员会（SAC/TC 307）归口。

本文件起草单位：

本文件主要起草人：

引 言

为规范应急指挥无线宽带自组网系统的技术体制，实现不同供应商提供的设备之间的互联互通，满足应急宽带自组网能够快速部署、自由组网的要求，推动应急管理体系的无线通信数字化建设，特制定本文件。

应急指挥无线宽带自组网系统系列标准拟包含如下部分：

- 第1部分：总体要求。目的在于确定应急指挥无线宽带自组网的技术体制，设备的功能、性能，设备的环境适应性等基本要求；
- 第2部分：物理层。目的在于规范了自组网设备间承载数据链路层传输通道的信息并由天线收发的基本技术，包括发现组网、链路适配、功率控制、编码和调制传输等；
- 第3部分：数据链路层。目的在于规范了应急指挥无线宽带自组网系统技术规范的数据链路层的协议，包括无线资源控制协议（RRC）、分组数据汇聚协议（PDCP）、无线链路控制协议（RLC）、媒体接入控制协议（MAC）；
- 第4部分：网络层。目的在于规范了应急指挥无线宽带自组网系统空中接口网络层协议功能，包括网络层协议架构与功能、路由管理、消息和消息字段定义等内容。

应急指挥无线宽带自组网系统技术规范 第3部分：数据链路层

1 范围

本文件规定了应急指挥无线宽带自组网系统技术规范的数据链路层的协议，包括：无线资源控制协议（RRC）、分组数据汇聚协议（PDCP）、无线链路控制协议（RLC）、媒体接入控制协议（MAC）。

本文件适用于基于LTE TDD实现的应急指挥无线宽带自组网系统。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

逻辑主控节点 Logical master node

也称为主控节点，负责节点的固定资源的分配，主控节点与物理层节点无绑定关系。

非主控节点 Non master node

网路中的节点，除了主控节点，剩余的节点都是非主控节点。

节点标志 NODE ID

使用节点固定资源对应的序号作为节点的ID。

预留接入资源 Reserve access resources

由被接入节点在收到MSG1之后预分配一个空闲资源给新接入节点临时使用的资源，通过RAR中的Temporary C-RNTI获取。

4 缩略语

下列缩略语适用于本文件。

AM 确认模式（Acknowledged Mode）

ARQ 自动重传请求（Automatic Repeat Request）

BCCH 广播控制信道（Broadcast Control Channel）

BCH 广播信道（Broadcast Channel）

CQI 信道质量指示（Channel Quality Indicator）

LCG 逻辑信道组（Logical Channel Group）

MAC 媒体接入控制（Medium Access Control）

PDCP 分组数据汇聚协议（Packet Data Convergence Protocol layer）

QoS 服务质量（Quality of Service）

RB 无线承载（Radio Bearer）

RF 无线帧（Radio Frame）

RLC	无线链路控制 (Radio Link Control)
RNTI	无线网络临时标识 (Radio Network Temporary Identifier)
RRC	无线资源控制 (Radio Resource Control)
SAP	服务接入点 (Service Access Point)
SI	系统信息 (System Information)
SIB	系统信息块 (System Information Block)
SI-RNTI	系统信息 RNTI (System Information RNTI)
SON	自组网网络 (Self-Organizing Network)
SRB	信令无线承载 (Signalling Radio Bearer)
TB	传输块 (Transport Block)
TM	透明模式 (Transparent Mode)
UM	非确认模式 (Unacknowledged Mode)

5 概述

5.1 介绍

本文档的结构如下：

第 6 节主要描述 RRC 协议模型，介绍协议及相关协议过程。

第 7 节主要描述 PDCP 协议模型，介绍协议架构以及相关协议过程。

第 8 节主要描述 RLC 协议模型，介绍协议架构以及相关协议过程。

第 9 节主要描述 MAC 协议模型，介绍协议架构以及相关协议过程。

5.2 总体架构

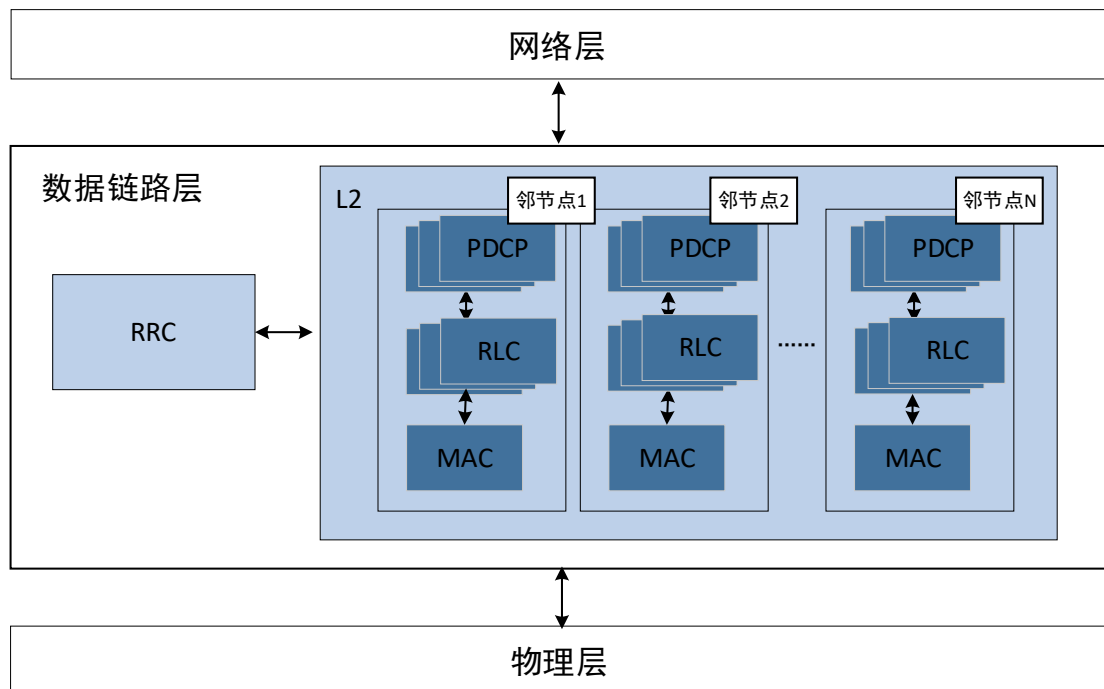


图 1 数据链路层架构图

如图 1 网络层架构图所示，数据链路层位于网络层和物理层之间，主要有 RRC 和 L2 两大部分构

成，RRC 主要负责节点入网、退网以及连接控制等功能，L2 中包含多个实体，每个实体包含 PDCP、RLC、MAC 三部分，主要负责数据的接收和发送。

6 RRC

6.1 概述

6.1.1 介绍

本章节主要阐述 RRC 模块，包括：RRC 的架构、RRC 相关的协议流程、RRC 消息的格式等。

6.1.2 架构

6.1.2.1 节点分类

网络中的节点分为两类：主控节点与非主控节点。

主控节点：负责是否允许新节点入网的判断以及固定资源的管理。主控节点并不固定，当主控节点退网后，按照 Sec6.2.4.2 中的描述，需要选择非主控节点成为新的主控节点。

非主控节点：除了主控节点，其余的节点都是非主控节点。

6.1.2.2 节点状态

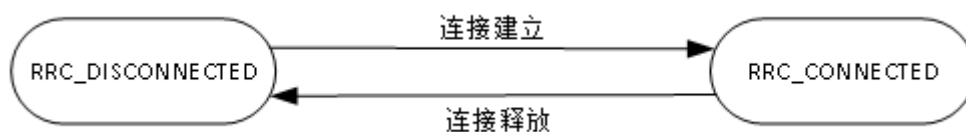


图 2 节点状态转换

节点状态转换如图 2 所示，节点有两种状态：RRC_CONNECTED 状态和 RRC_DISCONNECTED 状态，如图 2 所示。当建立了 RRC 连接，节点进入 RRC_CONNECTED 状态。在 RRC_CONNECTED 状态下，RRC 执行如下动作：

- 递交/发送来自上层的路由通告消息；
- 信令消息的发送、接收、转发；
- 系统消息的发送与接收；
- 无线资源维护；
- 邻节点检测；
- 邻子网的发现与合并。

如果节点退网释放 RRC 连接，则进入 RRC_DISCONNECTED 状态。在 RRC_DISCONNECTED 状态下，RRC 执行如下动作：

- 网络搜索或者网络部署；
- 建立连接；

6.1.2.3 无线承载

“信令无线承载”（SRB）仅用于 RRC 消息传输的无线承载（RB），定义如下 SRB：

- SRB1 用于 RRC 消息（除了测量报告），使用 DCCH 逻辑信道；

——SRB2 用于 RRC 测量报告消息，使用 DCCH 逻辑信道。

6.1.2.4 复帧结构

复帧结构如图 3 所示，一个无线帧为 5ms，一个复帧有 32 个无线帧组成，共 160ms。

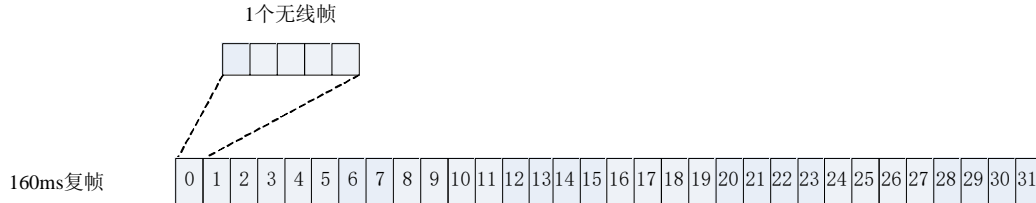


图 3 复帧结构

6.1.3 服务

6.1.3.1 提供给上层的服务

RRC协议提供给上层的服务如下：

- 广播网络层的路由通告信息；
- 上报链路状况。

6.1.3.2 L2 提供给 RRC 的服务

L2提供给RRC的主要服务：

- PDCP：完整性保护和加密；
- RLC：信息的可靠与按序传输，支持分段和串接；
- MAC：接入流程。

至于由分组数据汇聚协议层（PDCP）（例如完整性和加密）提供的服务，其详细的信息在Sec 6 PDCP中给出。由无线链路控制层（例如RLC模式）提供的服务，其在Sec 7RLC中有规定。由媒介接入控制层（例如逻辑信道）提供的服务，其详细的信息在Sec 8 MAC中有给出。

6.1.4 功能

RRC协议包括如下主要的功能：

- 广播系统信息，包括 MIB 信息、SIB1 路由信息、SIB2 全网节点相关信息；
- 网络搜索，包含节点的搜网、布网；
- RRC 连接控制，包含节点入网、退网；
- 无线资源分配，包括资源的分配和释放；
- 邻节点检测，包括邻节点的发现和删除；
- 参数变更，包含参数广播以及生效；
- 子网分裂与合并，包含子网的分裂、子网检测以及子网合并。

6.2 过程

6.2.1 概述

根据主要的功能：系统信息，网络搜索，连接控制，无线资源分配，邻区检测，参数变更，子网分裂与合并，提出相应的流程要求。注意：由 RRC 对接收的消息进行按序处理，即在开始对后面的消息进行处理之前，应该完成对早前接收到的消息的处理；在每个子章节中按照过程描述中的顺序进行执行。

6.2.2 系统信息

6.2.2.1 系统消息介绍

系统信息分成 MasterInformationBlock(MIB)和多个 SystemInformationBlocks (SIBs)。MIB 包括有限个最重要、最常用的传输参数，其需要从该网络中获得其它的信息，其在 BCH 上进行传输。SystemInformationBlockType1 是节点的路由信息，SystemInformationBlockType2 包含全网节点的相关信息，所有的 SI 消息是在 TCH 上进行传输。

6.2.2.2 系统消息调度

系统消息只能在本节点占用的无线资源上进行发送。MIB 在子帧 0 进行发送。SystemInformationBlockType1 根据网络层的发送周期进行发送。SystemInformationBlockType2 在一个复帧周期内至少发送一次，当该消息的信息元素发生变化时也需要发送。SIB1 和 SIB2 可以在任意子帧上发送，但是不能在同一子帧上发送。使用 SI-RNTI 处理所有的 SI 消息。

6.2.2.3 系统消息发送与接收

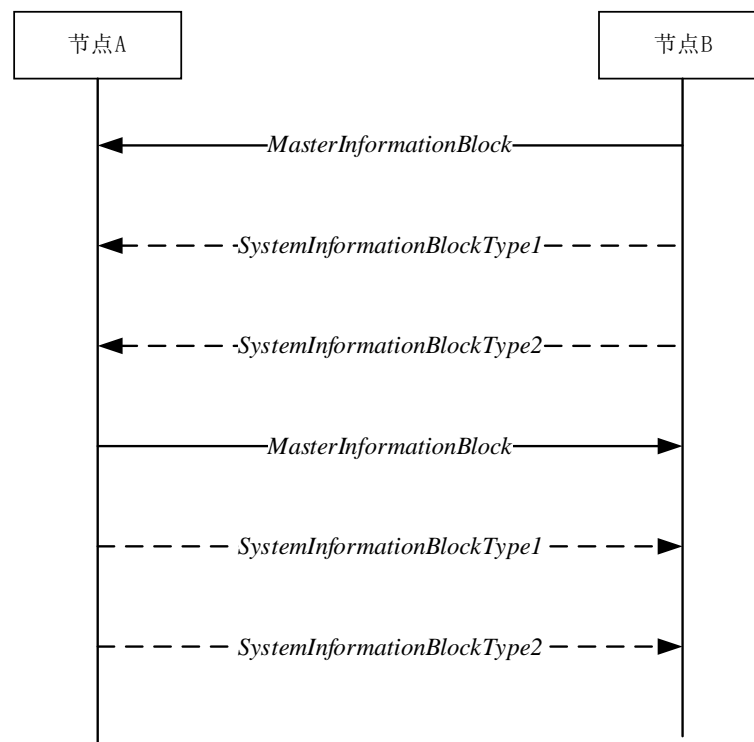


图 4 节点间系统消息发送与接收

如图 4 节点应用系统消息获取流程来进行系统消息的接收和发送。节点自开机搜网后，开始通过系统消息获取流程进行系统消息的接收，在 RRC 进入 RRC_CONNECTED 状态后进行系统消息的广播发送。

6.2.2.4 MasterInformationBlock 消息发送与接收

发送 MIB 消息，节点将：

- 1)按照MIB消息结构构造消息并在指定的子帧上进行发送；

2)MAC-I 的填写: 针对用户预配置的码流, 使用用户预配置的密钥, 经过 eia1 算法生成 MAC-I。
接收到 MIB 消息, 节点将:

1)针对用户预配置的码流, 使用用户预配置的密钥, 经过eia1算法生成MAC-I, 判断生成的MAC-I 与MIB消息中macI是否一致, 如果一致, 则认为MIB完整性保护检查通过;

2) 存储 MIB 中携带的其他网络配置参数:

1)否则:

2)丢弃该 MIB 消息。

6.2.2.5 SystemInformationBlockType1 消息发送与接收

节点收到网络层发送的路由消息后, 节点将:

1)按照SystemInformationBlockType1消息结构构造消息并进行发送;

接收到 SystemInformationBlockType1 消息, 节点将:

1)将消息中的NodeRouteInformation发送给网络层。

6.2.2.6 SystemInformationBlockType2 消息发送与接收

节点在驻留后需要周期性发送 SystemInformationBlockType2。

接收到 SystemInformationBlockType2 消息, 相关操作在后续相关流程或消息格式描述中进行阐述。

6.2.3 网络搜索与部署

当节点开机后需要进行网络搜索或者网络部署过程。

6.2.3.1 网络搜索

网络搜索流程是节点对指定的频点列表进行搜索, 并找到合适的网络的进行驻留的过程。合适网络的判断准则:

1)接收到MIB, 并且MIB完整性保护检查通过;

1)接收到SystemInformationBlockType2, 且其中的son-Identity相同。

如果节点找到合适的网络, 则驻留并进行连接建立。

6.2.3.2 网络部署

网络部署是指节点在指定的频点列表中挑选合适的频点并驻留的过程。如果在网络搜索中没有找到合适的网络, 节点进行网络部署。节点布网成功后进入 RRC_CONNECTED 状态:

1)选择任意空闲资源作为本节点的固定资源;

1)为本节点配置广播承载;

1)通知网络层入网成功;

1)通知底层入网成功;

1)发送MIB和SystemInformationBlockType2消息。

6.2.4 连接控制

节点在通过搜网找到合适的网络驻留后, 需要进行节点入网流程才能与网络中的节点建立连接。节点的连接建立是与邻节点 SRBs 和 DRBs 的建立。当节点远离周围的邻节点时, 会断开连接。图 5 是各节点一个新接入节点接入到网络的信令流程, 如果被接入节点是主控节点, 则图 5 中间的两个节点(被

节点节点、非主控节点) 的流程可以省略。

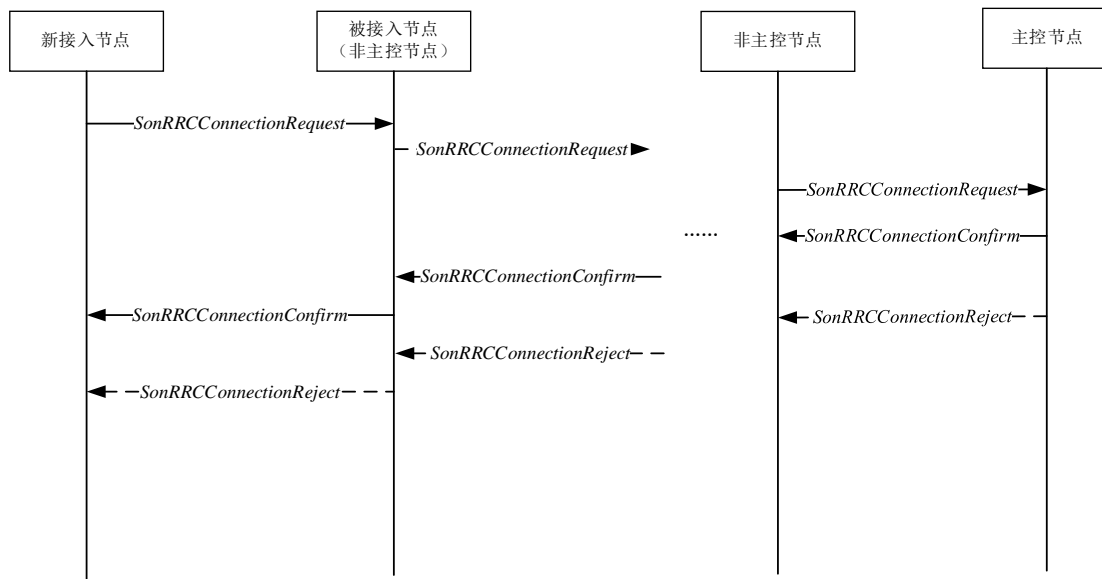


图 5 接入流程（被接入节点不是主控节点的）

6.2.4.1 节点入网

节点决定发起入网流程时，是以 SystemInformationBlockType2 中的 currentNodeAddress 为被接入节点启动接入流程：

- 1) 为本节点配置广播承载；
- 1) 通知MAC发起接入，并等待被接入节点分配的预留接入资源；
- 1) 启动定时器 T_{Msg2} ；
 - 2) 收到被接入节点分配的预留接入 Radio Frame 后，则停止定时器 T_{Msg2} ；
 - 2) 对接入节点进行 SRBs 和 DRBs 配置；
 - 2) 构造 SonRRCConnectionRequest 消息，并将 SonRRCConnectionRequest 消息在被接入节点分配的预留接入的 Radio Frame 上传输给被接入节点；
 - 2) 启动定时器 T_{RrcRsp} ，等待 SonRRCConnectionConfirm；
 - 3) 收到 SonRRCConnectionConfirm，停止定时器 T_{RrcRsp} ；
 - 4) SonRRCConnectionConfirm 中 occupiedRF 为主控节点分配给本节点的 NODE ID，本节点入网成功；
 - 5) 广播 MIB 和 SystemInformationBlockType2 消息。
 - 3) 收到 SonRRCConnectionReject，停止定时器 T_{RrcRsp} ；
 - 4) 节点入网失败，发起网络搜索流程。

对于被接入节点，在 RRC_CONNECTED 状态下，收到 MSG1 后的处理流程如下：

- 1) 选择空闲资源作为预留接入资源并通知MAC；
- 1) 对新接入节点进行SRBs和DRBs配置；
- 1) 启动定时器 T_{RrcReq} ，等待SonRRCConnectionRequest；

- 2) 收到 SonRRCConnectionRequest, 停止定时器 T_{RrcReq} ;
- 3) 如果被接入节点不是主控节点, 则将 SonRRCConnectionRequest 转发给主控节点:
 - 4) 启动定时器 T_{RrcRsp} , 等待 SonRRCConnectionConfirm;
 - 5) 收到 SonRRCConnectionConfirm, 停止定时器 T_{RrcRsp} , 认为新接入节点入网成功;
 - 6) 新接入节点的固定资源为 SonRRCConnectionConfirm 中的 occupiedRF;
 - 6) 将 SonRRCConnectionConfirm 转发给新接入节点;
 - 6) 构造 SystemInformationBlockType2 消息并广播。
 - 5) 收到 SonRRCConnectionReject, 停止定时器 T_{RrcRsp} , 认为新接入节点入网失败;
 - 6) 将 SonRRCConnectionReject 发送给新接入节点;
 - 6) 释放对应节点的 SRBs 和 DRBs 配置;
- 3) 被接入节点是主控节点:
 - 4) 节点入网准则: 网络没有达到最大节点个数, 并且 SonRRCConnectionRequest 中携带的 pre-occupiedRF 没有被分配给其他节点占用。
 - 4) 根据节点入网准则判定新节点可以入网:
 - 5) 分配 pre-occupiedRF 对应的 Radio Frame 作为新接入节点的固定资源;
 - 5) 构造 SonRRCConnectionConfirm 发送给新接入节点;
 - 5) 构造 SystemInformationBlockType2 并广播, 其中的 messageSequenceNumber +1;
 - 4) 否则:
 - 5) 构造 SonRRCConnectionReject, 发送给新接入节点。

主控节点为非被接入节点, 收到 SonRRCConnectionRequest:

- 1) 根据节点入网准则判定可以入网:
 - 2) 分配 pre-occupiedRF 对应的 Radio Frame 作为新接入节点的固定资源;
 - 2) 构造 SonRRCConnectionConfirm 发送给被接入节点;
 - 2) 构造 SystemInformationBlockType2 并广播, 其中的 messageSequenceNumber +1;
- 1) 否则:
 - 2) 构造 SonRRCConnectionReject, 发送给被接入节点。

节点入网后, 需要通知网络层本节点以及邻节点的连接状态。

节点入网后会收到 SystemInformationBlockType2 消息, 根据其中携带的到每个节点的跳数 metric, 更新本节点到全网其他节点的 metric, 如果 metric 发生变化:

- 1) 构造 SystemInformationBlockType2 并广播。

6.2.4.2 节点退网

当节点执行关机操作时:

- 1) 发送 SystemInformationBlockType2, 携带 currentNodeLeaveIndication 为 TRUE。
- 当节点收到 SystemInformationBlockType2, 其中 currentNodeLeaveIndication 为 TRUE:
- 1) 构造 SystemInformationBlockType2 并广播。
- 当节点判定到某个节点的 metric 超过网络最大节点个数时, 则认为该节点退网。
- 1) 当主控节点判定某个节点退网时, 则操作流程如下:

- 2)释放该节点的固定资源;
 - 2)通知底层节点退网;
 - 2)构造 SystemInformationBlockType2 并广播, 其中的 messageSequenceNumber +1;
 - 1)当非主控节点判定主控节点退网时, 原主控节点所占的固定资源后最近一个被固定占用的节点升级为新的主控节点:
 - 2)节点判定自己被选举为新的主控节点:
 - 3) 节点变为主控节点, 删除原主控节点的固定资源;
 - 3) 根据 sec5.2.8 流程变更全网节点的 PCI 参数;
 - 3) 构造 SystemInformationBlockType2 并广播, 其中的 messageSequenceNumber+1;
 - 3) 通知底层有节点退网、本节点成为主控节点;
 - 2)否则, 等待新主控节点广播的 SystemInformationBlockType2。
- 对于处在RRC_CONNECTED状态的节点, 当邻节点个数为0时, 或者主控节点释放本节点占用的固定资源, 则判定本节点退网, 操作流程如下:

- 1)删除本地配置;
- 1)执行网络搜索流程。

6.2.4.3 信令发送

RRC 要发送信令时, 需要向网络层获取路由信息, 发送给 PDCP 的信令结构如**错误!未找到引用源。**所示: 在信令前增加 16 个字节头: 目的节点的 MAC 地址 (6 字节)+10 字节填充, 用于实现信令的路由。10 字节填充默认写为 0xFF。表 1 是 RRC 发送信令的结构情况。

表 1 RRC 发送信令结构

6 字节	10 字节	变长
目的节点 MAC 地址	填充	编码后的信令

6.2.5 无线资源分配

6.2.5.1 资源分类

无线资源分为三类: 固定资源、动态资源、空闲资源。无线资源以无线帧为单位, 使用其在复帧中对应的序号进行标识。

- 1)固定资源: 节点入网时, 主控节点给新节点分配一个无线资源, 节点一直占用该资源直到退网。固定资源也作为节点的标识。
- 1)动态资源: 在节点入网后, 还可以动态的占用一些空闲的无线资源, 这些无线资源可以被随时占用或者释放。
- 1)空闲资源: 即没有被固定占用, 又没有被动态占用, 就被叫做空闲资源。每个节点只能在自己占用的固定资源和动态资源上进行业务数据的发送。

6.2.5.2 资源分配

资源分配主要包含固定资源的分配与动态资源的分配:

- 1)固定资源是在节点入网时, 由主控节点分配;

1)动态资源在节点入网后占用，每个节点只为自己占用动态资源，占用的准则如下：

- 2)根据本节点的 dataArrivalRate 在两跳内节点的总 dataArrivalRate 占比进行分配；
- 2)按照计算出的个数，对两跳内节点的空闲资源进行随机占用；
- 2)资源冲突释放资源后，对两跳内节点的空闲资源进行随机占用。

当节点计算出需要占用动态资源的个数比已占用的增加时，则需要再多占用相应数目的动态资源。当自己占用的动态资源改变时：

1)根据无线资源的占用情况，构造SystemInformationBlockType2并广播。

如果本节点或者邻节点的dataArrivalRate发生变化：

1)构造SystemInformationBlockType2并广播。

SystemInformationBlockType2 中包含发送节点及其一跳邻节点的动态资源占用情况，当节点收到SystemInformationBlockType2 后，需要更新无线资源的占用情况，如果无线资源变化：

1)构造SystemInformationBlockType2并广播。

6.2.5.3 资源释放

1)固定资源是在主控判定节点退网时删除。当节点不再占用固定资源时，释放自己占用的所有资源，执行网络搜索流程。

1)在如下情况下，进行动态资源的释放：

- 2)是根据动态资源分配的准则，当节点预期占用的动态资源个数比已占用的减少时，则需要释放相应数量的动态资源，每个节点动态资源的释放由本节点自行完成。当自己占用的动态资源改变时：
 - 3)根据无线资源的占用情况，构造 SystemInformationBlockType2 并广播。
- 2)节点判定自己脱网时，需要释放自己占用的所有资源，执行网络搜索流程。
- 2)节点判定本节点的动态资源跟邻节点有冲突时，则本节点释放有冲突的无线资源。

6.2.6 邻节点检测

节点在入网后开启周围的一跳邻节点的识别，当检测到邻节点后，节点通过接收邻节点的SystemInformationBlockType2 维护与邻节点关系。

6.2.6.1 邻节点发现

节点入网后，底层通过一跳邻节点识别策略发现新的邻节点。如果节点收到SystemInformationBlockType2，其中的 currentNodeId 指示不是已知的邻节点时，RRC 认为发现新的邻节点：

- 1)对相应节点进行SRBs和DRBs配置；
- 1)通知底层邻节点变化；
- 1)通知网络层发现新的邻节点；
- 1)构造SystemInformationBlockType2并广播。

6.2.6.2 邻节点删除

节点已经入网后，在以下情况下，节点需要删除与某个节点的邻节点关系：

- 1)一段时间没有收到该节点的SystemInformationBlockType2；
- 1)或者，与该节点之间发生无线链路失败（从RLC接收到已经达到重传最大次数的指示）；

- 1)或者，本节点作为被接入节点时，新接入节点入网失败；
- 当删除邻节点时，需要进行如下操作：
 - 1)删除相应节点的SRBs和DRBs配置；
 - 1)更新资源占用情况并通知底层；
 - 1)通知网络层发现新的邻节点；
 - 1)动态资源、与该节点的Metric变化，构造SystemInformationBlockType2并广播。

6.2.7 参数变更

6.2.7.1 参数广播

非主控节点变更参数的过程如图 6所示，主控节点变更参数的过程如图 7所示。

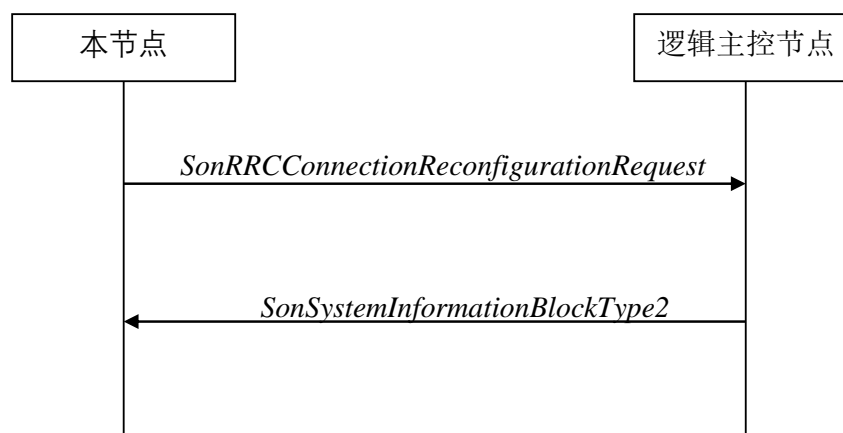


图 6 非主控节点发起参数变更的流程

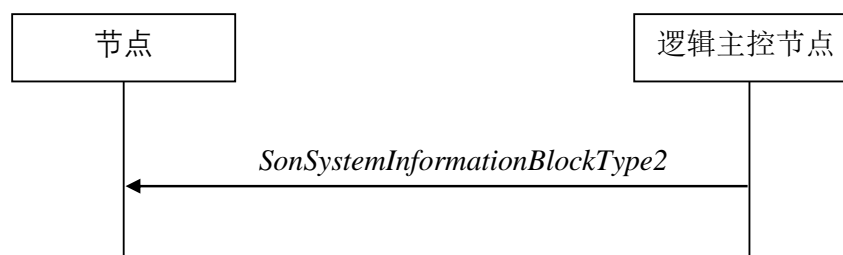


图 7 主控节点发起参数变更的流程

该过程旨在变更网络的全局参数，包括子网的频点，子网的带宽等参数。

已入网的任一节点，当发起子网频点或者带宽等参数的变更时，都可以发起该过程。未入网的节点，则不支持全网的子网参数变更。

若是在非主控节点启动该过程，应：

- 1)设置SonRRCConnectionReconfigurationRequest消息的内容，通过路由模块维护的路由表获取到逻辑主控节点的路由，将SonRRCConnectionReconfigurationRequest消息在DCCH上传输给下一跳节点。再经中继节点的转发，最终传输到逻辑主控节点；

- 1)启动定时器 $T_{ParaChg}$ ；

节点应如下设置SonRRCConnectionReconfigurationRequest消息的内容：

1)如果变更的是子网的频点，设置ParameterChangeInfo中的freqInfo，将其设置为待变更的频点；

1)如果变更的是子网的带宽，设置ParameterChangeInfo中的信息元素bandwidthInfo，将其设置为待变更的带宽；

1)如果变更的是子网的PCI，设置ParameterChangeInfo中的信息元素physCellId，将其设置为待变更的physCellId；

如果是在主控节点上发起参数变更、或者逻辑主控节点在接收到非主控节点发送的SonRRCConnectionReconfigurationRequest消息时，应：

1)根据接收到的ParameterChangeInfo进行全局参数配置过程；

1)根据如下设置SonSystemInformationBlockType2消息内容：

2)将 ParameterChangeInfo 中的 freqInfo 或 bandwidthInfo 设置为用户设置的参数、或将 physCellId 设置为节点选出的值；

2)将 ParameterChangeInfo 中的 activeTime，设置为全网统一生效的时间；

2)将 SonSystemInformationBlockType2 消息提供给底层进行传输；

1)当到达activeTime时，生效新的参数，设置SonSystemInformationBlockType2将不再携带信息元素 activeTime，并广播SonSystemInformationBlockType2；

6.2.7.2 参数生效

节点接收到携带新参数的SonSystemInformationBlockType2消息时，应：

1)如果是本节点触发的该参数变更，停止定时器TParaChg；

1)根据接收到的ParameterChangeInfo进行全局参数配置过程；

1)广播该SonSystemInformationBlockType2；

1)根据接收到的activeTime，生效新的参数。当新参数生效后，广播的 SonSystemInformationBlockType2中将不再携带信息元素activeTime。

6.2.8 子网分裂与合并

6.2.8.1 子网分裂

根据Sec5.2.4.2节点退网判断准则，当非主控节点升级为主控节点时，认为子网发生分裂。为了区分不同子网，新升级的主控节点选择一个新的PCI，根据Sec5.2.7中主控节点参数变更的描述变更全网的PCI。

6.2.8.2 子网检测

子网检测也就是测量发现其他子网的过程。测量包含两种类型：同频测试、异频测量。对于测量发现，节点在入网后应：

1)开启定时器TSubnetStable；该定时器未超时前收到SonSystemInformationBlockType2判断有其他节点接入或退出，重启定时器；

1)定时器TSubnetStable超时，开启子网检测过程，测量的对象是频点列表，该表可以是用户预配置；

6.2.8.3 子网合并

当节点测量检测到其他子网时，就发起子网合并过程，节点应：

1) 停止测量;

1) 开启定时器 T_{BgRead} , 尝试背景读取其他子网的信息, 信息包括但不限于 MIB;

2) 如果在定时器 T_{BgRead} 超时之前读到 MIB, 那么开始子网融合的判断:

3) 检查邻子网 MIB 中的 son_id (默认为 0) 与本子网是否相同:

4) 如果不同, 认为不是属于同一张网络, 停止融合过程;

4) 如果相同, 认为是同一张网络, 继续执行判断: 邻子网 MIB 中的 $totalNodeNum$ 与本子网的在网节点数的大小:

5) 如果本子网和邻子网的 $totalNodeNum$ 相等:

6) 比较邻子网和本子网的 $physical\ cell\ identity$ 大小, 按照小的 $physical\ cell\ identity$ 所在的子网往大的 $physical\ cell\ identity$ 所在的子网融合;

6) 如果两个子网的 $physical\ cell\ identity$ 相同, 那么比较两个子网的 $frequency$ 的大小, 按照小的 $frequency$ 所在的子网往大的 $frequency$ 所在的子网的规则进行融合

5) 如果本子网的 $totalNodeNum$ 大, 那么停止融合过程, 重新发起子网检测过程;

5) 如果本子网的 $totalNodeNum$ 小, 那么本子网发起融合到邻子网的过程;

2) 如果定时器 T_{BgRead} 超时, 那么停止融合过程;

1) 若被检测到的其他子网节点没有读取成功或者判定为不融合, 则节点重新开始子网检测。

根据上述子网融合规则, 当检测到其他子网的节点判定应发起到邻子网的融合时, 该节点应:

1) 设置 $SonSystemInformationBlockType2$ 的信息元素 $SubnetInfo$ 下属信息元素 $ActiveTime$, 将 $ActiveTime$ 设置为全网统一离网的时间; 设置信息元素 $SubnetInfo$, 携带目标子网的信息;

1) 广播该 $SonSystemInformationBlockType2$;

1) 当到达 $ActiveTime$ 时, 发起 Sec5.2.4.2 描述的主动退网流程;

1) 从当前子网退网后, 尝试接入目标子网。如果接入成功, 则认为该节点合并成功; 否则认为子网合并流程失败;

当检测到其他子网的节点的邻节点, 收到携带新子网信息的 $SonSystemInformationBlockType2$ 时:

1) 存储 $SubnetInfo$ 中携带的目标子网信息, 以及 $ActiveTime$;

1) 设置 $SonSystemInformationBlockType2$ 中的信息元素 $SubnetInfo$ 为目标子网、 $currentNodeLeaveIndication$ 为 TRUE, 并广播 $SonSystemInformationBlockType2$;

1) 当到达 $ActiveTime$ 时, 发起 Sec5.2.4.2 描述的主动退网流程;

从当前子网退网后, 尝试接入目标子网。如果接入成功, 则认为该节点合并成功; 否则认为接入失败, 尝试接入其他网络。

6.3 协议数据单元, 格式以及参数

6.3.1 概述

每一条 RRC 消息的内容在 5.3.2 节中有具体的描述, 使用 ASN.1 来描述该消息, 并且使用表格提供关于该信息元素进一步详细的信息。作为独立抽象类型的信息粒子句法, 在 5.3.3 节类似地有进一步描述。

在一条消息或抽象类型中, 显示信息元素的作用, 即在抽象概念 (ASN.1) 中把该 ASN.1 域描述成 OPTIONAL, 其实是借助抽象句法中附带的 OPTIONAL 描述的注释文本标签来进行阐述。所有的注释

文本标签仅仅用于下行方向。在表 2 中描述每个标签的意义。

表 2 注释文本标签说明

缩略词	意思
Cond conditionTag (仅仅使用在下行方向)	条件 表示一个用途视具体情况而定的信息元素。对于每一个 conditionTag, 具体的需求在下面 ASN.1 部分后面的表格中阐述。根据此条件, 如果一个域没有显示, 那么 UE 将不采取任何行动, 其中其余存在的值 (和/或相关的函数功能) 都可以继续使用, 除非有其他情况关于该域的明确说明。
Need OP (仅仅使用在下行方向)	可选 表示一个可以选择发送的信息元素。对于下行消息, 当过程文本或 ASN.1 部分后面的列表中没有关于该信息粒子的描述, 那么将不要求 UE 采取任何具体的行动。同时在此情况下, 过程文本或 ASN.1 部分后面的列表应该捕获该 UE 的行为。
Need ON (仅仅使用在下行方向)	可选, 没有动作 表示一个可以选择发送的信息粒子。如果 UE 接收了该消息, 并且缺省该信息元素, 那么 UE 将不采取任何行动, 继续沿用存在的值 (和/或者相关的功能)。
Need OR (仅仅使用在下行方向)	可选, 释放 表示一个可以选择发送的信息粒子。如果 UE 接收了该消息, 并且缺省该信息元素, 那么 UE 将终止/停止使用/删除任何存在的值 (和/或相关功能)。

在系统信息中任何具有 Need ON 的资源粒子将被看成为 Need OR。

对于下行包含一个或者多个扩展的组, 不应该指定Need域。一旦缺少该域, UE应

——对每个扩展, 包括在可选组中必须包含的扩展, 根据扩展所定义的 Need 域执行;

——不仅对直接包含在可选域中的扩展, 对于嵌套于将来应用的扩展都应用该操作。

注: 上述应用于使用双括号的非关键性扩展组, 也应用于一条消息后面的非关键性扩展, 或者一个 BIT STRING or OCTET STRING 中所包含的结构最后。

6.3.2 RRC 信息

6.3.2.1 通用消息结构

- SON-RRC-Definitions

该 ASN.1 部分是以 SON RRC PDU 定义为开头。

```
-- ASN1START
SON-RRC-Definitions DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
-- ASN1STOP
```

- SON-BCCH-BCH-Message

SON-BCCH-BCH-Message 为 RRC 消息的集合，其可能通过 BCCH 逻辑信道上的 BCH，在节点之间传送该消息。

```
-- ASN1START

SON-BCCH-BCH-Message ::= SEQUENCE {
    message SON-BCCH-BCH-MessageType
}

SON-BCCH-BCH-MessageType ::= MasterInformationBlock

-- ASN1STOP
```

- SON-BCCH-TCH-Message

SON-BCCH-TCH-Message 为 RRC 消息的集合，其可能通过 BCCH 逻辑信道上的 TCH，在节点之间传送该消息。

```
-- ASN1START

SON-BCCH-TCH-Message ::= SEQUENCE {
    message SON-BCCH-TCH-MessageType
}

SON-BCCH-TCH-MessageType ::= CHOICE {
    c1 CHOICE {
        sonsystemInformationBlockType1          SonSystemInformationBlockType1,
        sonsystemInformationBlockType2          SonSystemInformationBlockType2,
        spare5          NULL,
        spare4          NULL,
        spare3          NULL,
        spare2          NULL,
        spare1          NULL
    },
    messageClassExtension          SEQUENCE {}
}

-- ASN1STOP
```

- SON-DCCH-TCH-Message

SON-DCCH-TCH-Message 为 RRC 消息的集合，其可能通过 DCCH 逻辑信道上的 TCH，在节点之间传送该信息。

```
-- ASN1START

SON-DCCH-TCH-Message ::= SEQUENCE {
    message SON-DCCH-TCH-MessageType
}

SON-DCCH-TCH-MessageType ::= CHOICE {
    c1 CHOICE {
        sonRrcConnectionRequest          SonRRCConnectionRequest,
        sonRrcConnectionReject           SonRRCConnectionReject,
        sonRrcConnectionConfirm          SonRRCConnectionConfirm,
        spare5                NULL,
        spare4                NULL,
        spare3                NULL,
        spare2                NULL,
        spare1                NULL
    },
    messageClassExtension SEQUENCE {}
}

-- ASN1STOP
```

注：本节所包含的消息反映了当前的状态。

6.3.2.2 消息定义

– SonMasterInformationBlock

SonMasterInformationBlock 包含 BCH 上传输的系统信息。

信令无线承载： N/A

RLC-SAP： N/A

逻辑信道： BCCH

方向： 节点之间

SonMasterInformationBlock 消息

```
-- ASN1START

SonMasterInformationBlock ::= SEQUENCE {
    bandwidth          ENUMERATED {n6, n15, n25, n50, n75, n100},
    systemFrameNumber BIT STRING (SIZE (10)),
    transmissionMode  ENUMERATED {tm0, tm1, tm2, tm3},
    version            BIT STRING (SIZE (3)),
    currntNodeId      INTEGER (0..31),
}
```



```

totalNodeNum INTEGER (1..32),
antennaPortNum ENUMERATED {ap1, ap2, ap4},
txPower INTEGER (-40..50),
macI BIT STRING (SIZE (10)),
sonId INTEGER (0..7),
nonCriticalExtension SEQUENCE {} OPTIONAL
}
-- ASN1STOP

```

SonMasterInformationBlock 域描述如表 3 所示：

表 3 SonMasterInformationBlock 域描述

Bandwidth	参数：传输带宽配置。其中 n6 对应 6 个资源块， n15 对应 15 个资源块。
-----------	--

– SonRRCConnectionConfirm

SonRRCConnectionConfirm 消息用于允许新接入节点接入网络。

无线信令承载：SRB1

RLC-SAP：AM

逻辑信道：DCCH

方向：从主控节点到新接入节点

SonRRCConnectionConfirm 消息

```

-- ASN1START

SonRRCConnectionConfirm ::= SEQUENCE {
    criticalExtensions CHOICE {
        c1 CHOICE {
            rrcConnectionConfirm          SonRRCConnectionConfirm-IEs,
            spare                          NULL
        },
        criticalExtensionsFuture SEQUENCE {}
    }
}

SonRRCConnectionConfirm-IEs ::= SEQUENCE {
    occupiedRF          INTEGER (0..31),
    accessingNodeAddress NodeAddressIdentity,
    accessedNodeAddress NodeAddressIdentity,
    nonCriticalExtension SEQUENCE {} OPTIONAL
}

-- ASN1STOP

```

SonRRCConnectionConfirm 域描述如表 4 所示：

表 4 SonRRCConnectionConfirm 域描述

occupiedRF	逻辑主控节点分配给新接入节点的固定 RF。
------------	-----------------------

– SonRRCConnectionReject

SonRRCConnectionReject 消息用于拒绝接入网络。

信令无线承载： SRB1

RLC-SAP: AM

逻辑信道： DCCH

方向：从主控节点到新接入节点

SonRRCConnectionReject 消息

```

-- ASN1START

SonRRCConnectionReject ::= SEQUENCE {
    criticalExtensions CHOICE {
        c1 CHOICE {
            rrcConnectionReject          SonRRCConnectionReject-IEs,
            spare                          NULL
        },
        criticalExtensionsFuture SEQUENCE {}
    }
}

SonRRCConnectionReject-IEs ::= SEQUENCE {
    accessingNodeAddress          NodeAddressIdentity,
    accessedNodeAddress           NodeAddressIdentity,
    rejectCause ENUMERATED {congestion, spare7, spare6, spare5, spare4, spare3, spare2, spare1},
    nonCriticalExtension          SEQUENCE {} OPTIONAL -- Need OP
}

-- ASN1STOP

```

SonRRCConnectionReject 域描述如表 5 所示：

表 5 SonRRCConnectionReject 域描述

accessingNodeAddress	发起 RRC 连接建立的新接入节点标识
accessedNodeAddress	被接入节点的标识
rejectCause	拒绝接入原因

– SonRRCConnectionRequest

SonRRCConnectionRequest 消息用于请求接入网络。

信令无线承载： SRB1

RLC-SAP: AM

逻辑信道： DCCH

方向： 从新接入节点到主控节点

SonRRCConnectionRequest 消息

```
-- ASN1START

SonRRCConnectionRequest ::= SEQUENCE {
    criticalExtensions CHOICE {
        sonrrcConnectionRequest          SonRRCConnectionRequest-IEs,
        criticalExtensionsFuture          SEQUENCE {}
    }
}

SonRRCConnectionRequest-IEs ::= SEQUENCE {
    pre-occupiedRF INTEGER (0..31),
    accessingNodeAddress          NodeAddressIdentity,
    accessedNodeAddress          NodeAddressIdentity,
    nonCriticalExtension          SEQUENCE {} OPTIONAL    -- Need OP
}

-- ASN1STOP
```

SonRRCConnectionRequest 域描述如表 6 所示：

表 6 SonRRCConnectionRequest 域描述

accessingNodeAddress	发起 RRC 连接建立的新接入节点标识
accessedNodeAddress	被接入节点的标识
pre-occupiedRF	新接入节点预占的 RF id

— SonSystemInformationBlockType1

SonSystemInformationBlockType1 包含与路由相关的信息。

信令无线承载： N/A

RLC-SAP: N/A

逻辑信道： BCCH

方向： 节点之间

SonSystemInformationBlockType1 消息

```
-- ASN1START

SonSystemInformationBlockType1 ::= SEQUENCE {
```

```

messageSequenceNumber      INTEGER (0..255),
messageSegmentType         ENUMERATED {notLastSegment, lastSegment},
messageSegmentNumber       INTEGER (0..63),
nodeRouteInformation       NodeRouteInformation,
nonCriticalExtension       SEQUENCE {} OPTIONAL    -- Need OP
}
-- ASN1STOP

```

SystemInformationBlockType1 域描述如表 7 所示:

表 7 SystemInformationBlockType1 域描述

messageSequenceNumber	SIB1 消息序列号，表示 SIB1 消息新旧
messageSegmentTypeSIB1	消息分段类型，是否为最后一段消息
messageSegmentNumber	SIB1 消息分段段数信息
nodeRouteInformation	包含路由通告信息

– SonSystemInformationBlockType2

SonSystemInformationBlockType2 包含与参数变更、资源分配等信息。

信令无线承载: N/A

RLC-SAP: N/A

逻辑信道: BCCH

方向: 节点之间

SonSystemInformationBlockType2 消息

```

-- ASN1START

SonSystemInformationBlockType2 ::= SEQUENCE {
    ssfnCycle      INTEGER (0..65535),
    currentNodeId  INTEGER (0..31),
    messageSequenceNumber      INTEGER (0..255),
    currentNodeLeaveIndication  ENUMERATED {false,true},
    subnetInfo      SubnetInfo,
    currentNodeAddress      NodeAddressIdentity,
    mainNodeAddress NodeAddressIdentity,
    occupiedRadioFrameList  OccupiedRadioFrameList OPTIONAL,
    nonCriticalExtension    SEQUENCE {} OPTIONAL
}

-- ASN1STOP

```

SystemInformationBlockType2 域描述如表 8 所示:

表 8 SystemInformationBlockType2 域描述

currentNodeId	发送 SystemInformationBlockType2 节点的节点 id
currentNodeLeaveIndication	节点是否离网的指示
son-Identity	当前子网的网络标识
occupiedRadioFrameList	用于指示每个无线帧当前被节点固定和动态占用的情况，当前版本必须携带本字段。
subnetInfo	当前子网的信息，由子网频点和 PCI 组成
messageSequenceNumber	当网络中有节点入网或退网情况，主控节点都会更新 messageSequenceNumber，入网后网络稳定时，messageSequenceNumber 不进行更新。非主控节点的 messageSequenceNumber 需与主控节点保持一致。

6.3.3 RRC 信息元素

6.3.3.1 无线资源信息元素

– OccupiedRadioFrameList

IE OccupiedRadioFrameList 用于描述每个无线帧被节点固定和动态占用的情况。

OccupiedRadioFrameList 信息元素

```
-- ASN1START
OccupiedRadioFrameList ::= SEQUENCE (SIZE(maxTotalNodeNum)) OF OccupiedPerRadioFrame

OccupiedPerRadioFrame ::= SEQUENCE{
    fixedOccupiedPerRadioFrame          FixedOccupiedPerRadioFrame    OPTIONAL,
    dynamicOccupiedPerRadioFrame        DynamicOccupiedPerRadioFrame  OPTIONAL,
    ...
}
-- ASN1STOP
```

OccupiedRadioFrameList域描述如表9所示：

表 9 OccupiedRadioFrameList 域描述

fixedOccupiedPerRadioFrame	用于指示每个无线帧被节点固定占用的情况
dynamicOccupiedPerRadioFrame	用于指示每个无线帧被节点动态占用的情况。

– FixedOccupiedPerRadioFrame

IE FixedOccupiedPerRadioFrame 用于描述每个无线帧被节点固定占用的情况。

FixedOccupiedPerRadioFrame 信息元素

```

-- ASN1START

FixedOccupiedPerRadioFrame ::= SEQUENCE{
    Metric    INTEGER (0..255),
    Parameter    Parameter    OPTIONAL, -- Cond CurNode or One Hop Node
    nodeSN     OCTET STRING (SIZE(4)),
    ...
}

Parameter ::= SEQUENCE{
    dataArrivalRate INTEGER (0..65535),
    ...
}

-- ASN1STOP

```

FixedOccupiedPerRadioFrame 域描述如表10所示:

表 10 FixedOccupiedPerRadioFrame 域描述

metric	本节点与固定占用该 RF 的节点之间的跳数
nodeSN	节点标识，共 4 个字节（取值范围是 0~0xFFFFFFFF），其中最左侧的一个字节用于表示不同厂商，需要保证网络中的节点标识不能重复
parameter	固定占用该 RF 的节点的当前速率信息

– DynamicOccupiedPerRadioFrame

IE DynamicOccupiedPerRadioFrame 用于描述每个无线帧被节点动态占用的情况。

DynamicOccupiedPerRadioFrame 信息元素

```

-- ASN1START

DynamicOccupiedPerRadioFrame ::= SEQUENCE{
    nodeId     INTEGER (0..31),
    conflictIndication    ENUMERATED {false,true},
    ...
}

-- ASN1STOP

```

DynamicOccupiedPerRadioFrame域描述如表11所示:

表 11 DynamicOccupiedPerRadioFrame 域描述

nodeId	动态占用该无线帧的节点 id
conflictIndication	是否因与其他节点同时动态占用该无线帧而发生资源冲突。

6.3.3.2 路由信息元素

– NodeRouteInformation

IE NodeRouteInformation 用于描述路由相关信息。

NodeRouteInformation 信息元素

```
-- ASN1START
NodeRouteInformation ::= SEQUENCE {
    nodeRouteInfo  OCTET STRING
}
-- ASN1STOP
```

NodeRouteInformation域描述如表12所示：

表 12 NodeRouteInformation 域描述

nodeRouteInfo	每段路由通告的字节流。 注：为了判断路由通告来自哪一个邻节点，需要在每段路由通告的字节流末端携带本节点的 NODE_ID
---------------	---

6.3.3.3 其他信息元素

– ActiveTime

IE ActiveTime 用于描述激活时间。

ActiveTime 信息元素

```
-- ASN1START
ActiveTime ::= SEQUENCE{
    ssfnCycle  INTEGER (0..65535),
    ssfn       INTEGER (0..10239)
}
-- ASN1STOP
```

– NodeAddressIdentity

IE NodeAddressIdentity 用于描述节点标识。

NodeAddressIdentity 信息元素

```

-- ASN1START

NodeAddressIdentity ::= SEQUENCE{
    nodeSN OCTET STRING (SIZE(4)),
    ...
}

-- ASN1STOP

```

– SubnetInfo

IE SubnetInfo 用于描述子网信息。

SubnetInfo 信息元素

```

-- ASN1START

SubnetInfo ::= SEQUENCE{
    physCellId INTEGER (0..503),
    subnetFreq INTEGER (0..65535),
    leaveTime ActiveTime OPTIONAL,
    ...
}

-- ASN1STOP

```

SubnetInfo域描述如表13所示：

表 13 SubnetInfo 域描述

subnetFreq	子网所在的频点信息，绝对频点号 = 频率*10
leaveTime	当检测到其他子网时携带，表示小子网的离网时间

6.3.4 RRC 多样性和类型常量值

– 多样性和类型约束定义

```

-- ASN1START

maxTotalNodeNum INTEGER ::= 32 -- Maximum node sum for SON.

-- ASN1STOP

```


– SON-RRC-Definitions 末尾

```
-- ASN1START
END
-- ASN1STOP
```

6.4 定时器

定时器的具体情况如表14所示。

表 14 定时器

定时器	开始	停止	时长 (ms)	超时
T _{Msg2}	随机接入发送 msg1	收到 msg2	960	重新发起接入
T _{RrcReq}	被接入节点发送了 msg2 后	收到新接入节点的 SonRRCCo ⁿ nectionRequest	640	认为新接入节点接入失败
T _{RrcRsp}	传输 SonRRCCo ⁿ nectionRequest-v3	收到 SonRRCCo ⁿ nectionConfirm-v3 后	10240	重新发起接入
T _{ParaChg}	传输 SonRRCCo ⁿ nectionReconfigurationRequest-v3	收到对应允许参数变更的 SonSystemInformationBlockType2-v3 后	10240	认为参数变更失败
T _{SubnetStable}	节点入网成功后, 或者网络中有其他节点接入或者退出时	网络中有其他节点接入或者退出时	60000	发起邻子网检测
T _{BgRead}	底层检测到新子网后	读到邻子网的 MIB	650	认为子网融合失败

6.5 默认无线配置

6.5.1 SRB 配置

6.5.1.1 SRB1

SRB1的默认配置如表15所示。

表 15 SRB1 默认配置

Name	Value	Semantics description	Ver
RLC 配置选择	am		
Priority	1	最高优先级	
logicalChannelGroup	0		
LCID	1		

6.5.1.2 SRB2

SRB2的默认配置如表16所示。

表 16 SRB2 默认配置

Name	Value	Semantics description	Ver
RLC 配置选择	um		
Send-RLC-Config >Sn-FieldLength	size5		
recvive-RLC-Config >Sn-FieldLength	size5		
逻辑信道配置			
priority	2		
logicalChannelGroup	1		
LCID	2		

6.5.2 DRB 配置

6.5.2.1 DRB3

SRB3的默认配置如表17所示。

表 17 DRB3 默认配置

Name	Value	Semantics description	Ver
PDCP 配置选择			
pdcp-SN-Size	Size12		
RLC 配置选择	um		
send-RLC-Config >Sn-FieldLength	Size10		
recvive -RLC-Config >Sn-FieldLength	Size10		
逻辑信道配置			
priority	3		
logicalChannelGroup	1		
LCID	3		

6.5.2.2 DRB4

SRB4的默认配置如表18所示。

表 18 DRB4 默认配置

Name	Value	Semantics description	Ver
RLC 配置选择	am		

Name	Value	Semantics description	Ver
逻辑信道配置			
priority	5		
logicalChannelGroup	3		
LCID	4		

6.5.2.3 DRB5

SRB5的默认配置如表19所示。

表 19 DRB5 默认配置

Name	Value	Semantics description	Ver
PDCP 配置选择			
pdcp-SN-Size	Size12		
RLC 配置选择	um		
send-RLC-Config >Sn-FieldLength	Size10		
recvie -RLC-Config >Sn-FieldLength	Size10		
逻辑信道配置			
priority	4		
logicalChannelGroup	2		
LCID	5		

6.5.2.4 DRB7

SRB7的默认配置如表20所示。

表 20 DRB7 默认配置

Name	Value	Semantics description	Ver
PDCP 配置选择			
pdcp-SN-Size	Size12		
RLC 配置选择	um		
send-RLC-Config >Sn-FieldLength	size5		
recvie-RLC-Config >Sn-FieldLength	size5		
逻辑信道配置			
priority	6		
logicalChannelGroup	4		
LCID	7		

7 PDCP

7.1 概述

7.1.1 PDCP 架构

图 8 描述了 PDCP 子层一种可能的架构；图 8 不限制 PDCP 的实现。

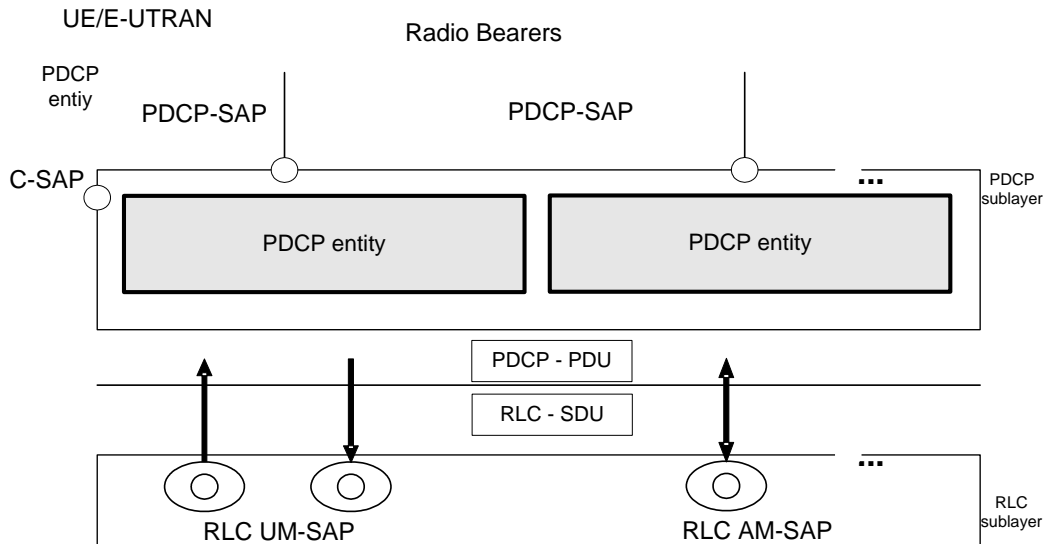


图 8 PDCP 层结构视图

每个 RB（即，DRB 和 SRB，除去 SRB0）关联于一个 PDCP 实体。根据 RB 特性（即：单向或双向）及 RLC 模式，每个 PDCP 实体关联于一个或两个（每个方向一个）RLC 实体。PDCP 实体位于 PDCP 子层。

PDCP 子层由 RRC 层配置，RRC 层在 RRC 规范部分进行规定。

7.1.2 PDCP 实体

PDCP 实体位于 PDCP 子层。可以为一个自组网节点定义多个 PDCP 实体。

每个 PDCP 实体携带一个无线承载的数据。

根据无线承载所携带的数据，PDCP 实体对应于控制平面或者用户平面。

图 9 描述了 PDCP 子层中 PDCP 实体的功能视图，不限制实现。

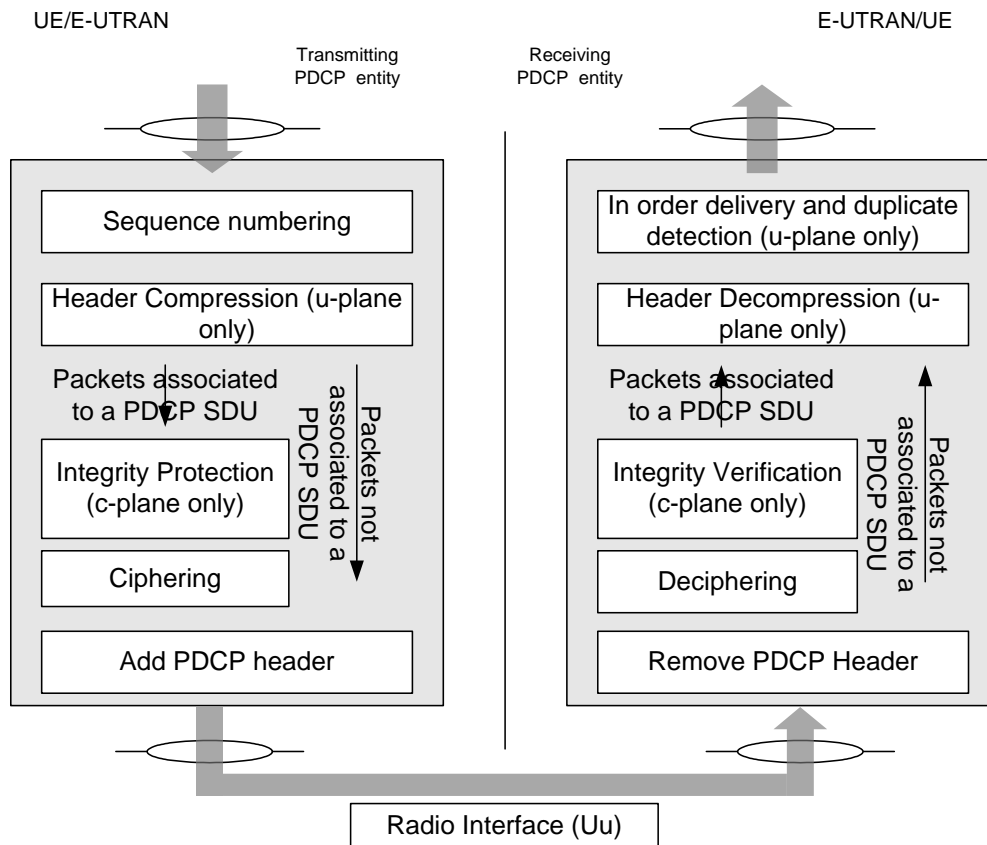


图 9 PDCP 层功能视图

7.1.3 业务

7.1.3.1 提供给上层的服务

PDCP 向 RRC 层以及上层业务转发模块提供服务。PDCP 向上层提供如下服务：

- 用户平面数据的传输；
- 控制平面数据的传输；
- 加密；
- 完整性保护。

支持的最大 PDCP SDU 为 8188 个字节。

7.1.3.2 从下层获得的服务

PDCP 从下层 RLC 获取如下服务。

- 确认数据传输服务，包括 PDCP PDU 成功传输的标识；
- 非确认数据传输服务；
- 按序传输，除了下层的重建时；
- 重复丢弃，除了下层的重建时。

7.1.4 功能

分组数据汇聚协议（PDCP）支持以下功能：

- 数据传输（用户平面或控制平面）；

- 对 PDCP SN 值的维护；
- 在下层重建的时候，按序传递上层 PDU；
- 在下层重建的时候，为映射到 RLC AM 的无线承载重复丢弃下层 SDU；
- 对用户平面数据及控制平面数据的加密及解密；
- 控制平面数据的完整性保护及完整性验证；
- 定时丢弃；
- 重复丢弃。

PDCP 使用 RLC 子层提供的服务。

PDCP 用于映射到 DCCH 及 DTCH 类型逻辑信道的 SRB 及 DRB。PDCP 不能用于其它类型的逻辑信道。

7.1.5 可传数据

为了上报 MAC 的缓存状态，UE 应认为 PDCP 控制 PDU 及以下所述数据作为 PDCP 层中的可传数据：

对于没有向底层下层传递 PDU 的 SDU：

- SDU 本身，如果 SDU 还没有被 PDCP 处理，或
- PDU，如果 SDU 已被 PDCP 处理。

7.2 PDCP 过程

7.2.1 PDCP 数据传输过程

7.2.1.1 发送数据传输过程

从上层接收到 PDCP SDU 以后，发送节点应：

1)启动与此PDCP SDU相关联的discardTimer（如果已配置）；

对于从上层接收到的 PDCP SDU，发送节点应：

1)关联相应于Next_PDCP_TX_SN 的PDCP SN到PDCP SDU；

1)按照5.5.4节的说明执行PDCP SDU的头压缩（如果已配置）

1)执行完整性保护（如适用），并使用基于TX_HFN的COUNT 以及关联于PDCP SDU的PDCP SN 值进行加密（如适用），分别见完保小节和加解密小节的描述；

1)将Next_PDCP_TX_SN 增加1；

1)如果Next_PDCP_TX_SN > Maximum_PDCP_SN：

2)将 Next_PDCP_TX_SN 置为 0；

2)将 TX_HFN 固定为 1；

1)将最后产生的PDCP Data PDU 传送给下层。

7.2.1.2 接收数据传输过程

7.2.1.2.1 DRB 过程

1、空闲

2、映射到 RLC AM 的 DRB 过程

对于映射到 RLC AM 的 DRB，在接收到下层的 PDCP Data PDU 时，接收节点应：

- 1)如果接收到的PDCP SN - Last_Submitted_PDCP_RX_SN > Reordering_Window 或者 $0 \leq \text{Last_Submitted_PDCP_RX_SN} - \text{接收到的 PDCP SN} < \text{Reordering_Window}$:
 - 2)如果接收到的 PDCP SN > Next_PDCP_RX_SN:
 - 3)使用基于 RX_HFN - 1 的 COUNT 值与接收到的 PDCP SN 值, 解密此 PDCP PDU, 如加解密小节所述;
 - 2)否则:
 - 3)使用基于 RX_HFN 的 COUNT 值与接收到的 PDCP SN 值, 解密此 PDCP PDU, 如加解密小节所述;
 - 2)丢弃此 PDCP SDU;
- 1)否则如果Next_PDCP_RX_SN - 接收的PDCP SN > Reordering_Window:
 - 2)将 RX_HFN 固定为 1;
 - 2)使用基于 RX_HFN 的 COUNT 值与接收到的 PDCP SN 值解密此 PDCP PDU;
 - 2)将 Next_PDCP_RX_SN 置为接收到的 PDCP SN + 1;
- 1)否则如果接受的 PDCP SN - Next_PDCP_RX_SN >= Reordering_Window:
 - 2)使用基于 RX_HFN - 1 的 COUNT 值与接收到的 PDCP SN 值解密此 PDCP PDU;
- 1)否则如果接收到的PDCP SN >= Next_PDCP_RX_SN:
 - 2)使用基于 RX_HFN 的 CONUT 值与接收到的 PDCP SN 值解密此 PDCP PDU;
 - 2)将 Next_PDCP_RX_SN 置为接收到的 PDCP SN + 1;
 - 2)如果 Next_PDCP_RX_SN 大于 Maximum_PDCP_SN:
 - 3)将 Next_PDCP_RX_SN 置为 0;
 - 3)将 RX_HFN 固定为 1;
- 1)否则如果接收到的PDCP SN < Next_PDCP_RX_SN:
 - 2)使用基于 RX_HFN 的 COUNT 值与接收到的 PDCP SN 值解密此 PDCP PDU;
- 1)如果上面没有丢弃此PDCP PDU:
 - 2)执行 PDCP PDU 的解密;
 - 2)如果一个具有相同 PDCP SN 值的 PDCP PDU 被存储:
 - 3)丢弃此 PDCP SDU;
 - 2)否则:
 - 3)存储此 PDCP SDU;
 - 2)如果由于下层重建导致 PDCP 没有接收到此 PDCP PDU:
 - 3)按照相关 COUNT 值的升序排列, 传递给上层;
 - 2)所有存储的, 相关 COUNT 值小于接收 PDCP SDU 的 COUNT 值的 PDCP SDU;
 - 2)所有存储的, 从接收到的 PDCP SDU 的 COUNT 值开始, 连续 COUNT 值对应的 PDCP SDU;
 - 3)将 Last_Submitted_PDCP_RX_SN 置为最后递交给高层的 PDCP SDU 的 PDCP SN 值;
 - 2)否则如果接收到的 PDCP SN = Last_Submitted_PDCP_RX_SN + 1, 或者接收到的 PDCP SN = Last_Submitted_PDCP_RX_SN - Maximum_PDCP_SN:

3)把相关 COUNT 值按升序传递给上层;

2)所有存储的,从接收到的 PDCP SDU 的 COUNT 值开始,连续 COUNT 值对应的 PDCP SDU;

3)将 Last_Submitted_PDCP_RX_SN 置为最后递交给高层的 PDCP SDU 的 PDCP SN 值。

3、映射到 RLC UM 的 DRB 过程

对于映射到 RLC UM 的 DRB,在接收到下层的 PDCP Data PDU 以后,UE 应:

1)如果接收到的 PDCP SN < Next_PDCP_RX_SN:

2)将 RX_HFN 固定为 1;

1)使用基于 RX_HFN 的 COUNT 值与接收到的 PDCP SN 值,解密此 PDCP Data PDU;

1)将 Next_PDCP_RX_SN 置为接收到的 PDCP SN 值 + 1;

1)如果 Next_PDCP_RX_SN > Maximum_PDCP_SN:

2)将 Next_PDCP_RX_SN 置为 0;

2)将 RX_HFN 固定为 1;

1)将最后产生的 PDCP SDU 递交给上层。

7.2.1.2.2 SRB 过程

对于 SRB,在接收到下层的 PDCP Data PDU 以后,UE 应:

1)如果接收的 PDCP SN < Next_PDCP_RX_SN:

2)使用基于 RX_HFN + 1 的 COUNT 值与接收到的 PDCP SN 值来解密 PDU 及确认 PDU 的完整性(如适用);

1)否则:

2)使用基于 RX_HFN 的 COUNT 值与接收到的 PDCP SN 值来解密此 PDU 及确认其完整性(如适用);

1)完整性验证使用,并且成功通过;或者

1)如果完整性验证不适用:

2)如果接收的 PDCP SN < Next_PDCP_RX_SN:

3)将 RX_HFN 固定为 1;

2)将 Next_PDCP_RX_SN 置为接收到的 PDCP SN 值 + 1;

2)如果 Next_PDCP_RX_SN > Maximum_PDCP_SN:

3)将 Next_PDCP_RX_SN 置为 0;

3)将 RX_HFN 固定为 1;

2)将最后产生的 PDCP SDU 递交给上层;

1)否则,如果完整性验证适用,但是失败:

2)丢弃接收到的 PDCP Data PDU;

2)将完整性验证失败报告给上层。

7.2.2 PDCP 丢弃

当用于 PDCP SDU 的 discardTimer 终止,或 PDCP SDU 的成功传输被 PDCP 状态报告确认,节点应丢弃此 PDCP SDU 连同对应的 PDCP PDU。如果对应的 PDCP PDU 已经成功传递给下层,则此丢弃需要指示给下层。

7.2.3 加密和解密

可不实现加解密，使用空加解密算法。

加密功能包括加密和解密，在 PDCP 中执行。对于控制平面，加密的数据单元是 PDCP PDU 的数据部分及 MAC-I。对于用户平面，加密的数据单元是 PDCP PDU 的数据部分；加密不适用于 PDCP 控制 PDU。

加密算法和PDCP实体使用的密钥由上层RRC配置，上层可以从本地操作管理中获取密钥和具体加解密算法，生成控制平面密钥和用户平面密钥配置给PDCP。加解密算法如表21所示。

表 21 加解密算法表

序号	标志	加解密算法
0	EEA0	Null ciphering algorithm
1	Reserved	Reserved
2	Reserved	Reserved
3	128-EEA3	ZUC based algorithm

加密功能由上层激活。在安全激活之后，加密功能应分别用于所有上层指示的接收及发送 PDCP PDU。

PDCP 加密必需的参数如下，并用于加密算法。加密功能请求的输入包括 COUNT 值及 DIRECTION（传输的方向）。PDCP 请求的由上层提供的参数如下所列：

- BEARER (定义为无线承载标识符)；
 - KEY（加密 key、KRRCenc 与 KUPenc 分别用于控制平面和用户平面）。
- 加密和解密支持空算法和 ZUC 算法（祖冲之算法）。

7.2.4 完整性保护及确认

一期可不实现完整性保护及验证，使用空完整性保护及验证算法。

完整性保护功能包括完整性保护与完整性确认，并且在 PDCP 中执行用于与 SRB 关联的 PDCP 实体。受完整性保护的数据单元为 PDU 头和加密前 PDU 的数据部分。

完整性保护算法和PDCP实体使用的key由上层RRC配置，上层RRC可以从本地操作管理中获取密钥和具体完整性保护和验证算法，生成完整性保护密钥给PDCP。完整性保护和验证算法如表22所示。

表 22 完整性保护和验证算法表

序号	标志	完整性保护和验证算法
0	EIA0	Null Integrity Protection algorithm
1	Reserved	Reserved
2	Reserved	Reserved
3	128-EIA3	ZUC based algorithm

完整性保护功能由上层 RRC 激活。在安全激活以后，完整性保护功能应分别用于从上层指定的 PDU 之后（包括此 PDU）所有接收、发送的 PDU。

注：如果激活完整性保护功能的 RRC 消息自身被完整性保护着（使用其携带的配置信息），则此消

息需要在接收到此消息的 PDU 的完整性保护确认执行之前首先被 RRC 解码出来。

PDCP 请求用于完整性保护的参数如下，且作为完整性保护算法的输入。作为完整性保护算法的输入包括 COUNT 值和 DIRECTION 值（传输的方向）。PDCP 请求由上层提供的参数如下所列：

- BEARER (定义为无线承载指示符)；
- KEY (KRRCint)。

在发送时，发送节点计算 MAC-I 字段的值，在接收时，接收节点通过基于以上指定的输入参数计算 X-MAC 来确认 PDCP PDU 的完整性。如果计算得来的 X-MAC 值与接收的 MAC-I 值相对应，则完整性保护确认成功。

完整性保护和验证支持空算法和 ZUC 算法（祖冲之算法）。

7.2.5 未知的，意外的以及错误的协议数据的处理

当 PDCP 实体接收到一个包括保留值或非法值的 PDCP PDU 时，PDCP 实体应：

- 1) 1) 丢弃接收到的 PDU。

7.3 协议数据单元，格式及参数

7.3.1 协议数据单元

7.3.1.1 PDCP 数据 PDU

PDCP 数据 PDU 用来传输：

- PDCP SDU SN 值；和
- 包含一个未压缩 PDCP SDU 的用户平面数据；或者
- 包含一个压缩 PDCP SDU 的用户平面数据；或者
- 控制平面数据；或者
- 一个只用于 SRB 的 MAC-I 字段。

7.3.1.2 PDCP 控制 PDU

PDCP 控制 PDU 用来传输：

- 指示丢失的以及不是在 PDCP 重建之后的 PDCP SDU 的 PDCP 状态报告。

7.3.2 格式

7.3.2.1 综述

PDCP PDU 是一个长度按字节排列的比特串(即，8 位的整数倍)。在数据格式图中，比特串由表格来描述，其中最高位是表格第一行最左边的位，最低位是最后一行最右边的位，更简单的说，比特串是从左到右逐行读取的。PDCP PDU 中每个参数字段的比特顺序都是以最左位为最高位，以最右位为最低位。

PDCP SDU 也是长度按字节排列的比特串（即，8 位的整数倍）。从第一位开始，PDCP PDU 中包括一个压缩的 SDU 或未压缩的 SDU。

7.3.2.2 控制平面 PDCP Data PDU

图 10 所示为携带控制平面 SRB 数据的 PDCP Data PDU 的格式。

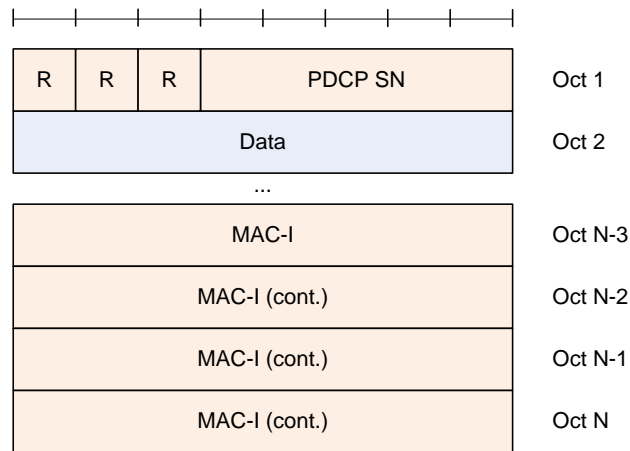


图 10 SRB 的 PDCP Data PDU 的格式

7.3.2.3 具有长 PDCP SN 值（12bit）的用户平面 PDCP Data PDU

图 11 所示为使用 12bit SN 值的 PDCP Data PDU 的格式。此格式适用于携带映射到 RLC AM 或 RLC UM 的 DRB 的数据的 PDCP Data PDU。

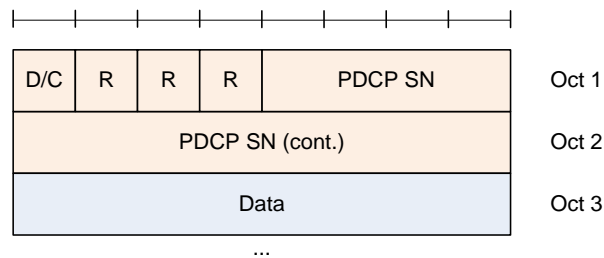


图 11 DRB 使用 12bit SN 值的 PDCP Data PDU 的格式

7.3.2.4 具有短 PDCP SN 值（7bit）的用户平面 PDCP Data PDU

图 12 所示为使用 7bit SN 值的 PDCP Data PDU 的格式。此格式适用于携带映射到 RLC UM 的 DRB 的数据的 PDCP Data PDU。

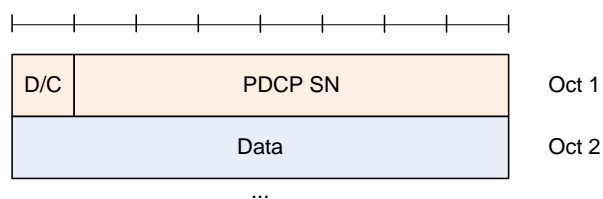


图 12 DRB 使用 7bit SN 值的 PDCP Data PDU 的格式

7.3.3 参数

7.3.3.1 综述

如果在每个字段的定义中没有其他说明，则参数中的 bit 应按照如下进行说明：最左位为第一位且是最高位，最右位为最后一位且是最低位。

除非另作说明，整数编码为无符号整数的标准二进制。就一切情况而论，PDU 中 bit 的读取顺序为 MSB 到 LSB。

7.3.3.2 PDCP SN

长度：5，7 或者 12 比特，如表 23 所示。

表 23 PDCP SN 长度

Length	Description
5	SRBs
7	DRBs, if configured by upper layers(pdcpc-SN-Size [2])
12	DRBs, if configured by upper layers(pdcpc-SN-Size [2])

7.3.3.3 数据

长度：可变。

数据字段可以包括以下两者之一：

- 未压缩 PDCP SDU (用户平面数据，或者控制平面数据)；或者
- 压缩 PDCP SDU (只是用户平面数据)。

7.3.3.4 MAC-I

长度：32 比特。

MAC-I 字段携带一个消息认证码，按照 5.7 节的指定进行计算得来的。

对于未经过完整性保护得控制平面数据，MAC-I 字段仍然存在，宜用 0 作为填充位填充。

7.3.3.5 COUNT

长度：32 比特。

加密及完整性保护维护一个COUNT值。COUNT值由HFN和PDCP SN组成。PDCP SN值的长度由上层配置。COUNT的格式如图 13所示。

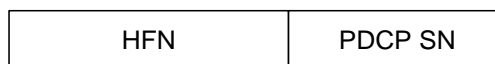


图 13 COUNT 的格式

HFN 部分的长度等于 32 减去 PDCP SN 的长度。

注：在进行 COUNT 相关值的比较时，节点将视 COUNT 为 32 比特的值，值为循环值（例如，值为 $2^{32}-1$ 的 COUNT 小于值为 0 的 COUNT）。

7.3.3.6 R

长度：1 比特。

保留值。在此规范的当前版本，保留位都置为 0。在接收机处忽略保留位。

7.3.3.7 D/C

长度：1 比特。如表 24 所示。

表 24 D/C 域

Bit	Description
0	Control PDU
1	Data PDU

7.3.3.8 PDU 类型

长度：3 比特。如表 25 所示。

表 25 PDU 类型

Bit	Description
000	PDCP status report
001	Interspersed ROHC feedback packet
010-111	reserved

7.4 变量，常量及定时器

7.4.1 状态变量

本节描述 PDCP 实体中使用的状态变量，以此来详细说明 PDCP 协议。

所有的状态变量都是非负的整数。

PDCP 实体的传输方应维护以下状态变量：

——Next_PDCP_TX_SN

变量 Next_PDCP_TX_SN 表示特定 PDCP 实体的下一个 PDCP SDU 的 PDCP SN 值。在 PDCP 实体置 Next_PDCP_TX_SN 初始值为 0。

——TX_HFN

变量 TX_HFN 表示 HFN 值，用以产生特定 PDCP 实体发送的 PDCP PDU 的 COUNT 值。在 PDCP 实体置 TX_HFN 初始值为 0。

PDCP 实体的接收方需要维护一下状态变量：

——Next_PDCP_RX_SN

变量 Next_PDCP_RX_SN 表示特定 PDCP 实体接收的下一个预期 PDCP SN。在 PDCP 实体置 Next_PDCP_RX_SN 初始值为 0。

——RX_HFN

变量 RX_HFN 表示 HFN 值,用以产生特定 PDCP 实体接收的 PDCP PDU 使用的 COUNT 值。在 PDCP 实体应置 RX_HFN 初始值为 0。

—— Last_Submitted_PDCP_RX_SN

对于映射到 RLC AM 上的 DRB 的 PDCP 实体, 变量 Last_Submitted_PDCP_RX_SN 表示最后一个传输到上层的 PDCP SDU 的 SN 值。在 PDCP 实体应置 Last_Submitted_PDCP_RX_SN 初始值为 4095。

7.4.2 定时器

DRB 每个 PDCP 实体的发送端应维护以下定时器:

——discardTimer

定时器的持续时间由上层配置[2]。在发送端, 一个新定时器在接收到上层的 SDU 后开启。

7.4.3 常量

——Reordering_Window

表示重排窗口的大小。对于映射到 RLC AM 的无线承载, 其大小等于 2048, 也即, PDCP SN 空间的一半。

——Maximum_PDCP_SN 是:

4095 如果 PDCP 实体配置为使用 12bit SN 值;

127 如果 PDCP 实体配置为使用 7bit SN 值;

31 如果 PDCP 实体配置为使用 5bit SN 值。

8 RLC

8.1 概述

8.1.1 RLC 结构

8.1.1.1 UM Mode RLC 实体

UM RLC 实体能被配置为通过如下逻辑信道发送或接收 RLC PDUs, 两个非确认模式对等实体的模型如图 14 所示:

——DTCH

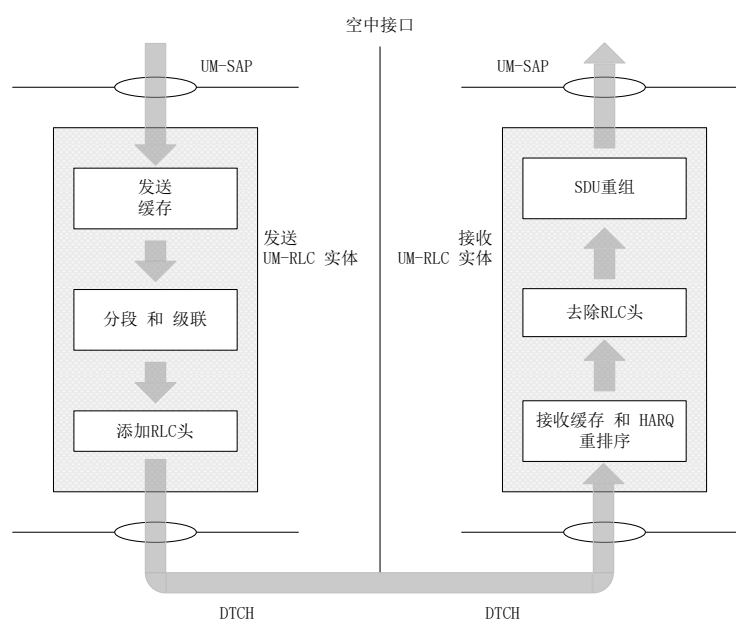


图 14 两个非确认模式对等实体的模型

UM RLC 实体发送或接收如下的 RLC 数据 PDU:

——UMD PDU。

8.1.1.2 发送 UM RLC 实体介绍

当发送 UM RLC 实体由 RLC SDUs 构造 UMD PDUs 时, 应:

- 对 RLC SDUs 进行分段和/或级联, 使得 UMD PDUs 的大小适合于下层 (MAC) 通知的特定传输机会所指示的 RLC PDU(s) 总大小;
- 在 UMD PDU 包含相关的 RLC 头。

8.1.1.3 接收 UM RLC 实体介绍

当接收 UM RLC 实体接收到 UMD PDUs 时, 应:

- 检测该 UMD PDUs 是否被重复接收, 并丢弃重复的 UMD PDUs;
- 若被乱序接收, 则对 UMD PDUs 进行重排序;
- 检测下层 UMD PDUs 的丢失, 并避免过多的重排序延迟;
- 从重排序后的 UMD PDUs 重组出 RLC SDUs (已检测出丢失的 RLC PDUs 不算在内), 并且按 RLC SN 的升序向上层分发 RLC SDUs;
- 因为属于一个特定 RLC SDU 的一个 UMD PDU 在下层丢失, 而导致不能被重组到该 RLC SDU 的接收到的 UMD PDUs, 则丢弃该 RLC SDU 中已经收到的 UMD PDUs。

8.1.1.4 AM Mode RLC 实体

8.1.1.4.1 概述

AM RLC 实体能被配置为通过如下逻辑信道发送或接收 RLC PDUs, 确认模式实体的模型如图 15 所示:

——DCCH 或 DTCH

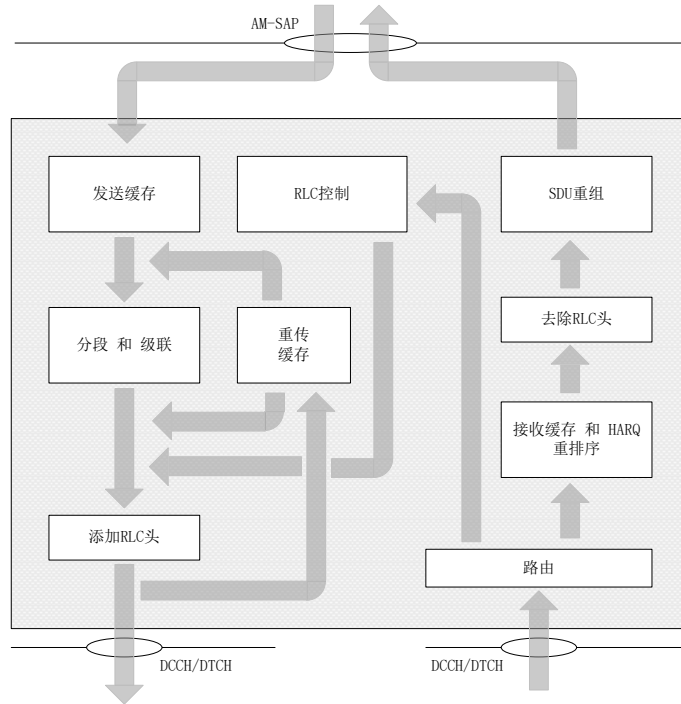


图 15 确认模式实体的模型

AM RLC 实体发送或接收如下的 RLC 数据 PDUs:

- AMD PDU;
- AMD PDU 段。

AM RLC 实体发送或接收如下的 RLC 控制 PDU:

- STATUS PDU。

8.1.1.4.2 发送部分

当 AM RLC 实体的发送部分从 RLC SDUs 构造 AMD PDUs 时，应：

- 对 RLC SDUs 进行分段和/或级联，使得 AMD PDUs 的大小适合于下层（MAC）通知的特定传输机会所指示的 RLC PDU(s)的总大小。

AM RLC 实体的发送部分支持 RLC 数据 PDUs 的重传（ARQ）：

- 若该要被重传的 RLC 数据 PDU 不适合于下层（MAC）通知的特定传输机会所指示的 RLC PDU(s)的总大小，AM RLC 实体能将该 RLC 数据 PDU 重分段为多个 AMD PDU 段；
- 重分段的分段数量没有限制。
- 当 AM RLC 实体的发送部分从接收来自上层的 RLC SDUs 中构造 AMD PDUs，或者从将被重传的 RLC 数据 PDUs 中构造 AMD PDU 段时，应：
- 在 RLC 数据 PDU 中包含相关的 RLC 头。

8.1.1.4.3 接收部分

当 AM RLC 实体的接收部分接收 RLC 数据 PDUs 时，应：

- 检测该 RLC 数据 PDUs 是否被重复接收，并丢弃重复的 RLC 数据 PDUs；
- 若被乱序接收，则对 RLC 数据 PDUs 进行重排序；
- 检测下层 RLC 数据 PDUs 的丢失，并向其对等的 AM RLC 实体请求重传；
- 从重排序后的 RLC 数据 PDUs 重组出 RLC SDUs，并且按顺序分发给上层。

8.1.2 服务

8.1.2.1 提供给上层的服务

由 RLC 提供给上层的服务如下：

- UM 数据传输；
- AM 数据传输，包括给上层提供 PDU_s 发送成功的指示。

8.1.2.2 期望从下层获得的服务

RLC 期望从下层（即 MAC）获得的服务如下：

- 数据传输；
- 通知 RLC 传输机会，及在该传输机会中将被发送 RLC PDU(s) 尺寸的总大小。

8.1.3 功能

RLC 子层支持以下功能：

- 上层 PDU_s 的传输；
- 通过 ARQ 纠错（仅适用于 AM 数据传输）；
- RLC SDU_s 的级联，分段和重组；
- RLC 数据 PDU_s 的重分段（仅适用于 AM 数据传输）；
- RLC 数据 PDU_s 的重排序；
- 重复检测；
- RLC SDU 丢弃；
- RLC RESET；
- 协议错误检测（仅适用于 AM 数据传输）。

8.2 过程

8.2.1 数据传输过程

8.2.1.1 UM 数据传输

8.2.1.1.1 发送操作

8.2.1.1.1.1 概述

当向下层分发一个新的 UMD PDU 时，发送 UM RLC 实体应：

- 1) 设置该 UMD PDU 的 SN 为 VT (US)，然后 VT (US) 加 1。

8.2.1.1.2 接收操作

8.2.1.1.2.1 概述

按照如下的状态变量 VR (UH)，接收 UM RLC 实体应维护一个重排序窗口：

- 1) 若 $(VR(UH) - UM_Window_Size) \leq SN < VR(UH)$ ，则 SN 落在该重排序窗口内；
- 1) 否则，SN 落在重排序的窗口外。

当从下层接收到一个 UMD PDU 时，接收 UM RLC 实体应：

- 1) 或者丢弃该接收到的 UMD PDU，或者把它放在接收缓存中（见 8.2.1.1.2.2 节）；

1)若该接收到的UMD PDU被放在接收缓存中:

1)更新状态变量, 重组并向上层分发RLC SDUs, 且按需要启动或停止t_Reordering (见 8.2.1.1.2.3节);

当 t_Reordering 超时, 接收 UM RLC 实体应:

1)更新状态变量, 重组且向上层分发RLC SDUs, 且按需要启动t_Reordering (详见 8.2.1.1.2.4节)。

8.2.1.1.2.2 从下层接收到 UMD PDU 时的操作

当从下层接收到 $SN = x$ 的 UMD PDU, 接收 UM RLC 实体应:

1)若 $VR(UR) < x < VR(UH)$ 且该 $SN = x$ 的UMD PDU在之前已被接收; 或

1)若 $(VR(UH) - UM_Window_Size) \leq x < VR(UR)$:

2)丢弃该接收到的 UMD PDU;

1)否则:

2)把该接收到的 UMD PDU 放在接收缓存中。

8.2.1.1.2.3 UMD PDU 被放在接收缓存时的操作

当 $SN = x$ 的 UMD PDU 放在接收缓存中时, 接收 UM RLC 实体应:

1)若 x 落在重排序窗口之外:

2)更新 $VR(UH)$ 为 $x + 1$;

2)从任何 SN 落在重排序窗口之外的 UMD PDUs 重组 RLC SDUs, 当这样做时去除 RLC 头, 且按 RLC SN 的升序, 向上层分发重组后的之前未分发过的 RLC SDUs;

1)若 $VR(UR)$ 落在重排序窗口之外:

2)置 $VR(UR)$ 为 $(VR(UH) - UM_Window_Size)$;

2)若接收缓存包含一个 UMD PDU, 其 $SN = VR(UR)$:

2)更新 $VR(UR)$ 为尚未接收到的, $SN >$ 当前 $VR(UR)$ 的, 第一个 UMD PDU 的 SN ;

2)从任何 $SN <$ 更新的 $VR(UR)$ 的 UMD PDUs 重组 RLC SDUs, 当这样做时去除 RLC 头, 且按 RLC SN 的升序, 向上层分发重组后的之前未分发过的 RLC SDUs;

1)若t-Reordering正在运行:

2)若 $VR(UX) \leq VR(UR)$; 或

2)若 $VR(UX)$ 落在重排序窗口之外, 并且 $VR(UX)$ 不等于 $VR(UH)$:

2)停止且重置 t_Reordering;

1)若t_Reordering未运行 (包括由于上述动作导致t_Reordering被停止时的情况):

2)若 $VR(UH) > VR(UR)$:

3)启动 t_Reordering;

3)设置 $VR(UX)$ 为 $VR(UH)$ 。

8.2.1.1.2.4 当 t_Reordering 超时的操作

当 t_Reordering 超时, 接收 UM RLC 实体应:

1)更新 $VR(UR)$ 为尚未接收到的, $SN \geq VR(UX)$ 的, 第一个UMD PDU的 SN ;

1)从任何SN<更新的VR(UR)的UMD PDUs重组RLC SDUs, 当这样做时去除RLC头, 且按RLC SN的升序, 向上层分发重组后的之前未分发过的RLC SDUs;

1)若VR(UH) > VR(UR):

2)启动 t_Reordering;

2)设置 VR(UX) 为 VR(UH)。

8.2.1.2 AM 数据传输

8.2.1.2.1 发送操作

8.2.1.2.1.1 概述

AM RLC 实体的发送部分应传输 RLC 控制 PDUs 优先于 RLC 数据 PDUs。AM RLC 实体的发送部分应传输重传 RLC 数据 PDUs 优先于新的 RLC 数据 PDUs。

按照如下状态变量 VT(A)和 VT(MS), AM RLC 实体的发送部分应维护一个发送窗口:

1)若VT(A) <= SN < VT(MS), 则SN落在传输窗口内;

1)否则, SN落在传输窗口外。

AM RLC 实体的发送部分不应向下层分发任何其 SN 落在传输窗口之外的 RLC 数据 PDU。

当向下层分发一个新的 AMD PDU 时, AM RLC 实体的发送部分应:

1)置AMD PDU的SN为VT(S), 然后VT(S)加1。

AM RLC 实体的发送部分能接收一个 RLC 数据 PDU 的肯定应答 (被对等 AM RLC 实体成功接收的确认), 通过如下:

来自它的对等 AM RLC 实体的 STATUS PDU。

1)当接收到一个AMD PDU的肯定应答, 其SN = VT(A), AM RLC实体的发送部分应:

2)在 SN 落在 VT(A)<=SN<=VT(S)的范围内, 置 VT(A)等于其肯定应答尚未接收到的 SN 最小的 AMD PDU 的 SN。

若一个传送的 RLC SDU 的所有相关 AMD PDUs 的肯定应答均已收到:

1)向上层发送该RLC SDU成功发送的指示。

8.2.1.2.2 接收操作

8.2.1.2.2.1 概述

按照如下状态变量 VR(R)和 VR(MR), AMR RLC 实体的接收部分应维护一个接收窗口:

1)若VR(R) <= SN < VR(MR), SN落在接收窗口内;

1)否则, SN落在接收窗口外。

当接收到来自下层的一个 RLC 数据 PDU, AM RLC 实体的接收部分应:

1)或者丢弃该接收到的RLC数据PDU, 或者把它放在接收缓存中 (见8.2.1.2.2.2节);

1)若该接收到的RLC数据PDU被放在接收缓存中:

2)更新状态变量, 重组并向上层传输 RLC SDUs, 且按需要启动或停止 t_Reordering (见 8.2.1.2.2.3 节)。

当 t_Reordering 超时, AM RLC 实体的接收部分应:

1)更新状态变量, 且按需要启动t_Reordering (见8.2.1.2.2.4节)。

8.2.1.2.2.2 从下层接收到 RLC 数据 PDU 时的操作

当从下层接收到 RLC 数据 PDU，其包含 SN=x 的 AMD PDU 的字节段序号 y 到 z，AM RLC 实体的接收部分应：

- 1)若x落在接收窗口之外；或
- 1)若SN=x的AMD PDU的字节段序号y到z在之前已被接收：
 - 2)丢弃该接收到的 RLC 数据 PDU；
- 1)否则：
 - 2)该接收的 RLC 数据 PDU 放在接收缓存中；
 - 2)若包含在该 RLC 数据 PDU 的 AMD PDU 的一些字节段之前已被接收：
 - 3)丢弃该重复的字节段。

8.2.1.2.2.3 RLC 数据 PDU 被放在接收缓存时的操作

当 SN=x 的 RLC 数据 PDU 被放在接收缓存时，AM RLC 实体的接收部分应：

- 1)若 $x \geq VR(H)$ ：
 - 2)更新 VR(H) 为 $x+1$ ；
- 1)若该SN = VR(MS)的AMD PDU的所有字节段被接收：
 - 2)更新 VR(MS) 为非所有字节段均被接收的， $SN >$ 当前 VR(MS)的，第一个 AMD PDU 的 SN。
- 1)若 $x = VR(R)$ ：
 - 2)若 SN = VR(R)的 AMD PDU 的所有字节段被接收：
 - 3)更新 VR(R) 为非所有字节段均被接收的， $SN >$ 当前 VR(R)的，第一个 AMD PDU 的 SN；
 - 3)更新 VR(MR)为更新的 $VR(R) + AM_Window_Size$ ；
 - 3)将 SN 落在接收窗口之外的所有 AMD PDU_s 重组为 RLC SDUs，并依次处理 SN=VR(R)的 AMD PDU 的字节段，当这样做时去除 RLC 头，且按序向上层分发重组后的之前未分发的 RLC SDUs；
- 1)若t_Reordering正在运行：
 - 2)若 $VR(X) = VR(R)$ ；或
 - 2)如 VR(X)落在接收窗口外，且 VR(X)不等于 VR(MR)：
 - 3)停止并重置 t_Reordering；
- 1)若t_Reordering未在运行（包括由于上述动作导致t_Reordering停止时的情况）：
 - 2)若 $VR(H) > VR(R)$ ：
 - 3)启动 t_Reordering；
 - 3)置 VR(X)为 VR(H)。

8.2.1.2.2.4 当 t_Reordering 超时的操作

当 t_Reordering 超时，AM RLC 实体的接收部分应：

- 1)更新VR(MS)为非所有字节段均已被接收的， $SN \geq VR(X)$ 的，第一个AMD PDU的SN；
- 1)若 $VR(H) > VR(MS)$ ：

- 2)启动 $t_{\text{Reordering}}$;
- 2)设置 $VR(X)$ 为 $VR(H)$ 。

8.2.2 ARQ 过程

ARQ 过程仅由AM RLC实体执行。

8.2.2.1 重传

AM RLC 实体的发送部分能接收一个 AMD PDU 或一个部分 AMD PDU 的否定应答（其对等 AM RLC 实体的接收失败通知），通过如下：

1)来自对等AM RLC实体的STATUS PDU。

1)当通过来自对等AM RLC实体的STATUS PDU，接收到一个AMD PDU或一个部分AMD PDU的否定应答，AM RLC实体的发送部分应：

2)若相应 AMD PDU 的 SN 落在 $VT(A) \leq SN < VT(S)$ 范围内：

3)认为该接收到否定应答的 AMD PDU 或部分 AMD PDU，用于重传。

当一个 AMD PDU 或一个部分 AMD PDU 被认为用于重传，AM RLC 实体的发送部分应：

1)若被认为用于重传的AMD PDU为首次重传：

2)置与该 AMD PDU 相关的 RETX_COUNT 为 0；

1)否则，若它（被认为重传的该 AMD PDU 或该 AMD PDU 的部分）已确定要重传，或它的部分已确定要重传：

2)RETX_COUNT 加 1；

2)若 $RETX_COUNT = \text{maxRetxThreshold}$ ：

3)向上层指示已达到最大重传。

当重传一个 AMD PDU 时，AM RLC 实体的发送部分应：

1)若该AMD PDU能完全适合于下层在该特定传输机会所指示的RLC PDU(s)的总大小：

2)除 P 域外（P 域因按照 8.2.2.2 节设置），向下层分发该 AMD PDU；

1)否则：

2)对该 AMD PDU 分段，形成一个新的 AMD PDU 分段，使其将适合于下层在该特定传输时刻指示的 RLC PDU(s)的总大小，并分发该新 AMD PDU 分段到下层。

当重传 AMD PDU 的部分时，AM RLC 实体的发送部分应：

1)必要时对AMD PDU的部分进行分段，形成新的AMD PDU分段，使其将适合于下层在该特定传输时刻指示的RLC PDU(s)的总大小，并分发该新AMD PDU分段到下层。

当形成一个新的AMD PDU分段时，AM RLC实体的发送部分应：

1)仅映射原AMD PDU的数据域到新AMD PDU段的数据域；

1)置新AMD PDU段的头，与在第8.3节的描述一致；

1)按照 8.2.2.2节设置P域。

8.2.2.2 轮询

一个 AM RLC 实体能轮询它的对等 AM RLC 实体，以触发该对等 AM RLC 实体的 STATUS 报告。

8.2.2.2.1 AMD PDU 或 AMD PDU 段的发送

对于一个新 AMD PDU 的组装，AM RLC 实体的发送部分应：

- 1) PDU_WITHOUT_POLL 增加1；
- 1) BYTE_WITHOUT_POLL 增加，映射到该RLC数据PDU的数据域的，数据域元的每个新字节；
- 1) 若 PDU_WITHOUT_POLL \geq pollPDU；或
- 1) 若 BYTE_WITHOUT_POLL \geq pollByte；
 - 2) 按如下描述，在该 RLC 数据 PDU 中包含一个轮询。

一旦一个 AMD PDU 或 AMD PDU 段组装，AM RLC 实体的发送部分应：

- 1) 若在该RLC数据PDU发送之后，发送缓存和重传缓存均空（排除等待确认的已发送RLC数据PDU）；或
- 1) 若在该RLC数据PDU发送之后，无新RLC数据PDU能被发送（例如由于窗口延迟（window stalling））；
 - 2) 按如下描述，在该 RLC 数据 PDU 中包含一个轮询。

为了在一个 RLC 数据 PDU 中包含一个轮询，AM RLC 实体的发送部分应：

- 1) 置RLC数据PDU的P域为"1"；
- 1) 置PDU_WITHOUT_POLL为0；
- 1) 置BYTE_WITHOUT_POLL为0；

在包含一个轮询的 RLC 数据 PDU 分发到下层之后，且若必要的 VT(S)增加之后，AM RLC 实体的发送部分应：

- 1) 置 POLL_SN 为 VT(S) - 1；
- 1) 若 t_pollRetransmit 未在运行：
 - 2) 启动 t_pollRetransmit；
- 1) 否则：
 - 2) 重启 t_pollRetransmit；

8.2.2.2.2 STATUS 报告的接收

一旦从 RLC AM 接收实体接收到一个 STATUS 报告，AM RLC 实体的发送部分应：

- 1) 若该STATUS报告包含一个RLC数据PDU的确定或否定应答，其序列号等于POLL_SN：
 - 2) 若 t_PollRetransmit 在运行：
 - 3) 停止并重置 t_PollRetransmit。

8.2.2.2.3 t_PollRetransmit 的超时

一旦 t_PollRetransmit 超时，AM RLC 实体的发送部分应：

- 1) 若发送缓存和重传缓存均空（排除等待确认的已发送RLC数据PDU）；或
- 1) 若无新RLC数据PDU能被发送（例如由于窗口延迟）；
 - 2) 认为该 SN= VT(S)-1 的 AMD PDU 用于重传，或
 - 2) 认为任何未被肯定应答的 AMD PDU 用于重传；
 - 2) 如 8.2.2.2.1 节所述，在一个 RLC 数据 PDU 中包含一个轮询。

8.2.2.3 状态报告。

AM RLC 实体向其对等 AM RLC 实体发送 STATUS PDUs，以提供 RLC PDUs（或它们的部分）的肯定和/或否定的应答确认。

RRC 配置是否状态禁止功能被用于一个 AM RLC 实体。

初始化 STATUS 报告的触发包括：

1)来自于其对等AM RLC实体的轮询：

1)当从下层接收RLC数据PDU，其SN = x，且P域置为“1”，AM RLC实体的接收部分应：

2)若该 PDU 将被丢弃，如 8.2.1.2.2.2 节所规定；或

2)若 $x < VR(MS)$ 或 $x \geq VR(MR)$ ：

3)触发一个 STATUS 报告；

2)否则：

3)延迟触发该 STATUS 报告，直至 $x < VR(MS)$ 或 $x \geq VR(MR)$ 。

注 1：这确保 RLC 状态报告在 HARQ 重排序之后发送。

1)检测RLC数据PDU的接收失败：

1)当t_Reordering超时，AM RLC实体的接收部分应触发一个STATUS报告。

注 2：t_Reordering 的超时触发 VR(MS)的更新，并触发一个 STATUS 报告，但该 STATUS 报告应在 VR(MS)被更新后触发。

当已触发了 STATUS 报告，AM RLC 实体的接收部分应：

1)若t_StatusProhibit未在运行：

2)在下层指示的首个传输机会，构造一个 STATUS PDU，并向下层分发；

1)否则：

2)在 t_StatusProhibit 超时后由下层指示的首个传输机会，即使 t_StatusProhibit 运行时状态报告被触发多次，也构造一个单独的 STATUS PDU，并向下层分发；

当一个 STATUS PDU 已被分发到下层，AM RLC 实体的接收部分应：

1)启动t_StatusProhibit。

当构造一个 STATUS PDU 时，AM RLC 实体应：

对于 SN 为 $VR(R) \leq SN < VR(MS)$ 的尚未被完全接收的 AMD PDUs，按 PDUs 的 SN 增序和 PDUs 内的字节段增序，从 SN=VR(R)开始，直到作为结果的 STATUS PDU 仍然适合于下层指示的 RLC PDU(s) 总大小的那一点：

1)对于一个无字节段已被接收的AMD PDU：

2)在 STATUS PDU 中包含一个 NACK_SN，其被设置为该 AMD PDU 的 SN；

1)对于一个部分被接收AMD PDU的，尚未被接收的字节段的连续序列：

2)在 STATUS PDU 中包含一个 NACK_SN，SOstart 和 SOend 的集合；

2)置 ACK_SN 为，未在作为结果的 STATUS PDU 中指示为丢失的，下一个未被接收的 RLC 数据 PDU 的 SN。

8.2.3 SDU 丢弃过程

当从上层（如 PDCP）指示丢弃一个特定的 RLC SDU 时，若该 RLC SDU 没有任意一部分已经被

映射到一个RLC PDU上,则AM RLC实体的发送部分或发送UM RLC实体应丢弃该被指示的RLC SDU。

8.2.4 Reset 过程

8.2.4.1 概述

RLC RESET 过程可以由本节点 RLC 实例或其他节点 RLC 实例触发。

8.2.4.2 RLC 发起的 RESET

8.2.4.2.1 本节点 RLC 发起的 RESET

本节点RLC判断对端节点RLC信息异常时,发起RLC RESET过程(详见8.2.4.3节)

8.2.4.2.2 其他点 RLC 发起的 RESET

本节点RLC收到对端节点发送的RESET PDU,发起RLC RESET过程(详见8.2.4.3节)

8.2.4.2.3 RESET 执行过程

- 1)丢弃所有RLC SDUs;
- 1)停止并重置所有定时器;
- 1)重置所有状态变量为它们的初始值。
- 1)若是本节点RLC发起的RESET过程
 - 2)向对端节点 RLC 发送 RESET PDU。
- 1)通知MAC进行RESET

8.2.5 处理未知的、未预见的及错误的协议数据

当 RLC 实体接收到一个包含保留的或无效值的 RLC PDU, RLC 实体应:

- 1)丢弃该接收到的PDU。

8.3 协议数据单元、格式及参数

8.3.1 协议数据单元

RLC PDUs 能被分类为 RLC 数据 PDUs 和 RLC 控制 PDUs。在 8.3.1.1 节中的 RLC 数据 PDUs 用于 UM 和 AM RLC 实体传输上层 PDUs (即 RLC SDUs)。在 8.3.1.2 节中的 RLC 控制 PDUs 用于 AM RLC 实体执行 ARQ 过程。

8.3.1.1 RLC 数据 PDU

——UMD PDU

UMD PDU 用于 UM RLC 实体传输上层 PDU。

——AMD PDU

AMD PDU 用于 AM RLC 实体传输上层 PDU。它被用在 AM RLC 实体首次传输 (部分的) 该 RLC SDU 时,或 AM RLC 实体重传一个无须执行重分段的 AMD PDU 时。

——AMD PDU 段

AMD PDU 段用于 AM RLC 实体传输上层 PDU。它被用在 AM RLC 实体需要重传 AMD PDU 的部分时。

8.3.1.2 RLC 控制 PDU

——STATUS PDU

STATUS PDU 用于 AM RLC 实体的接收部分向对等 AM RLC 实体,通知关于 RLC 数据 PDU 已被成功接收的信息,和被 AM RLC 实体的接收部分检测到丢失的 RLC 数据 PDU 的信息。

8.3.2 格式和参数

RLC PDU 的格式见 8.3.2.1 节,其参数见 8.3.2.2 节。

8.3.2.1 格式

8.3.2.1.1 概述

RLC PDU 是一个比特串。在 8.3.2.1.2 至 8.3.2.1.5 节的图中,比特串用表格表示,其中第一位和最高有效位为该表格第一行最左边的比特,最后一位和最低有效位为该表格的最后一行最右边的比特,并且多数情况下,比特串从左到右读取,之后再按行的顺序读取。

RLC SDU 为按字节对齐的比特串(即为 8 比特的倍数)。一个 RLC SDU 从前面的首个比特开始被包含于一个 RLC PDU。

8.3.2.1.2 UMD PDU

UMD PDU 包含一个数据域和一个 UMD PDU 头。

UMD PDU 头包含一个固定部分(在每个 UMD PDU 存在的域)和一个扩展部分(一个 UMD PDU 必要时才存在的域)。UMD PDU 头的固定部分自身按字节对齐,且由一个 FI、一个 E 和一个 SN 构成。UMD PDU 头的扩展部分自身按字节对齐,包含 E(s)和 LI(s)。

UM RLC 实体由 RRC 配置,使用 5 比特的 SN 或 10 比特的 SN。当配置使用 5 比特 SN 时,UMD PDU 头的固定部分的长度为 1 字节,如图 16 所示。当配置使用 10 比特 SN 时,UMD PDU 头固定部分的长度与 AMD PDU 头固定部分的长度相同,都被换成 R1 域的 D/C、Radio Frame 和 P 域除外,如图 17 所示。UMD PDU 头的扩展部分与 AMD PDU 头扩展部分的长度相同(忽略配置的 SN 大小)。

UMD PDU 头包含一个扩展部分,仅当 UMD PDU 中存在多于一个的数据域元时,此时每个数据域元,除了最后一个,均存在一个 E 和一个 LI。此外,当一个 UMD PDU 头包含奇数个 LI(s)时,在最后的 LI 后面填充 4 个填充位。SN 为 5 比特的 UMD PDU (奇数个 LI,即 $K = 1, 3, 5, \dots$)如图 18 所示,SN 为 5 比特的 UMD PDU (偶数个 LI,即 $K = 1, 3, 5, \dots$)如图 19 所示,SN 为 10 比特的 UMD PDU (奇数个 LI,即 $K = 1, 3, 5, \dots$)如图 20 所示,SN 为 10 比特的 UMD PDU (偶数个 LI,即 $K = 1, 3, 5, \dots$)如图 21 所示。

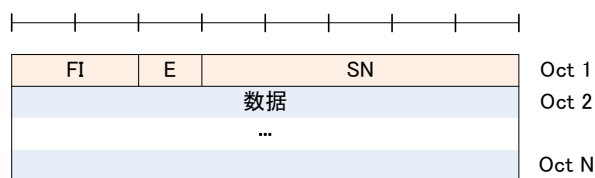


图 16 SN 为 5 比特的 UMD PDU (无 LI)

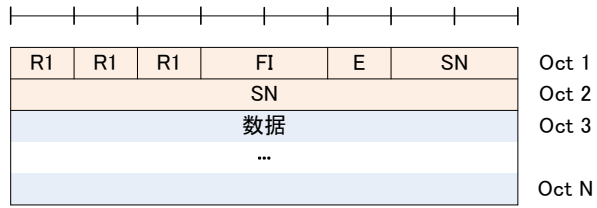


图 17 SN 为 10 比特的 UMD PDU (无 LI)

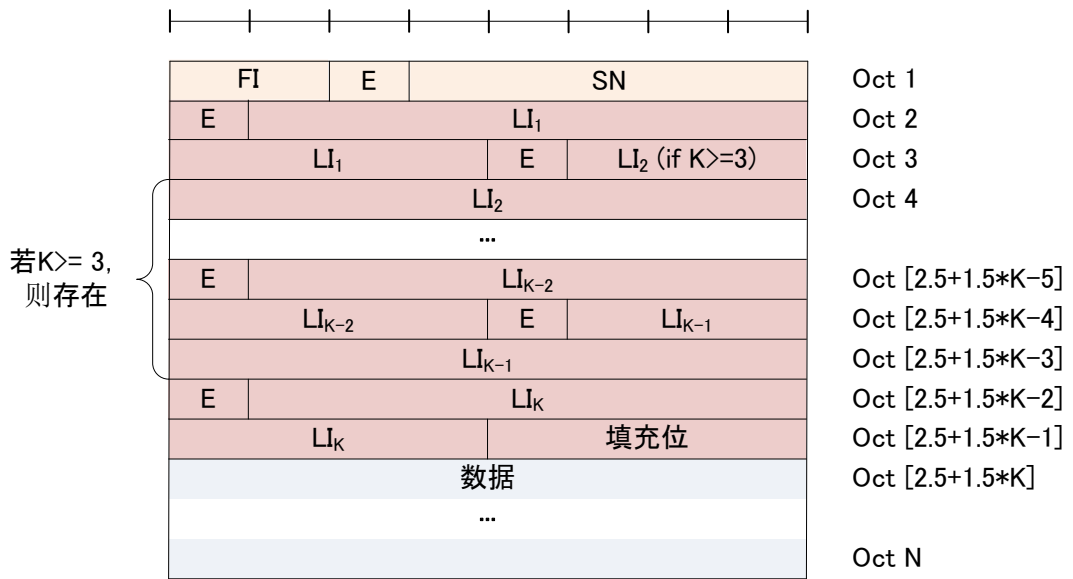


图 18 SN 为 5 比特的 UMD PDU (奇数个 LI, 即 $K = 1, 3, 5, \dots$)

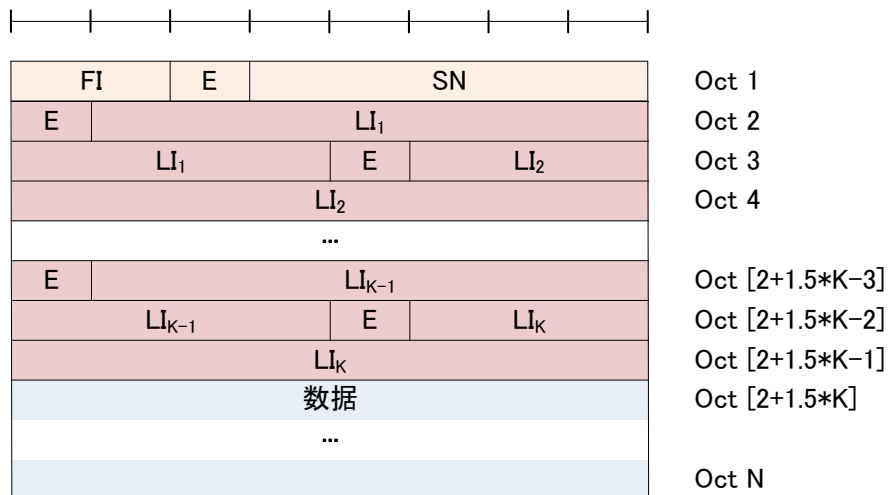


图 19 SN 为 5 比特的 UMD PDU (偶数个 LI, 即 $K = 2, 4, 6, \dots$)

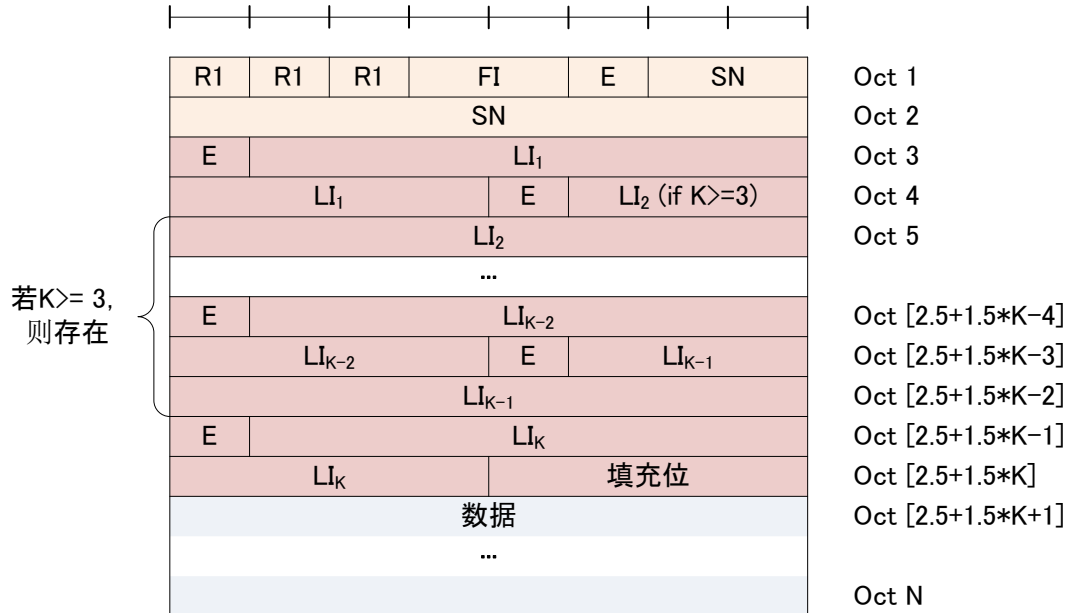


图 20 SN 为 10 比特的 UMD PDU (奇数个 LI, 即 $K = 1, 3, 5, \dots$)

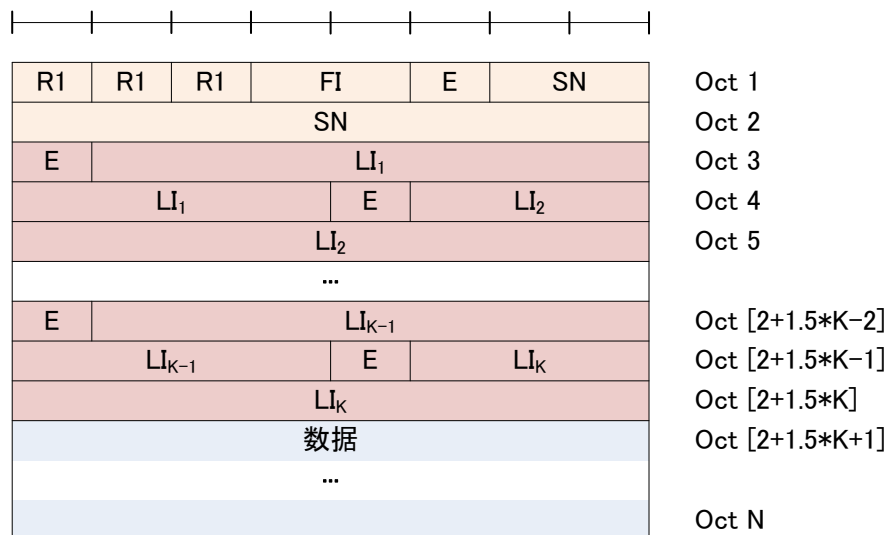


图 21 SN 为 10 比特的 UMD PDU (偶数个 LI, 即 $K = 2, 4, 6, \dots$)

8.3.2.1.3 AMD PDU

AMD PDU 包含一个数据域和一个 AMD PDU 头。

AMD PDU 头包含一个固定部分 (在每个 AMD PDU 存在的域) 和一个扩展部分 (一个 AMD PDU

必要时才存在的域)。AMD PDU 头的固定部分自身按字节对齐,且包含一个 D、一个 Radio Frame、一个 P、一个 FI、一个 E 和一个 SN。AMD PDU 头自身按字节对齐, 包含 E(s)和 LI(s)。

AMD PDU 头包含一个扩展部分, 仅当 AMD PDU 中存在多于一个的数据域元时, 此时每个数据域元, 除了最后一个, 均存在一个 E 和一个 LI, 如图 22 所示。此外, 当一个 AMD PDU 头包含奇数个 LI(s)时, 在最后的 LI 后面填充 4 个填充位。AMD PDU (奇数个 LI, 即 $K = 1, 3, 5, \dots$) 如图 23 所示, AMD PDU (偶数个 LI, 即 $K = 1, 3, 5, \dots$) 如图 24 所示。

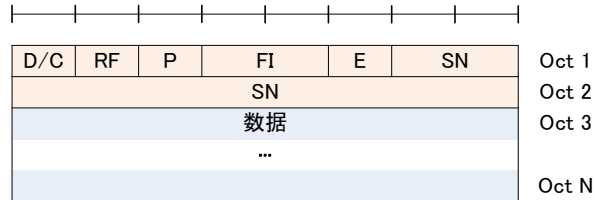


图 22 AMD PDU (无 LI)

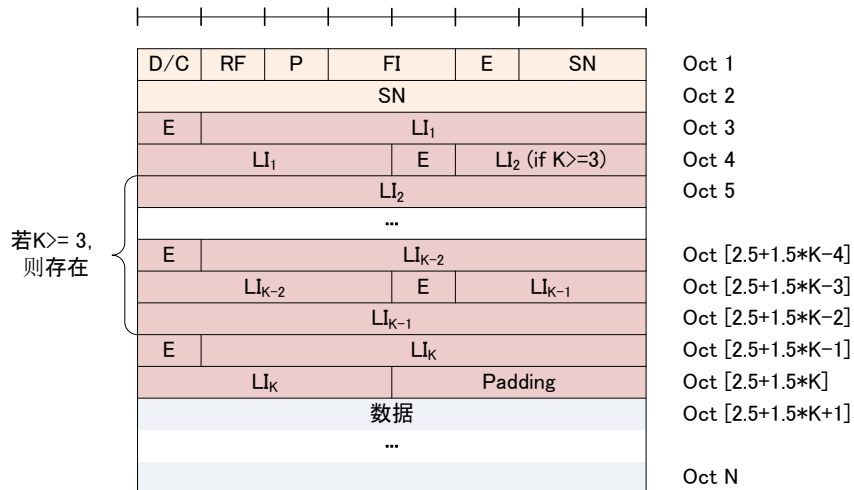


图 23 AMD PDU (奇数个 LI, 即 $K = 1, 3, 5, \dots$)

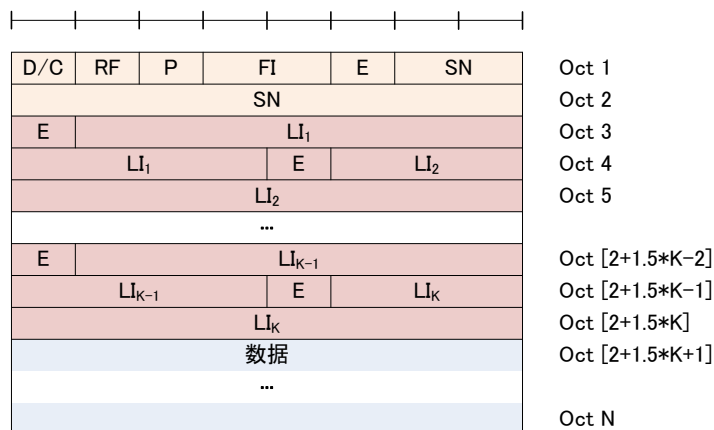


图 24 AMD PDU (偶数个 LI, 即 $K = 2, 4, 6, \dots$)

8.3.2.1.4 AMD PDU 段

AMD PDU 段包含一个数据域和一个 AMD PDU 段头。

AMD PDU 段头包含一个固定部分(在每个 AMD PDU 分段中存在的域)和一个扩展部分(一个 AMD PDU 分段必要时才存在的域)。AMD PDU 分段头的固定部分自身按字节对齐,且包含一个 D/C、一个 Radio Frame、一个 P、一个 FI、一个 E、一个 SN、一个 LSF 和一个 SO。AMD PDU 段头的扩展部分自身按字节对齐,包含 E(s)和 LI(s)。

AMD PDU 段头包含一个扩展部分,仅当 AMD PDU 段中存在多于一个数据域元时,此时每个数据域元,除了最后一个外,均存在一个 E 和一个 LI,如图 25 所示。此外,当一个 AMD PDU 头包含奇数个 LI(s)时,在最后的 LI 后面填充 4 个填充位。AMD PDU 分段(奇数个 LI,即 $K = 1, 3, 5, \dots$)如图 26 所示,AMD PDU 分段(偶数个 LI,即 $K = 2, 4, 6, \dots$)如图 27 所示,

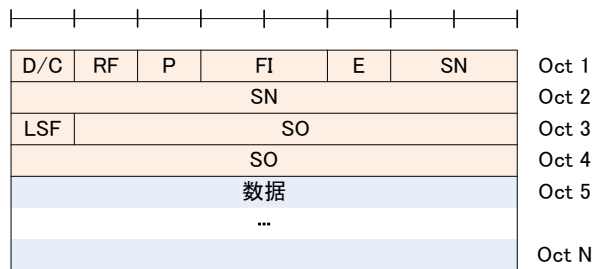


图 25 AMD PDU 分段 (无 LI)

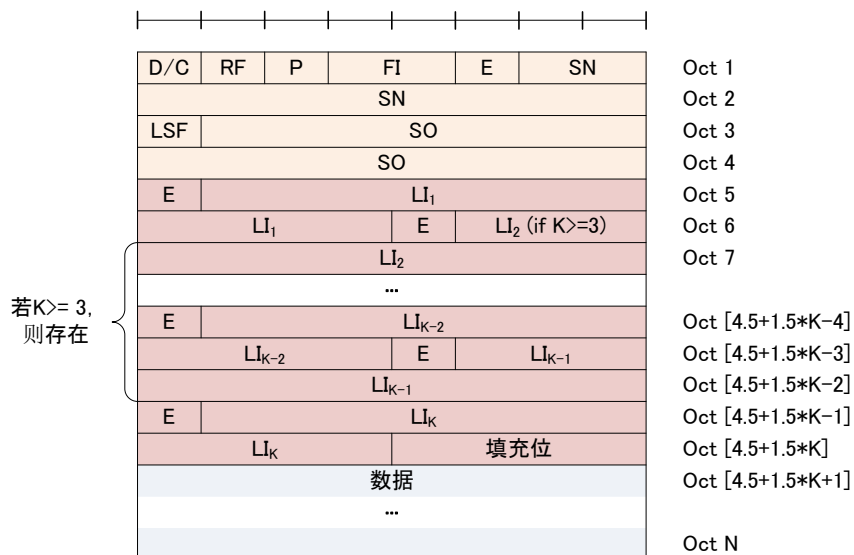
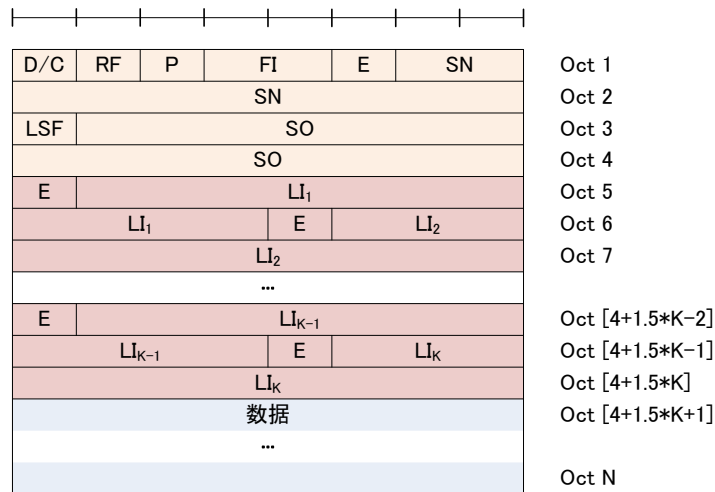


图 26 AMD PDU 分段 (奇数个 LI, 即 $K = 1, 3, 5, \dots$)

图 27 AMD PDU 分段（偶数个 LI，即 $K = 2, 4, 6, \dots$ ）

8.3.2.1.5 STATUS PDU

STATUS PDU 包含一个 STATUS PDU 载荷和一个 RLC 控制 PDU 头。

RLC 控制 PDU 头包含一个 D/C 和一个 CPT 域。

STATUS PDU 载荷从 RLC 控制 PDU 头之后的第一个比特开始，其包括一个 ACK_SN 和一个 E1，零个或多个由一个 NACK_SN、一个 E1 和一个 E2 组成的集合，可能还包括用于每个 NACK_SN 的一个 SOstart 和一个 SOend 的集合，如图 28 所示。若必要，1 至 7 个填充位包含于 STATUS PDU 的末尾部分，以满足按字节对齐。

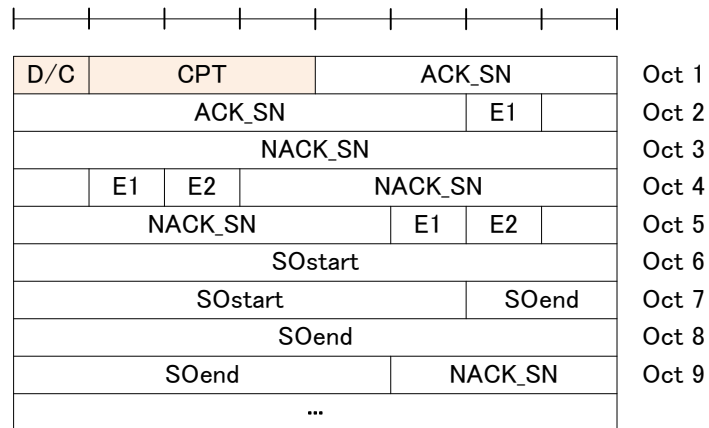


图 28 STATUS PDU

8.3.2.1.6 RESET PDU

RESET PDU 长度约定为 2 字节，其中包含一个 RLC 控制 PDU 头。

RLC 控制 PDU 头包含一个 D/C 和一个 CPT 域，如图 29 所示。

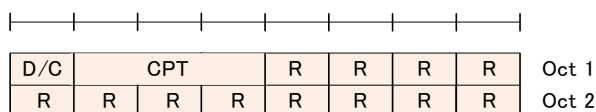


图 29 RESET PDU

8.3.2.2 参数

8.3.2.2.1 概述

8.3.2.2.2 至 8.3.2.2.19 节每个域的定义中，参数中的比特的表示，其第一和最高有效位为最左边的比特，最后一位和最低有效位为最右边的比特。除非额外说明，否则整数按无符号整数的标准二进制编码方式进行编码。

8.3.2.2.2 数据域

数据域元按其在发送端到达 RLC 实体的顺序映射到数据域。

对于 UMD PDU 和 AMD PDU：

- 数据域大小的粒度为一个字节；
- 最大的数据域大小，为最大 TB 大小减去，最小 MAC PDU 头大小与最小 RLC PDU 头大小的和。

对于 UMD PDU，AMD PDU 和 AMD PDU 段：

- 如下的任何一个，能被映射到一个 UMD PDU，AMD PDU 或 AMD PDU 段的数据域：
 - 零个 RLC SDU 段和一个或多个 RLC SDUs；
 - 一个或两个 RLC SDU 段和零个或多个 RLC SDUs；
- RLC SDU 段映射到该数据域的开始部分，或者末尾部分；
- 一个大于 2047 八位组的 RLC SDU 或 RLC SDU 段只能被映射到数据域的末尾部分；
- 当有两个 RLC SDU 段时，它们属于不同的 RLC SDUs。

8.3.2.2.3 序列号 (SN) 域

长度：对于 AMD PDU 和 AMD PDU 段，为 10 比特；对于 UMD PDU，为 5 比特或 10 比特（可配置）。

SN 域指示相应 UMD 或 AMD PDU 的序列号。对于一个 AMD PDU 段，其 SN 域指示构造该 AMD PDU 段的原始 AMD PDU 的序列号。对于每个 UMD 或 AMD PDU，序列号逐增加一。

8.3.2.2.4 扩展比特 (E) 域

长度：1 比特。

该 E 域指示其后跟随的是数据域，还是一个 E 域和 LI 域的集合。E 域的解释见表 26 和表 27。

表 26 E 域解释（对于信息头中的固定部分中的 E 域）

值	描述
0	数据域从信息头固定部分之后八位组开始
1	一个 E 域和 LI 域的集合从信息头固定部分之后的八位组开始

表 27 E 域解释（对于信息头中的扩展部分的 E 域）

值	描述
0	数据域从跟随于 E 域的 LI 域之后八位组开始
1	一个 E 域和 LI 域的集合从本 E 域后的 LI 域之后的比特开始

8.3.2.2.5 长度指示（LI）域

长度：11 比特。

LI 域指示相应数据域元的字节长度，该数据域元出现在由一个 UM 或 AM RLC 实体来发送/接收的 RLC 数据 PDU 中。出现在 RLC 数据 PDU 头中的第一个 LI，对应于出现在 RLC 数据 PDU 数据域的第一个数据域元；出现在 RLC 数据 PDU 头中的第二个 LI，对应于出现在 RLC 数据 PDU 数据域的第二数据域元；以此类推。值 0 被保留。

8.3.2.2.6 成帧信息（FI）域

长度：2 比特。

FI 域指示一个 RLC SDU 是否是在数据域的开始部分，和/或在数据域的末尾部分被分段。特别地，该 FI 域指明该数据域的第一个字节，是否对应 RLC SDU 的第一个字节，以及该数据域的最后一个字节，是否对应 RLC SDU 的最后一个字节。FI 域的解释见表 28。

表 28 FI 域解释

值	描述
00	该数据域的第一个字节，对应于 RLC SDU 的第一个字节。 该数据域的最后一个字节，对应于 RLC SDU 的最后一个字节。
01	该数据域的第一个字节，对应于 RLC SDU 的第一个字节。 该数据域的最后一个字节，不对应于 RLC SDU 的最后一个字节。
10	该数据域的第一个字节，不对应于 RLC SDU 的第一个字节。 该数据域的最后一个字节，对应于 RLC SDU 的最后一个字节。
11	该数据域的第一个字节，不对应于 RLC SDU 的第一个字节。 该数据域的最后一个字节，不对应于 RLC SDU 的最后一个字节。

8.3.2.2.7 段偏移（SO）域

长度：15 比特。

SO 域以字节为单位，指示在原始 AMD PDU 内 AMD PDU 段的位置。特别地，SO 域指示在原始 AMD PDU 数据域内，相对于对应的 AMD PDU 段的数据域首字节的位置。原始 AMD PDU 数据域中的首字节，SO 域值表示为"000000000000000"，即从 0 开始计数。

8.3.2.2.8 末段标志（LSF）域

长度：1 比特。

LSF 域指示 AMD PDU 段的最后字节是否对应于一个 AMD PDU 的最后字节。LSF 域的解释见表 29。

表 29 LSF 域解释

值	描述
0	AMD PDU 分段的最后一个字节，不对应于 AMD PDU 的最后一个字节。
1	AMD PDU 分段的最后一个字节，对应于 AMD PDU 的最后一个字节。

8.3.2.2.9 数据/控制 (D/C) 域

长度：1 比特。

D/C 域指示 RLC PDU 是否是 RLC 数据 PDU，或是 RLC 控制 PDU。D/C 域的解释见表 30。

表 30 D/C 域解释

值	描述
0	控制 PDU
1	数据 PDU

8.3.2.2.10 重分段标志 (Radio Frame) 域

长度：1 比特。

Radio Frame 域指示 RLC PDU 是否是 AMD PDU 或是 AMD PDU 段。Radio Frame 域的解释见表 31。

表 31 Radio Frame 域解释

值	描述
0	AMD PDU
1	AMD PDU 段

8.3.2.2.11 轮询比特 (P) 域

长度：1 比特。

P 域指示 AM RLC 实体的发送部分是否从它的对等 AM RLC 实体请求一个 STATUS 报告。P 域的解释见表 32。

表 32 P 域解释

值	描述
0	没有请求 Status 报告
1	请求了 Status 报告

8.3.2.2.12 保留 (R) 域

长度：1 比特。

对于本协议的此版本，R 域是保留域。接收实体应忽略此域。

8.3.2.2.13 保留 1 (R1) 域

长度：1 比特。

对于本协议的此版本，R1 域是保留域。发送实体应置 R1 域为"0"。接收实体应忽略此域。

8.3.2.2.14 控制 PDU 类型 (CPT) 域

长度：3 比特。

CPT 域指示 RLC 控制 PDU 的类型。CPT 域的解释见表 33。

表 33 CPT 域解释

值	描述
000	STATUS PDU
001	RESET PDU
002-111	保留（对于本协议的此版本，接收实体将丢弃带有该编码的 PDU 的）

8.3.2.2.15 确认 SN (ACK_SN) 域

长度：10 比特。

ACK_SN 域指示下一个未收到的，未在 STATUS PDU 中被报告为丢失的，RLC 数据 PDU 的 SN。当 AM RLC 实体的发送部分收到一个 STATUS PDU，这意味着它的对等 AM RLC 实体已接收到，直到但不包括 SN = ACK_SN 的，所有的 AMD PDU，除了那些在 STATUS PDU 使用 NACK_SN 指示的 AMD PDU 的，和在 STATUS PDU 使用 NACK_SN、SOstart 和 SOend 指示的 AMD PDU 的部分。

8.3.2.2.16 扩展比特 (E1) 域

长度：1 比特。

E1 域指示其后是否其后跟随着一个 NACK_SN，E1 和 E2 的集合。E1 域的解释见表 34。

表 34 E1 域解释

值	描述
0	其后未跟随着一个 NACK_SN，E1 和 E2 集合。
1	其后跟随着一个 NACK_SN，E1 和 E2 集合。

8.3.2.2.17 否定应答 SN (NACK_SN) 域

长度：10 比特。

NACK_SN 域指示 AM RLC 实体的接收部分检测为丢失的 AMD PDU（或其部分）的 SN。

8.3.2.2.18 扩展比特(E2)域

长度：1 比特。

E2 域指示其是否其后跟随着一个 SOstart 和 SOend 的集合。E2 域的解释见表 35。

表 35 E2 域解释

值	描述
0	此 NACK_SN，未跟随着一个 SOstart 和 SOend 集合。
1	此 NACK_SN，跟随着一个 SOstart 和 SOend 集合。

8.3.2.2.19 SO 开始 (SOstart) 域

长度：15 比特。

SOstart 域（与 SOend 域一起）指示在 AM RLC 实体的接收部分已检测为丢失的，SN = NACK_SN（和此 SOstart 相联的 NACK_SN）的 AMD PDU 的部分。特别地，SOstart 域指示 AMD PDU 数据域内，以字节为单位，AMD 部分的第一个字节的位置。原始 AMD PDU 数据域的第一个字节，SOstart 域值表示为"000000000000000"，即从 0 开始计数。

8.3.2.2.20 SO 结束 (SOend) 域

长度：15 比特。

SOend 域（与 SOstart 域一起）指示在 AM RLC 实体的接收部分已检测为丢失的，SN = NACK_SN（和此 SOend 相联的 NACK_SN）的 AMD PDU 的部分。特别地，SOend 域指示 AMD PDU 数据域内，以字节为单位，AMD 部分的最后一个字节的位置。原始 AMD PDU 数据域的第一个字节，SOend 域值表示为"000000000000000"，即从 0 开始计数。特定的 SOend 值"111111111111111"，用于指示 AMD PDU 的丢失部分包括直至 AMD PDU 末字节的所有字节。

8.4 变量、常量和定时器

8.4.1 状态变量

本节描述用于 AM 和 UM 实体中的状态变量，以规范 RLC 协议。本节定义的状态变量为规范性的。所有状态变量和所有计数器为非负整数。

所有与 AM 数据传输相关的状态变量，能从 0 到 1023 取值。在本部分中包含的，与 AM 数据传输相关的，状态变量的所有算术运算，均按 AM 模数进行模运算（即，最终值=[算术运算得到的值] 模 1024）。

所有与 UM 数据传输相关的状态变量，能从 0 到 $2^{[sn-FieldLength]} - 1$ 取值。在本部分中包含的，与 UM 数据传输相关的，状态变量的所有算术运算，均按 UM 模数进行模运算（即，最终值=[算术运算得来的值] 模 $2^{[sn-FieldLength]}$ ）。

AMD PDU 和 UMD PDU 被整数序列号 (SN) 循环计数：对于 AMD PDU 从 0 到 1023，对于 UMD PDU 从 0 到 $2^{[sn-FieldLength]} - 1$ 。

当进行状态变量或 SN 值的算术比较时，应使用一个模运算基数。

VT(A)和 VR(R)应分别假定为 AM RLC 实体的发送部分和接收部分的模运算基数。该模基数从所有涉及的值中减去，然后进行绝对比较（如， $VR(R) \leq SN < VR(MR)$ 求值于 $[VR(R) - VR(R)] \text{ 模 } 1024 \leq [SN - VR(R)] \text{ 模 } 1024 < [VR(MR) - VR(R)] \text{ 模 } 1024$ ）。

$VR(UH) - UM_Window_Size$ 应在一个 UM RLC 实体的接收部分被假定为模运算基数。该模基数被从所有涉及的值减去，然后进行绝对比较（如， $(VR(UH) - UM_Window_Size) \leq SN < VR(UH)$ 求值于 $[(VR(UH) - UM_Window_Size) - (VR(UH) - UM_Window_Size)] \text{ 模 } 2^{[sn-FieldLength]} \leq [SN - (VR(UH) - UM_Window_Size)] \text{ 模 } 2^{[sn-FieldLength]} < [VR(UH) - (VR(UH) - UM_Window_Size)] \text{ 模 } 2^{[sn-FieldLength]}$ ）。

每个 AM RLC 实体的发送部分应维护如下状态变量：

——VT(A) - 确认状态变量

该状态变量保持下一个按序将接收到一个肯定应答的 AMD PDU 的 SN 值，且其作为发送窗口的下

边界。该状态变量初始置为 0，且当任何时候该 AM RLC 实体接收到一个 $SN = VT(A)$ 的 AMD PDU 的肯定应答时，被更新。

——VT(MS) - 最大发送状态变量

该状态变量等于 $VT(A) + AM_Window_Size$ ，且其作为发送窗口的上边界。

——VT(S) - 发送状态变量

该状态变量保持分配给下一个新产生的 AMD PDU 的 SN 值。该状态变量初始置为 0，且当任何时候该 AM RLC 实体分发一个 $SN = VT(S)$ 的 AMD PDU 时，被更新。

——POLL_SN - 轮询发送状态变量

在最近的轮询比特置为“1”的 RLC 数据 PDU 发送时，该状态变量保持 $VT(S)-1$ 的值。其初始置为 0。

每个 AM RLC 实体的发送部分应维护如下计数器：

——PDU_WITHOUT_POLL - 计数器

此计数器初始置为 0。它从最近被发送的轮询比特开始，计数发送的 AMD PDU。

——BYTE_WITHOUT_POLL - 计数器

此计数器初始置为 0。它从最近被发送的轮询比特开始，计数发送的数据字节。

——RETX_COUNT - 计数器

该计数器计数一个 AMD PDU（见 8.2.2.1 节）的重传次数。每个需要被重传的 PDU 有一个 RETX_COUNT 计数器。

每一个 AM RLC 实体的接收部分应维护如下状态变量：

——VR(R) - 接收状态变量

该状态变量保持的 SN 值，紧跟在按序被完全接收的最后 AMD PDU 之后，且其作为接收窗口的下边界。其初始置为 0，且当任何时候 AM RLC 实体收到 $SN = VR(R)$ 的 AMD PDU 时，被更新。

——VR(MR) - 最大可接受接收状态变量

该状态变量等于 $VR(R) + AM_Window_Size$ ，且保持超出接收窗口的第一个 AMD PDU 的 SN 值，并作为接收窗口的上边界。

——VR(X) - t-Reordering 状态变量

此变量保持的 SN 值，紧跟在触发了 t-Reordering 的 RLC 数据 PDU 的 SN 之后。

——VR(MS) - 最大 STATUS 发送状态变量

该状态变量保持当一个 STATUS PDU 需要被构造时，能被“ACK_SN”指示的 SN 的最高可能值。其初始置为 0。

——VR(H) - 最高收到状态变量

该状态变量保持的 SN 值，紧跟于在收到的 RLC 数据 PDU 中，SN 最高的 RLC 数据 PDU 的 SN 值之后。其初始值置为 0。

每个发送 UM RLC 实体应维护如下状态变量：

——VT(US)

该状态变量保持给下一个新产生的 UMD PDU 的 SN 值。其初始置为 0，且当任何时候该 UM RLC 实体分发一个 $SN=VT(US)$ 的 UMD PDU 时，被更新。

每一个接收 UM RLC 实体应维护如下状态变量：

——VR(UR) - UM 接收状态变量

该状态变量保持最早的、还被视为要重排序的 UMD PDU 的 SN 值。其初始置为 0。

——VR(UX) - UM t-Reordering 状态变量

该状态变量保持的 SN 值，紧跟在触发了 t-Reordering 的 UMD PDU 的 SN 值后。

——VR(UH) - UM 最高收到状态变量

该状态变量保持的 SN 值，紧跟在接收的 UMD PDU 中具有最高 SN 值的 UMD PDU 的 SN 值后，且作为重新排序窗口的上边界。其初始置为 0。

8.4.2 常量

——AM_Window_Size

该常量在每个 AM RLC 实体的发送部分和接收部分，均使用于从 VT(A)来计算 VT(MS)，并从 VR(R)来计算 VR(MR)。AM_Window_Size = 256。

——UM_Window_Size

该常量在接收 UM RLC 实体，用于定义那些能够被接收且不导致接收窗口前进的 UMD PDU 的 SN。当配置 5 比特的 SN 时，UM_Window_Size = 16；当配置 10 比特的 SN 时，UM_Window_Size = 256。

8.4.3 定时器

如下定时器由 RRC 配置。

——t_PollRetransmit

AM RLC 实体的发送部分使用该定时器，以重传一个轮询（见 8.2.2.2 节）。

——t_Reordering

AM RLC 实体的接收部分和接收 UM RLC 实体使用该定时器，以检测 RLC PDU 在下层的丢失（见 8.1.1.2 和 8.1.2.2 节）。若 t_Reordering 运行，则 t_Reordering 不应被额外启动，即在一个给定的时间，每个 RLC 实体只有一个 t_Reordering 运行。

——t_StatusProhibit

AM RLC 实体的接收部分使用该定时器，以禁止 STATUS PDU 的发送（见 8.2.2.3 节）。

8.4.4 可配参数

如下参数由 RRC 配置。

——maxRetxThreshold

每个 AM RLC 实体的发送部分使用该参数，以限定一个 AMD PDU 的重传次数（见 8.2.2.1 节）。

——pollPDU

每个 AM RLC 实体的发送部分使用此参数，以触发每个 pollPDU PDUs 的一个轮询（见 8.2.2.2 节）。

——pollByte

每个 AM RLC 实体的发送部分使用此参数，以触发每个 pollByte 字节的一个轮询（见 8.2.2.2 节）。

9 MAC

9.1 概述

9.1.1 介绍

本章目的是从功能角度描述 MAC 架构及 MAC 实体。

9.1.2 MAC 架构

本节只给出模型，并不对具体操作做出指定或者约束。
RRC 对 MAC 的配置进行控制。

9.1.2.1 MAC 实体

MAC实体处理以下传输信道：

- 广播信道（BCH）；
- 数据传输信道（TCH）；
- 随机接入信道（RACH）；

图 30描绘了MAC实体的一种可能结构，不限制具体实现方式。

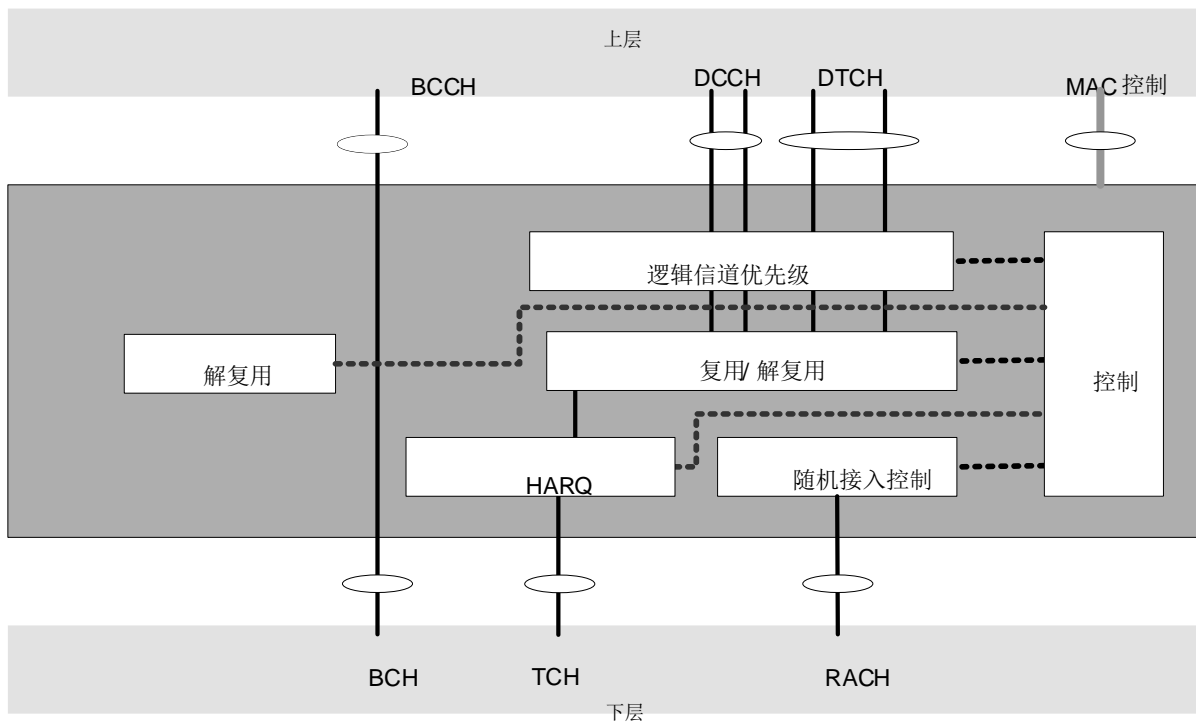


图 30 MAC 结构

9.1.3 服务

9.1.3.1 为上层提供的服务

本节描述MAC子层提供给上层的服务：

- 数据传输；

9.1.3.2 期待从物理层得到的服务

物理层提供如下服务给MAC：

- 数据传输业务；
- HARQ 反馈指示；
- 信道质量测量（CQI、RI）。

通过使用传输信道提供数据传输服务。传输信道的特性通过其传输格式（或格式集）来定义，传输格式指示了正在讨论中的用于传输信道的物理层过程，如信道编码/交织/和针对具体业务的速率匹配等。

9.1.4 MAC 的功能

MAC 子层支持如下功能：

- 逻辑信道和传输信道间的映射；
- 将来自一个或不同逻辑信道上的 MAC SDUs 复用到传输块（TB），并通过传输信道递交到物理层；
- 将来自物理层在传输信道承载的 TB 块解复用为一条或者不同逻辑信道上的 MAC SDUs；
- 利用 HARQ 纠错；
- 通过动态调度实现不同节点间的优先级处理；
- 同一节点的不同逻辑信道之间的优先级处理；
- 逻辑信道优先级的划分；
- 传输格式的选择。
- 随机接入。

9.1.5 信道结构

MAC子层在下面定义的信道上进行操作。传输信道是MAC和层一间的服务接入点，逻辑信道是MAC和RLC间的服务接入点。

9.1.5.1 传输信道

MAC使用的传输信道见表 36。

表 36 MAC 使用的传输信道

传输信道名称	缩写
Transport Channel	TCH
Random Access Channel	RACH
Broadcast Channel	BCH

9.1.5.2 逻辑信道

MAC层在逻辑信道上提供数据传输业务。根据MAC提供的不同数据传输服务，定义了一组逻辑信道类型。

逻辑信道类型通过所传信息类型来定义。

MAC提供的控制和业务信道见表 37。

表 37 MAC 使用的逻辑信道

逻辑信道名称	缩写	控制信道	业务信道
Broadcast Control Channel	BCCH	X	
Dedicated Control Channel	DCCH	X	
Dedicated Traffic Channel	DTCH		X

9.1.5.3 传输信道到逻辑信道的映射

逻辑信道到传输信道的映射取决于RRC层配置的复用关系。

9.1.5.3.1 信道映射

MAC实体负责逻辑信道到传输信道 传输信道到逻辑信道的映射。信道的映射如图 31所示以及下表 38所示。

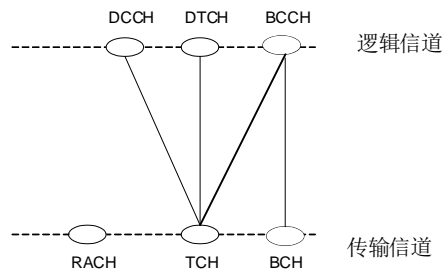


图 31 信道映射

表 38 信道映射 (channel mapping)

逻辑信道	传输信道		
	TCH	RACH	BCH
DCCH	X		
DTCH	X		
BCCH	X		X

9.1.5.4 传输信道到物理信道的映射

传输信道到物理信道的映射如图 32所示:

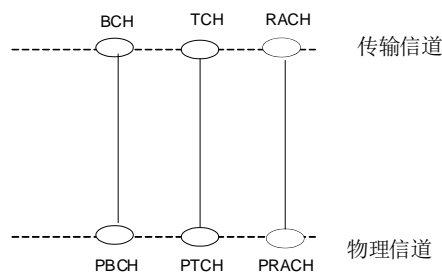


图 32 传输信道到物理信道的映射

9.2 MAC 过程

9.2.1 随机接入过程

9.2.1.1 随机接入过程初始化

本节描述的随机接入过程是由RRC来触发的。

在触发随机接入过程之前，假设以下信息已知：

- 1)随机接入前导码组及每组中可用的随机接入前导码集：前导码组1-64。
- 1)随机接入响应窗ra-ResponseWindowSize = 1（被接入节点占用的非相邻的下一个Radio Frame子帧
2)
- 1)功率抬升因子powerRampingStep;
- 1)前导码最大传输次数preambleTransMax;
- 1)前导码初始发射功率preambleInitialReceivedTargetPower。

注：在每次随机接入过程触发前，上述参数可通过高层配置进行更新。

随机接入过程应按如下步骤执行：

1)开始选择随机接入资源（见9.2.1.2节）

9.2.1.2 随机接入资源选择

随机接入资源选择过程应按如下步骤执行：

1)在随机接入前导码组中随机选择一个随机接入前导码ra-PreambleIndex。随机函数应满足每个前导码被选中的概率相等；

1)设置PRACH掩码索引ra-PRACH-MaskIndex为0。

1)根据被接入节点的资源占用情况，在被接入节点固定占用Radio Frame的子帧2和3。

1)PRACH发起随机接入。

1)执行随机接入前导码传输过程（见9.2.1.3节）。

9.2.1.3 随机接入前导码传输

随机接入前导码传输过程应按如下步骤执行：

1)设置PREAMBLE_RECEIVED_TARGET_POWER为preambleInitialReceivedTargetPower + DELTA_PREAMBLE + (PREAMBLE_TRANSMISSION_COUNTER - 1) * powerRampingStep；

1)指示物理层使用已选择的PRACH，相应的RA-RNTI，前导码索引以及 PREAMBLE_RECEIVED_TARGET_POWER发送随机接入前导码。

9.2.1.4 随机接入响应接收

一旦随机接入前导码被传输，为接收到随机接入响应，节点应在随机接入响应窗中监测以RA-RNTI（见下面定义）为标识的PDCCH。随机接入响应窗始于包含前导码发送结束的子帧，结束于被接入节点下一个不相邻Radio Frame的子帧2。与发送随机接入前导码使用的PRACH资源相关的RA-RNTI通过下式获得：

$$RA-RNTI = 1 + ((t_id * 4) + preamble_idx) \% 60 \quad (1)$$

式中：

t_id为MSG1发送的子帧号，取值范围为(0 ≤ t_id < 5)；

preamble_idx为随机接入过程中随机选择的接入前导码，取值范围(1 ≤ preamble_idx ≤ 64a)。

当节点成功接收一个包含与已发送随机接入前导码相匹配的随机接入前导码标识的随机接入响应后可以停止监听随机接入响应。

1)如果节点在该TTI收到以RA-RNTI加扰的PCCH指示，且接收到的TB被成功解码，

2)如果随机接入响应包含一个与已传输随机接入前导码（参见 9.2.1.3 节）相同的随机接入前导码标识，则节点应：

3)认为此次随机接入响应接收成功；

3)向低层指示 preambleInitialReceivedTargetPower 和应用于最近一次前导码发送过程的功率抬升总量（即 (PREAMBLE_TRANSMISSION_COUNTER wer 标识的 powerRampingStep)）；

3)随机接入前导码由节点 MAC 选择：

4)将随机接入响应中携带的值设置为临时 C-RNTI;

如果在随机接入响应窗内没有收到随机接入响应,或者所有收到的随机接入响应中都不包含与已传输的随机接入前导码相匹配的随机接入前导码标识,则认为随机接入响应接收不成功,节点应执行如下操作:

1)将PREAMBLE_TRANSMISSION_COUNTER加1;

1)如果PREAMBLE_TRANSMISSION_COUNTER = preambleTransMax + 1, 则:

2)向高层指示随机接入问题。

1)否则:

2)立刻寻找下一个可用于接入的子帧(见9.2.1.2节),重新发起接入

1)执行随机接入资源选择过程(见9.2.1.2节)。

9.2.1.5 随机接入过程完成

当随机接入过程成功完成时,节点应:

1)如果存在,丢弃显示指示的ra-PreambleIndex及ra-PRACH-MaskIndex;

9.2.2 BCCH 数据传输

当节点需要读取BCCH:

1)如果在该TTI接收到以SI-RNTI加扰的下行分配,则:

2)未定义冗余版本, SFN 在 MIB 信息中携带。

2)广播没有 HARQ 进程。

9.2.3 TCH 数据传输

9.2.3.1 HARQ 操作

9.2.3.1.1 HARQ 实体

节点中存在一个 HARQ 实体,其维护一定数量并行的 HARQ 进程,从而实现在等待上一次传输成功与否的反馈期间继续进行连续的传输。

并行 HARQ 进程每个 CC、每个码字对应的数目为 16 个。

在一个给定的 TTI,如果该 TTI 节点可以发送数据,HARQ 实体为将要发生的传输确定一个 HARQ 进程。HARQ 实体还把从物理层接收到的 HARQ 反馈(ACK/NACK)发送给相应的 HARQ 进程。

对每个 TTI, HARQ 实体应:

1)指定该TTI对应的HARQ进程;

2)如果 NDI 标志翻转或者是初始传输

3)将 MAC PDU, 以及 HARQ 信息传递给指定的 HARQ 进程;

3)通知相应的 HARQ 进程触发一次新的传输

2)否则

3)将上行授权和 HARQ 信息(冗余版本)传递给指定的 HARQ 进程;

3)通知相应的 HARQ 进程触发一次重传。

9.2.3.1.2 HARQ 进程

每个 HARQ 进程均对应一个 HARQ 缓存。

每个 HARQ 进程需要维护状态变量 CURRENT_TX_NB, 用来指示当前缓存中 MAC PDU 已经传输的次数; 以及状态变量 HARQ_FEEDBACK, 用来指示当前缓存中 MAC PDU 的 HARQ 反馈。CURRENT_TX_NB 在每个 HARQ 进程初始化的时候被置 0。

冗余版本固定为 0。

新数据传输使用根据物理层上报的 CQI、带宽和实际数据量计算出的 MCS。重传使用上一次传输尝试的传输资源以及 MCS。

HARQ 传输的最大重传次数用户可配置, 可以配置 0-3 次, 其中 0 代表不做 HARQ 重传。

当收到该传输块的 HARQ 反馈时, HARQ 进程应:

1) 设置 HARQ_FEEDBACK 为收到的值。

如果 HARQ 实体请求一次新数据传输, HARQ 进程应:

1) 设置 CURRENT_TX_NB 为 0;

1) 将 MAC PDU 存储在相应的 HARQ 缓存中;

1) 设置 HARQ_FEEDBACK 为 NACK;

1) 按照下面描述进行传输。

如果 HARQ 实体请求一次重传, HARQ 进程应:

1) 如果 HARQ_FEEDBACK = NACK:

2) 按照下面的描述进行传输。

为进行一次传输, HARQ 进程应:

1) 通知物理层按照计算出的 MCS 以及与 CURRENT_IRV 相应的冗余版本进行一次传输; 完成上述动作之后, HARQ 进程应:

1) 如果 CURRENT_TX_NB = 最大传输次数 - 1:

2) 清空 HARQ 缓存。

9.2.3.2 逻辑信道优先级

当有新数据传输时, 节点会进行逻辑信道的优先级处理。

RRC 通过为每个逻辑信道指示相关优先级处理参数来进行上行数据的调度: priority, 其数值越高, 优先级越低;

优先级高的数据优先进行传输。

9.2.4 BCH 接收

当需要接收 BCH 时, 节点应:

1) 接收并尝试对 BCH 解码;

1) 如果 BCH 上的 TB 被成功解码:

2) 将成功解码的 MAC PDU 递交高层。

9.2.5 MAC reset

如果 RLC 发起 reset 流程, MAC 应:

1) 重置所有的 HARQ 进程

9.2.6 对于未知, 预料以外以及错误数据的处理

对于 MAC 实体通过以 C-RNTI、或已配置的分配信息收到的 MAC PDU，如果包含预留或者无效值，MAC 实体应：

1) 丢弃该 MAC PDU。

9.3 协议数据单元，格式和参数

9.3.1 协议数据单元

9.3.1.1 概述

MAC PDU 是在长度上以字节对齐的比特串（即 8bit 的倍数）。在 9.3 节的图中，比特串以表格的形式表示，其中最高有效位位于表格第一行的最左侧，最低有效位位于表格最后一行的最右侧，通常，比特串从左到右逐行读取。MAC PDU 中每个参数域的比特顺序都是将第一个即最高有效位放在最左边，而将最后一个即最低有效位放在最右边。

MAC SDU 是在长度上以字节对齐的比特串（即 8bit 的倍数）。SDU 包含在 MAC PDU 中，从第一个比特开始。

节点应忽略接收 MAC PDUs 中的预留位的值。

9.3.1.2 MAC PDU（TCH，除了随机接入响应）

一个 MAC PDU 包含一个 MAC 头，0 或多个 MAC 服务数据单元 (MAC SDU)，以及可能的填充，如图 33 所示。

MAC 头和 MAC SDUs 的长度可变。

一个 MAC PDU 头包含一个或多个 MAC PDU 子头；每个子头对应一个 MAC SDU 或填充。

除 MAC PDU 内最后一个子头以及固定长度 MAC 控制单元的子头之外，MAC PDU 子头包含六个头字段 R/R/E/LCID/F/L。MAC PDU 内最后一个子头以及固定长度的 MAC 控制单元子头包含四个头字段 R/R/E/LCID。填充对应的 MAC PDU 子头也包含四个头字段 R/R/E/LCID，如图 34 所示。

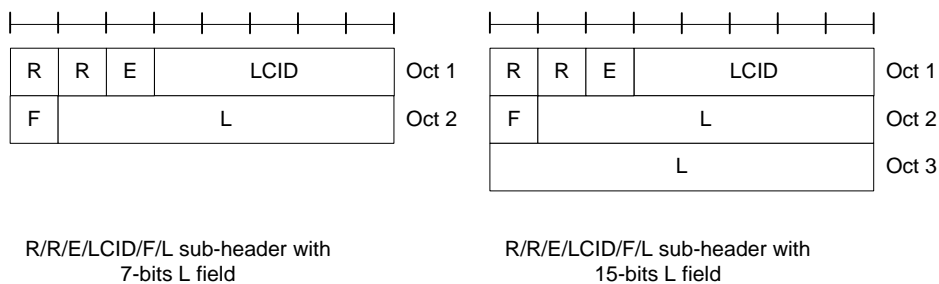


图 33 R/R/E/LCID/F/L MAC 子头

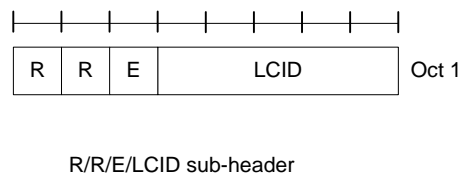


图 34 R/R/E/LCID MAC 子头

MAC PDU 子头的顺序与对应的 MAC SDUs、MAC 控制单元以及填充的顺序一致。

MAC 控制单元位于所有 MAC SDU 的前面。

填充位于 MAC PDU 的末尾处，除需要添加一字节或两字节填充的情况外。填充可以是任意值，节点忽略之。当填充在 MAC PDU 末尾时，允许 0 个或多个填充字节。

当需要添加 1 字节或 2 字节填充时,与该填充对应的 1 个或 2 个子头被置放在 MAC PDU 最起始的位置, 其它子头之前。

每个节点每个 TB 块最多可传输一个 MAC PDU。每个 TTI 最多可传输一个 MCH MAC PDU, 如图 35 所示。

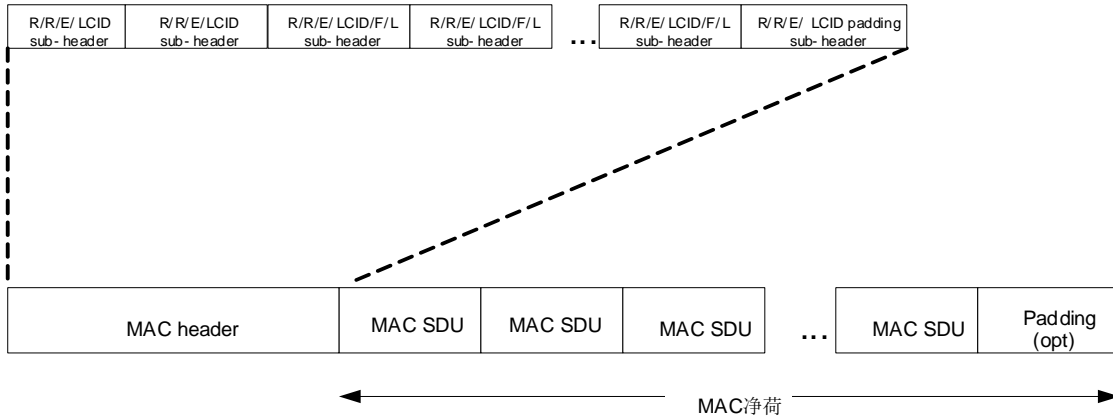


图 35 包括 MAC 头, MAC SDU 以及填充的 MAC PDU 格式

9.3.1.3 MAC PDU (随机接入响应)

MAC PDU 由一个 MAC 头、一个 MAC 随机接入响应 (MAC RAR)以及可能的填充组成, 如图 38 所示。

MAC 头长度固定 7 字节。

MAC PDU 头由一个或 MAC PDU 子头组成; 对应于一个 MAC RAR。

MAC PDU 子头由三个头部域 E/T/RAPID 组成(如图 36 所示)。

MAC RAR 由 4 个字段组成: R/Timing Advance Command/Send Power/Temporary C-RNTI , 如图 37 所示。

在最后一个 MAC RAR 后可能出现填充。 填充的出现与否以及其长度基于 TB 块的大小, 即 MAC 头和 MAC RAR 数量的大小, 如图 38 所示。

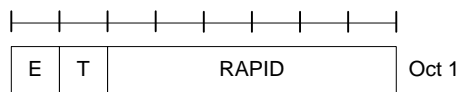


图 36 E/T/RAPID MAC 子头

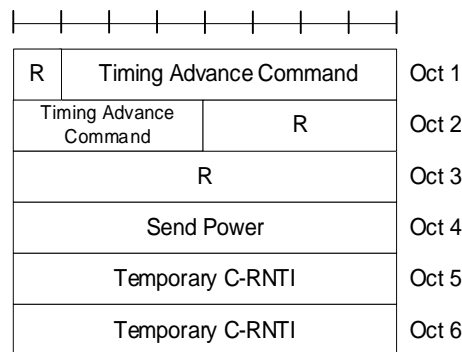


图 37 MAC 随接入响应

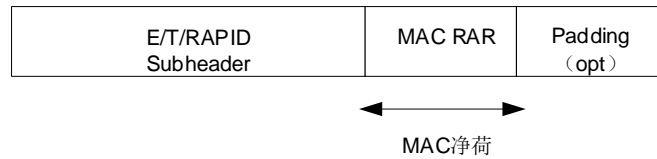


图 38 包括 MAC 头以及 RAR 的 MAC PDU 格式

9.3.2 格式和参数

9.3.2.1 TCH (除 RAR)

MAC 头大小可变，由下列域组成：

——**LCID**: 逻辑信道 ID 域，标识每个 MAC SDU 对应的逻辑信道、MAC 控制单元的类型或填充，TCH 的 LCID 分别参见表 39。MAC PDU 中每个 MAC SDU 或者填充对应一个 LCID 域。除此之外，当需要一字节或二字节的填充但不能添加在 MAC PDU 尾部时，MAC PDU 中会包含一或二个附加的 LCID 域。LCID 域长度为 5 比特；

——**L**: 长度域，指示对应 MAC SDU 或可变长度 MAC 控制单元的字节数。除最后一个子头以及固定长度 MAC 控制单元对应的子头外，MAC PDU 子头中有一个 L 域。L 域的大小由 F 域指示；

——**F**: 格式域，指示长度域的长度，如表 27 所示。除最后一个子头以及固定长度 MAC 控制单元对应的子头外，每个 MAC PDU 子头都包含一个 F 域，F 域长度为 1bit。如果 MAC SDU 或可变长度 MAC 控制单元的长度小于 128 字节，应将 F 域置为 0，否则置为 1，如表 40；

——**E**: 扩展域，扩展域是一个标志位，指示 MAC 头中是否还有其他的域。如果 E 域设定为“设定”，则表示其后至少有另一组的 R/R/E/LCID 域。如果 E 域设置为“设置”，表示从其后的字节起为 MAC SDU 或者 MAC 控制单元或者填充。

——**R**: 预留比特，设置为置为比。

MAC 头和子头都是字节对齐。

表 39 TCH 的 LCID 值

索引值	LCID 值
00000	Reserved
00001-01010	Identity of the logical channel
01011-11110	Reserved
11111	Padding

表 40 F 域的值

索引值	L 域大小(比特)
0	7
1	15

9.3.2.2 随机接入响应的 MAC 头

MAC 头大小固定 1 字节，由下列域组成：

- E: 扩展域, 扩展域是一个标志位, 指示 MAC 头中是否还有其他的域。E 域设置为“设置, 表示从其后的字节起为 MAC RAR 或者填充;
 - T: 类型域, 类型域是一个标志位, T 域为“为型, 指示子头中存在一个随机接入前导码 ID (RAPID) 域;
 - RAPID: 随机接入前导码标识域指明了已发送的随机接入前导码 (参见 9.3.1.3 节)。RAPID 域的长度为 6bits。
- MAC 头和子头都是字节对齐。

9.3.2.3 随机接入响应 MAC 净荷

MAC RAR 长度固定, 包含下列域:

- R: 预留比特, 设置为 留比特;
- Timing Advance Command: 时间提前命令域, 指示用于节点时间调整量的索引值, T_A (0, 1, 2, 指示用于节点时间最远支持 110km)。时间提前命令域的长度是 11bits;
- Send Power: MSG2 的发送功率, 长度是 8bits;
- Temporary C-RNTI: 临时 C-RNTI 域, 指示节点在随机接入过程中使用的临时 C-RNTI。临时 C-RNTI 域的长度是 16 bits。

MAC RAR 字节对齐。

9.4 变量和常量

9.4.1 RNTI 值

RNTI 值如表 41 所示。

表 41 RNTI 值

值 (hexa-decimal)	RNTI
0x1-0x3C	RA-RNTI
0xF000-0xFF80	C-RNTI
0xFFFFD	广播 RNTI
0xFFFFF	SI-RNTI

注: C-RNTI 的第 11 到第 7 位 (0x0F80) 为对应的节点 id 号, 取值范围 0~31。

RNTI 用途如表 42 所示。

表 42 RNTI 用途

RNTI	用途	传输信道	逻辑信道
SI-RNTI	Broadcast of System Information	BCH	BCCH
Temporary C-RNTI	Msg3 transmission	TCH	DCCH, DTCH
C-RNTI	Dynamically scheduled unicast transmission	TCH	DCCH, DTCH
RA-RNTI	Random Access Response	RACH	N/A

参 考 文 献

- [1] ETSI TC SAGE Specification: "Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 1: UEA2 and UIA2 specifications" v2.1.
- [2] ETSI TC SAGE "Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3; Document 1: 128-EEA3 & 128-EIA3 Specification" version 1.7.
- [3] NIST: "Advanced Encryption Standard (AES) (FIPS PUB 197) "
- [4] NIST Special Publication 800-38A (2001): "Recommendation for Block Cipher Modes of Operation".
- [5] NIST Special Publication 800-38B (2001): "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication".

临时格式

《应急指挥无线宽带自组网系统技术规范

第3部分：数据链路层》

(征求意见稿 送审稿 报批稿)

编制说明

标准编制组

2024年7月

一、工作简况

（一）任务来源

根据《应急管理部办公厅关于印发 2023 年第二批行业标准制修订计划的通知》（应急厅函〔2023〕288 号）要求，应急管理部科技和信息化司组织起草了《应急指挥无线宽带自组网系统技术规范 第 3 部分：数据链路层》，编号为 2023-YJ-06，项目周期 16 个月，由 TC307 应急管理 with 减灾救灾标准化技术委员会组织审查。

（二）制定背景

党的十八大以来，以习近平同志为核心的党中央高度重视应急管理体系和能力建设。根据《“十四五”国家应急体系规划》，强化数字技术在灾害事故应对中的运用，全面提升监测预警和应急处置能力。加强空、天、地、海一体化应急通信网络建设，提高极端条件下应急通信保障能力。

为了落实国家十四五政策要求，解决大震巨灾场景下的多层次多队伍协同救援指挥的难题，快速构建救援现场指挥体系，应急管理部提出了数字化战场建设的相关工作，其中构建一张具有自主知识产权的、移动的、带宽高、传输距离远，组网灵活，性能稳定的无线骨干网络，是数字化战场网络基础的核心。目前国内掌握无线宽带自组网技术的厂商所采用的技术体制、协议都各不相同，因此设备间无法实现空口互通，这与数字化战场对宽带自组网的需求是相悖的。因此，研制应急指挥无线宽带自组网系统技术规范的主要目的是根据当前应急管理部对无线宽带自组网的需求，充分采纳

现有厂商在相关技术的长处，统一无线宽带自组网各层的技术实现，使得采用本标准研制的自组网产品能够通过空口互联互通，满足数字化战场对无线自组网所需的功能要求和性能要求。

（三）起草小组人员组成及所在单位

根据标准编制计划，由应急管理部大数据中心面向社会公开征集标准编制单位，并于 2022 年 8 月成立标准起草工作组，共同开展标准制订工作。

（四）主要起草过程

2022 年 3 月，通过应急管理部网站公开征集参与编制单位。收集 45 家单位有效材料。

2022 年 6 月，通过对申请单位进行逐一审核，组织专家听取了 17 家申请单位的线上汇报，遴选出 7 家参与编制单位并在部门门户网站应急装备之家公示。

2022 年 8 月，召开标准编制工作启动会，成立了标准起草工作组。

2022 年 12 月，完成技术规范初版冻结。

2023 年 3 月，各参研单位依据冻结的技术规范版本完成原型机研制工作并开展互联互通测试。

2023 年 5 月，标准编制组向 TC307 递交了标准立项申请并开展标准草案稿编制工作。

2023 年 6 月，通过立项审查。

2023 年 10 月，完成行标立项。

2023 年 11 月，完成宽带自组网基本功能和性能指标实

实验室测试，根据测试过程对标准内容进行完善。

2023年12月，完成外场模拟测试，根据测试过程对标准内容进一步完善。

2024年5月，参与“应急使命·2024”演习，针对自组网应用情况对标准规范进一步修改完善，形成标准征求意见稿。

二、标准编制原则、主要技术内容及其确定依据

（一）标准编制原则

本文件按照 GB/T 1.1-2020（标准化工作导则 第一部分：标准化文件的结构和起草规则）的规定起草。

本文件编写过程中遵循全面、科学、合理、可行的原则，做到既先进又切实可行。

（二）标准主要技术内容及确定依据

《应急指挥无线宽带自组网系统技术规范 第3部分：数据链路层》（征求意见稿）共有9章，主要包含范围、规范性引用文件、术语和定义、缩略语、概述、RRC、PDCP、RLC，以及MAC相关内容。

1. 范围

本文件规定了应急指挥无线宽带自组网系统技术规范的数据链路层的协议，包括：无线资源控制协议（RRC）、分组数据汇聚协议（PDCP）、无线链路控制协议（RLC）、媒体接入控制协议（MAC）。

本文件适用于基于LTE TDD实现的应急指挥无线宽带自组网系统。

2. 规范性引用文件

本文件没有引用其它国家标准或行业标准。

3. 术语和定义

本文件定义适用本标准的术语。

4. 缩略语

该章节主要定义使用本标准的缩略语。

5. 概述

该章节说明了数据链路层的总体架构和主要功能。

6. RRC

该章节主要定义 **RRC** 协议模型的架构、服务、功能、过程、数据格式、定时器和默认配置。

7. PDCP

该章节主要定义 **PDCP** 协议模型的架构、业务、功能、过程、数据格式、定时器和常量。

8. RLC

该章节主要定义 **RLC** 协议模型包括结构、服务、功能、过程、数据格式、定时器、常量和变量。

9. MAC

该章节主要定义 **MAC** 协议模型包括架构、服务、功能、信道结构、过程、数据格式、常量和变量。

(三) 标准修订变化及依据 (仅修订标准需要列出)

无。

三、试验验证的分析、综述报告、技术经济论证, 预期的经济效益、社会效益和生态效益。

无。

四、与国际、国外同类标准技术内容的对比情况

国际、国外可公开的标准和技术主要有 IEEE 802.11 系列标准、802.15、以及 MANET 等；802.11 标准的 AD Hoc 模式主要用于高速率、短距离通信，且移动速度较慢；802.15 则适用于低功耗、短距离网络；802.11p 专为车载 Ad hoc 网络（VANET）设计，相对传输速率较低；MANET 则未定义物理层，主要关注链路层和网络层协议；3GPP 的 D2D 技术则关注点对点的通信，并无路由能力。

编制组在充分调研了上述各个标准后，决定采用 LTE 的技术体制，结合 MANET 的路由技术，设计了一套全新的无线宽带自组网。

五、以国际标准为基础的起草情况、是否合规引用或采用国际国外标准以及未采用国际标准的原因

无。

六、与有关法律、行政法规及相关标准水平的关系

（一）与有关法律、行政法规、标准关系

现有的宽带自组网设备厂家繁多，存在着技术体制多，协议不同，无法互联互通，没有统一标准的现状；因此本文件与现行法律法规和政策及其它强制性标准没有矛盾。

（二）配套推荐性标准的制定情况（强制性标准应填写）

无。

七、重大分歧意见的处理过程及依据

异厂商互联问题。最初在需求层面，各厂商对设备使用

情况的考虑是差异巨大的，主要存在两种观点，一是异厂商设备在实战中的互联需求是可以使用除空口互联方式的技术实现的。二是，真正的实战中必定会存在空口互通的情况，且使用空口互通外的其他方式会存在中心节点失效问题。经过各参研单位的谨慎考虑，最终决定了使用空口互通来作为基本需求。一切技术提案以此为基础。

八、作为强制性标准或推荐性标准的建议及理由

本标准适用于应急指挥通信宽带自组网设备，是厂商对应急宽带自组网设备的设计、开发基准。是各级应急管理部门招标、检测与验收的依据，应为推荐性标准。

九、标准自发布日期至实施日期的过渡期建议及理由

本标准 of 推荐性行业标准，无需设置过渡期。

十、与实施标准有关的政策措施

由于本标准是推荐性行业标准，因此为使标准能够尽快得到落实和应用，建议标准起草单位在行业主管部门的领导下，能够出台配套的相关政策并对标准内容进行宣传贯彻工作，加快推动标准实施。

十一、是否需要对外通报的建议及理由。

无。

十二、废止现行有关标准的建议

无。

十三、涉及专利的有关说明

无。

十四、标准所涉及的产品、过程或者服务目录

无。

十五、其他应予以说明的事项

无。