



中华人民共和国通信行业标准

YD/T 2856.2-2015

2GHz TD-SCDMA 数字蜂窝移动通信网 多载波高速分组接入 Uu 接口层 2 技术要求 第 2 部分：RLC 协议

2GHz TD-SCDMA digital cellular mobile telecommunication network
Multi-carrier HSPA Uu interface layer 2 technical requirements
Part 2: RLC protocol

(3GPP TS 25.322 V10.1.0, Radio Link Control (RLC) protocol
specification, NEQ)

2015-07-14 发布

2015-10-01 实施

中华人民共和国工业和信息化部 发布

目 次

前 言	III
1 范围	1
2 规范性引用文件	1
3 缩略语	1
4 概述	3
4.1 内容	3
4.2 RLC子层结构概述	3
5 功能	10
6 提供给上层的业务	10
6.1 业务/功能在逻辑信道上的映射	11
7 期望从MAC获得的业务	13
8 层与层通信的单元	13
8.1 RLC和上层之间的原语	13
8.2 原语参数	15
9 对等层通信元素	16
9.1 协议数据单元	16
9.2 格式和参数	17
9.3 协议状态	32
9.4 状态变量	36
9.5 定时器	38
9.6 协议参数	40
9.7 指定功能	41
10 未知、未预见和错误协议数据的处理	50
10.1 错误序列号	50
10.2 不一致的状态指示	50
10.3 无效的PDU格式	50
10.4 CRC错误的RLC PDU	50
11 基本过程	51
11.1 透明模式数据（TMD）传送过程	51
11.2 非确认模式数据（UMD）传送过程	52
11.3 确认模式数据（AMD）传送过程	56
11.4 RLC复位过程	60

11.5 STATUS PDU传送过程.....	63
11.6 显式信令的SDU丢弃过程.....	64
11.7 流量控制.....	68
参考文献.....	75

前 言

YD/T 2856《2GHz TD-SCDMA数字蜂窝移动通信网 多载波高速分组接入 Uu接口层2技术要求》是2GHz TD-SCDMA数字蜂窝移动通信网多载波高速分组接入Uu接口系列标准之一，该系列标准的结构和名称如下：

- a) 《2GHz TD-SCDMA 数字蜂窝移动通信网 多载波高速分组接入 Uu 接口物理层技术要求》；
- b) YD/T 2856《2GHz TD-SCDMA 数字蜂窝移动通信网 多载波高速分组接入 Uu 接口层 2 技术要求》；
- c) YD/T 2857《2GHz TD-SCDMA 数字蜂窝移动通信网 多载波高速分组接入 Uu 接口 RRC 层技术要求》。

随着技术的发展，还将制定后续的相关标准。

YD/T 2856《2GHz TD-SCDMA数字蜂窝移动通信网 多载波高速分组接入 Uu接口层2技术要求》分为三个部分：

- a) 第1部分：MAC 协议；
- b) 第2部分：RLC 协议；
- c) 第3部分：PDCP 协议。

本部分是YD/T 2856《2GHz TD-SCDMA数字蜂窝移动通信网 多载波高速分组接入 Uu接口层2技术要求》的第2部分。

本部分按照GB/T 1.1-2009给出的规则起草。

本部分对应于3GPP TS 25.322《RLC协议》（版本 v10.1.0），一致性程度为非等效，主要差异为删除了FDD相关的内容。

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这些专利的责任。

本部分由中国通信标准化协会提出并归口。

本部分起草单位：工业和信息化部电信研究院、大唐电信科技产业集团、中兴通讯股份有限公司、鼎桥通信技术有限公司、中国普天信息产业股份有限公司、新邮通信设备有限公司、重庆重邮信科通信技术有限公司、北京展讯高科通信技术有限公司。

本部分主要起草人：常永宏、许芳丽、严 杲、徐 菲、宋爱慧、黄 河、马志锋、王浩然、贺 刚、王 梅、赵训威、师延山、段红光。

2GHz TD-SCDMA数字蜂窝移动通信网

多载波高速分组接入 Uu接口层2技术要求

第2部分：RLC协议

1 范围

本部分规定了2GHz TD-SCDMA数字蜂窝移动通信网空中接口层2的RLC协议的功能，RLC提供给上层的业务和期望从下层获得的业务，以及层与层之间通信的原语，确立了对等层通信的元素结构和参数的具体要求，给出了RLC协议的基本过程和对未知、未预见和错误事件的处理方法。

本部分适用于2GHz TD-SCDMA数字蜂窝移动通信网中具有多载波高速分组接入功能的空中接口RLC协议。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

3GPP TS 33.102 3G安全、安全体系结构（3G security, security architecture）

3 缩略语

下列缩略语适用于本文件。

ARQ	Automatic Repeat Request	自动重复请求
AM	Acknowledged Mode	确认模式
AMD	Acknowledged Mode Data	确认模式数据
BCCH	Broadcast Control Channel	广播控制信道
BCH	Broadcast Channel	广播信道
C-	Control-	控制-
CC	Call Control	呼叫控制
CCCH	Common Control Channel	公共控制信道
CCH	Control Channel	控制信道
CCTrCH	Coded Composite Transport Channel	编码合成传送信道
CN	Core Network	核心网
CRC	Cyclic Redundancy Check	循环冗余校验
CTCH	Common Traffic Channel	公共业务信道
DC	Dedicated Control (SAP)	专用控制 (SAP)
DCCH	Dedicated Control Channel	专用控制信道
DCH	Dedicated Channel	专用信道
DL	Downlink	下行链路
DTCH	Dedicated Traffic Channel	专用业务信道
FACH	Forward Link Access Channel	前向链路接入信道
FCS	Frame Check Sequence	帧校验序列

FDD	Frequency Division Duplex	频分双工
GC	General Control (SAP)	一般控制 (SAP)
HO	Handove	切换
ITU	International Telecommunication Union	国际电信联盟
Kbps	kilo-bits per second	千比特/秒
L1	Layer 1 (physical layer)	层1 (物理层)
L2	Layer 2 (data link layer)	层2 (数据链路层)
L3	Layer 3 (network layer)	层3 (网络层)
LI	Length Indicator	长度指示器
MAC	Medium Access Control	媒质接入控制
MBMS	Multimedia Broadcast Multicast Service	多媒体广播组播服务
MCCH	MBMS point-to-multipoint Control Channel	MBMS点对多点控制信道
MM	Mobility Management	移动性管理
MRW	Move Receiving Window	移动接收窗口
MS	Mobile Station	移动台
MSCH	MBMS point-to-multipoint Scheduling Channel	MBMS点对多点调度信道
MTCH	MBMS point-to-multipoint Traffic Channel	MBMS点对多点业务信道
MUX	Multiplexing	复用
Nt	Notification (SAP)	通知 (SAP)
PCCH	Paging Control Channel	寻呼控制信道
PCH	Paging Channel	寻呼信道
PDU	Protocol Data Unit	协议数据单元
PHY	Physical layer	物理层
PhyCH	Physical Channels	物理信道
RACH	Random Access Channel	随机接入信道
RLC	Radio Link Control	无线链路控制
RNTI	Radio Network Temporary Identity	无线网络临时标识
RRC	Radio Resource Control	无线资源控制
SAP	Service Access Point	业务接入点
SCCH	Synchronization Control Channel	同步控制信道
SCH	Synchronization Channel	同步信道
SDU	Service Data Unit	业务数据单元
SHCCH	SHared channel Control Channel	共享信道控制信道
SN	Sequence Number	序号
SUFI	Super Field	超字段
TCH	Traffic Channel	业务信道
TDD	Time Division Duplex	时分双工

TFCI	Transport Format Combination Indicator	传输格式组合指示
TFI	Transport Format Indicator	传输格式指示
TTI	Transmission Time Interval	传送时间间隔
TM	Transparent Mode	传输模式
TPC	Transmit Power Control	发射功率控制
U-	User-	用户-
UE	User Equipment	用户设备
UL	Uplink	上行链路
UM	Unacknowledged Mode	非确认模式
UTRAN	Universal Terrestrial Radio Access Network	通用陆地无线接入网

4 概述

4.1 内容

本节描述RLC子层的结构。

4.2 RLC子层结构概述¹⁾

本节中所提供的模型仅为了对RLC子层进行定义，不对协议的实现进行规定或限制。RLC子层由三种RLC实体构成：透明模式（TM）、非确认模式（UM）和确认模式（AM）RLC实体。

4.2.1 RLC子层模型

图1描述了RLC模型中的不同RLC实体。

UM和TM RLC实体可以被配置成一个发送RLC实体或者一个接收RLC实体。发送RLC实体发送RLC PDU，接收RLC实体接收RLC PDU。AM RLC实体由一个发送部分和一个接收部分组成，其中AM RLC实体的发送部分发送RLC PDU；AM RLC实体的接收部分接收RLC PDU。

在“发送端”和“接收端”之间定义了基本过程（见11章）。在UM和TM情况下，发送RLC实体作为发送端，对等的RLC实体作为接收端。根据基本过程的情况，AM RLC实体可以作为发送端或者接收端。发送端是AMD PDU的发送者，接收端是AMD PDU的接收者。发送端和接收端可以位于UE或者UTRAN上。

对于每一个透明模式（TM）和非确认模式（UM）业务有一个发送和一个接收RLC实体。对于确认模式（AM）业务有一个发送和接收合并的实体。

在本文中，在没有特别注明的情况下，“发送”指“传送给下层”。每一个UM和TM RLC实体使用一个逻辑信道发送或者接收数据PDU。AM RLC实体可以配置成使用一个或者两个逻辑信道发送或者接收数据和控制PDU。如果配置成使用两个逻辑信道，那么它们具有相同的类型（DCCH或者DTCH）。在图1中，AM实体间的虚线描述了在不同逻辑信道上发送和接收RLC PDU的可能性，例如：在一个逻辑信道上传输控制PDU，在另一个上传输数据PDU。在4.2.1.1、4.2.1.2和4.2.1.3小节中对不同实体进行了更加详细的描述。

注¹⁾ 为了方便使用者对照阅读及编写者维护后续版本，本部分的章条号与所对应的3GPP标准保持一致。

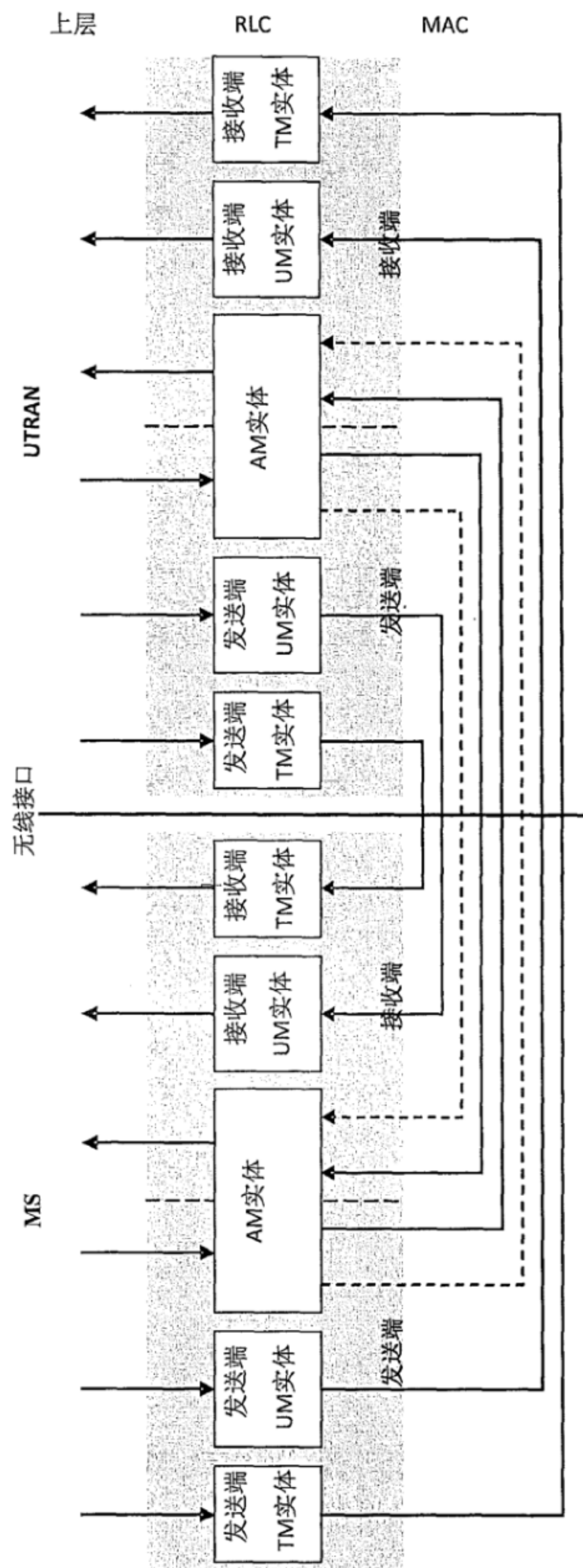


图1 RLC 子层总体模型

4.2.2 透明模式 (TM) RLC 实体

图2表示两个透明模式对等RLC实体的模型。图的下方对与下层进行通信的逻辑信道进行了描述。

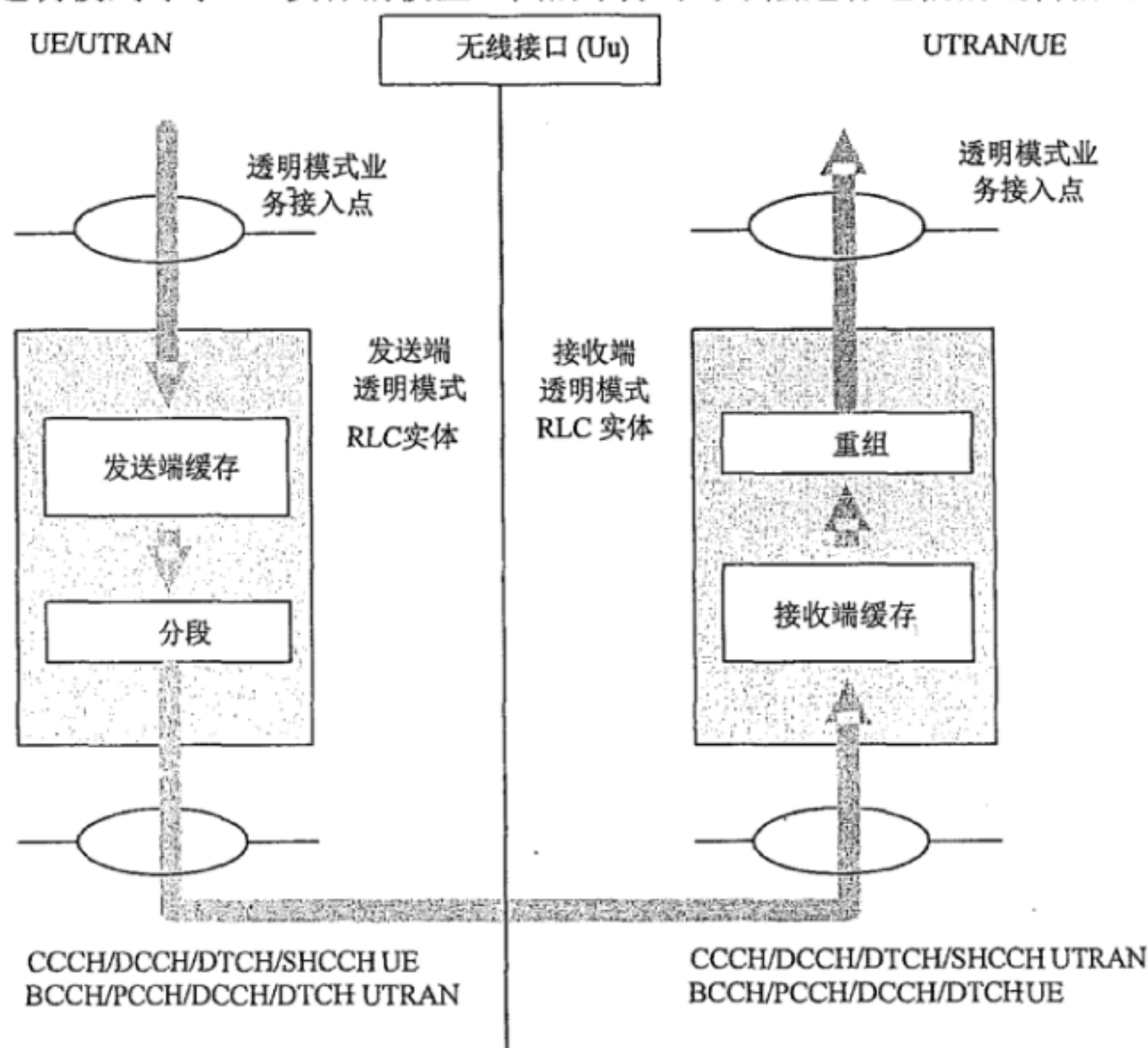


图2 两个透明模式对等实体的模型

4.2.2.1.1 发送 TM RLC 实体

发送 TM-RLC 实体通过 TM-SAP 从上层接收 RLC SDU。

所有接收到的 RLC SDU 的长度应是一个有效 TMD PDU 长度的整数倍。

如果上层进行了分段的配置，并且 RLC SDU 的长度大于下层对 TTI 使用的 TMD PDU 的大小，那么发送 TM RLC 实体对 RLC SDU 进行分段以适应 TMD PDU 的大小，并且不添加 RLC 包头。承载一个 RLC SDU 的所有 TMD PDU 都在同一个 TTI 中发送，该 TTI 中不发送任何其他 RLC SDU 的分段。

如果上层没有进行分段的配置，那么通过将一个 RLC SDU 放入一个 TMD PDU，可以在一个 TTI 中发送多于一个的 RLC SDU。一个 TTI 中的所有 TMD PDU 的长度必须相同。

当对一个 RLC SDU 的处理结束时，得到的一个或者多个 TMD PDU 将通过一个 BCCH、DCCH、PCCH、CCCH、SHCCH 或者一个 DTCH 逻辑信道发送给下层。

4.2.2.1.2 接收 TM RLC 实体

接收 TM-RLC 实体通过配置的逻辑信道从下层接收 TMD PDU。如果上层配置了分段，那么将对在一个 TTI 内接收到的所有 TMD PDU 进行重组得到 RLC SDU。

如果上层没有配置分段，那么每一个 TMD PDU 将被作为一个 RLC SDU。

接收 TM RLC 实体通过 TM-SAP 向上层发送 RLC SDU。

4.2.3 非确认模式 (UM) RLC 实体

图 3 显示了未配置重复避免和重排序功能时两个非确认模式对等 RLC 实体的模型。如图 3 所示的不同功能适用于 6.1 节描述的不同信道类型。

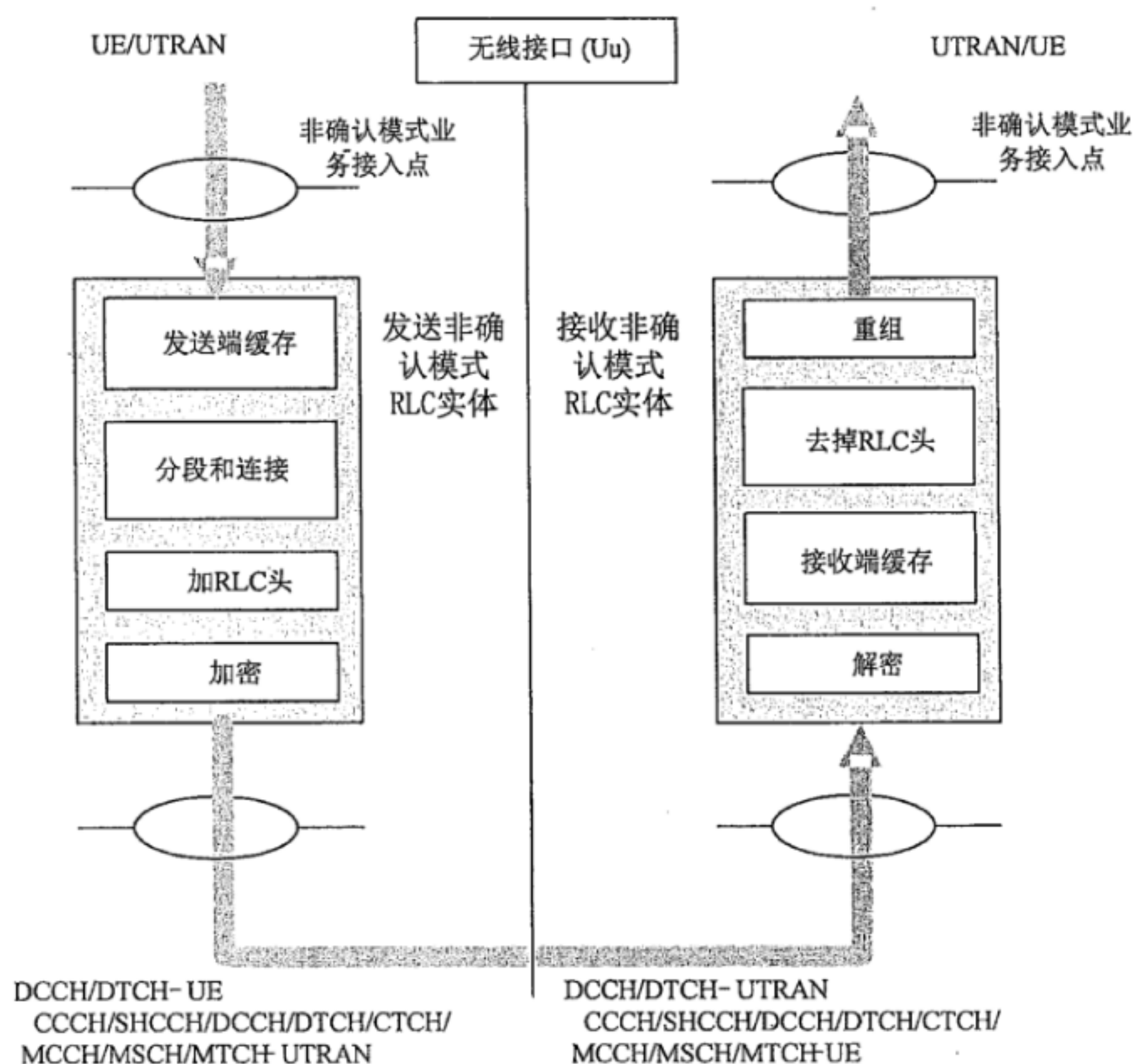


图3 未配置重复避免和重排序功能时两个非确认模式对等实体的模型

图4显示了配置重复避免和重排序功能时两个非确认模式对等RLC实体的模型。由于重复避免和重排序功能仅用于本Release的MCCH/CCCH，所以加密/解密功能被删除。

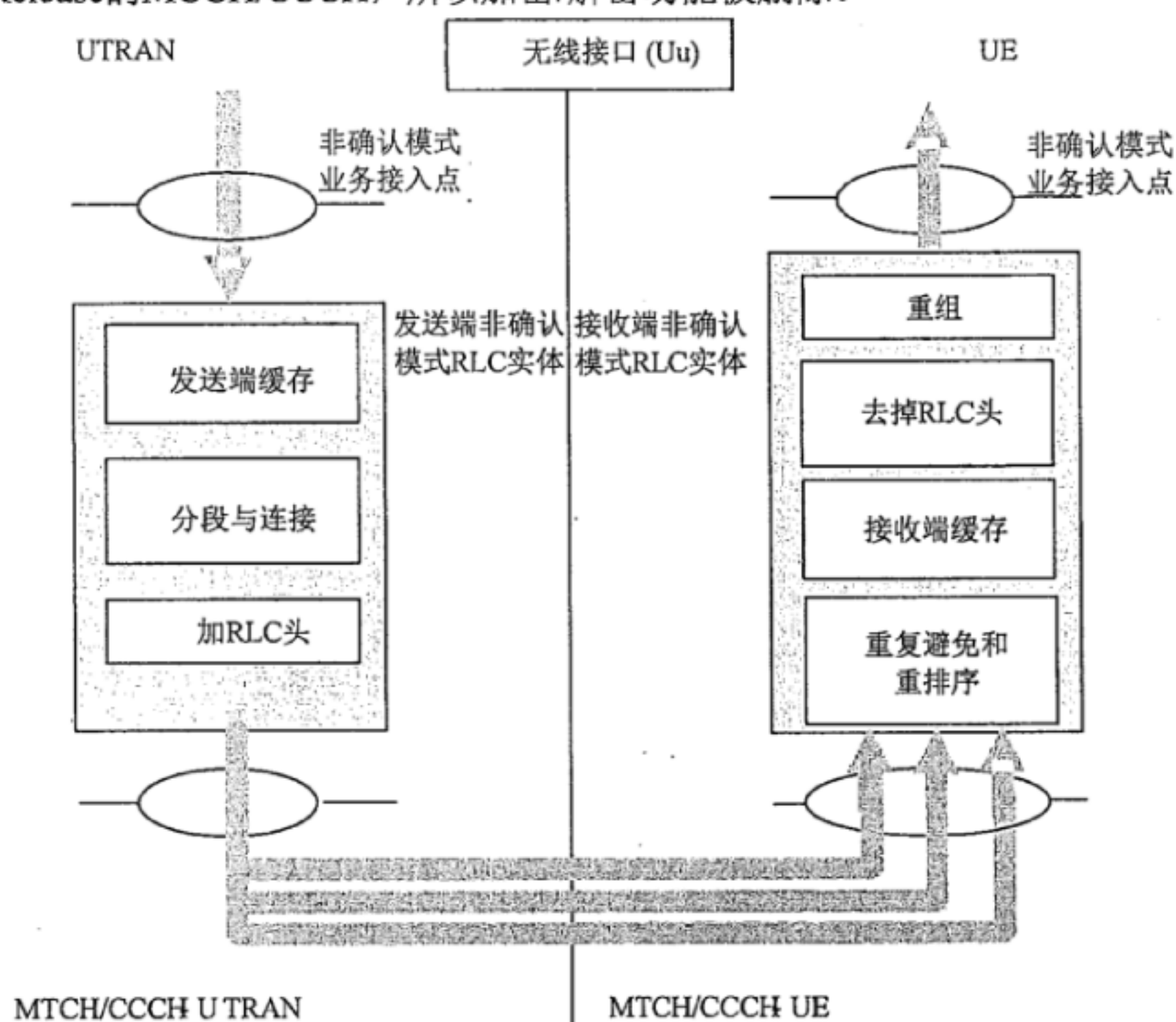


图4 配置了重复避免和重排序功能时两个非确认模式对等实体的模型

4.2.3.1.1 发送 UM RLC 实体

发送UM-RLC实体通过UM-SAP从高层接收RLC SDU。

如果RLC SDU的尺寸大于UMD PDU中的可用空间，那么发送UM RLC实体将对RLC SDU进行分段，得到适当大小的UMD PDU。分段和/或级联之后的UMD PDU大小应小于或等于最大UL UMD PDU大小。如果配置了MACi/is而且RLC PDU大小设置为“flexible size”，分段和/或级联之后的UMD PDU大小应大于或等于最小UL RLC PDU大小。如果待传数据的大小不足以创建一个最小尺寸的UMD PDU，那么也允许创建一个包含所有待传数据的UMD PDU，虽然其尺寸小于最小UL RLC PDU大小。UMD PDU可能包含分段的和/或者级联的RLC SDU。UMD PDU可能进行填充以确保得到有效的长度。使用长度标识（Length Indicator）来定义UMD PDU中RLC SDU的边界。长度标识也用于定义是否在UMD PDU中使用了填充。

如果配置并且启动了加密，那么UMD PDU（除了UMD PDU包头）在发送给下层之前被加密。

发送UM RLC实体通过一个CCCH、SHCCH、DCCH、CTCH或者一个DTCH逻辑信道向下层发送UMD PDU。

4.2.3.1.2 接收 UM RLC 实体

接收UM-RLC实体通过配置的逻辑信道从下层接收UMD PDU。当配置了重复避免和重排序功能时，可能会从下层收到一个或多个输入。当增加或删除输入时，可以在接收UM RLC实体内不改变缓冲区内容、状态变量和定时器。当未配置重复避免和重排序功能时，只会从下层收到一个输入并且不会被重配置。

如果配置了重复避免和重排序功能，它将是接收UM RLC实体中应用于输入UMD PDU流的第一个接收功能，该功能只能在UE侧配置并且不用于UTRAN。该功能完成对从一个或多个输入收到的UMD PDU的重复检测和重排序，产生顺序排列的PDU，并递交给下一个顺序RLC接收功能。

接收UM RLC实体对接收到的UMD PDU（除了UMD PDU包头）进行解密（如果配置并且启动了加密）。它从接收到的UMD PDU中去除RLC包头，并且重组RLC SDU（如果发送UM RLC实体进行了分段和/或者级联）。

若接收的UM RLC实体配置了SDU乱序递交的功能，该实体会在包含1个SDU的所有PDU都收到的时候尽快的重组出SDU，并递交给高层，尽管可能有更早的PDU还没有收到也不影响这里的处理。UM RLC接收实体将会存储不能组成SDU的PDU，并等待发送端重新传输丢失的PDU。当和一个SDU相关的PDU都收齐后将对应的PDU从缓存中删除，或者由于序号窗口的功能、或者由于存储定时器的功能也需将相应的PDU从缓存中删除。只有用户侧配置乱序递交的功能，且该功能只用于MCCH。接收UM RLC实体通过UM-SAP向上层传送RLC SDU。如果配置了SN_Delivery，映射了RLC SDU的RLC PDU序列号也需要通过UM-SAP向上层传递。

4.2.4 确认模式（AM）RLC 实体

图5显示了确认模式RLC实体模型。

AM RLC实体可以被配置为使用一个或者两个逻辑信道。图5显示了使用一个逻辑信道（实线）和使用两个逻辑信道（虚线）时AM RLC实体的模型。

如果配置了一个逻辑信道，AM RLC实体的发送侧通过这个逻辑信道向下层发送AMD和控制PDU。如果配置了固定的RLC PDU大小，AMD PDU和控制PDU应该具有相同的RLC PDU大小。如果配置了可变的RLC PDU大小，AMD PDU大小可以变化，只要不超过最大RLC PDU大小即可。

在上行链路配置两个逻辑信道的情况下，在第一个逻辑信道上发送AMD PDU和除确认状态报告、MRW ACK SUFI及WINDOW SUFI之外的控制PDU，在第二个逻辑信道上发送确认状态报告、MRW ACK SUFI和WINDOW SUFI。在下行链路配置两个逻辑信道的情况下，可以在两个逻辑信道中的任意一个上发送AMD PDU和控制PDU。

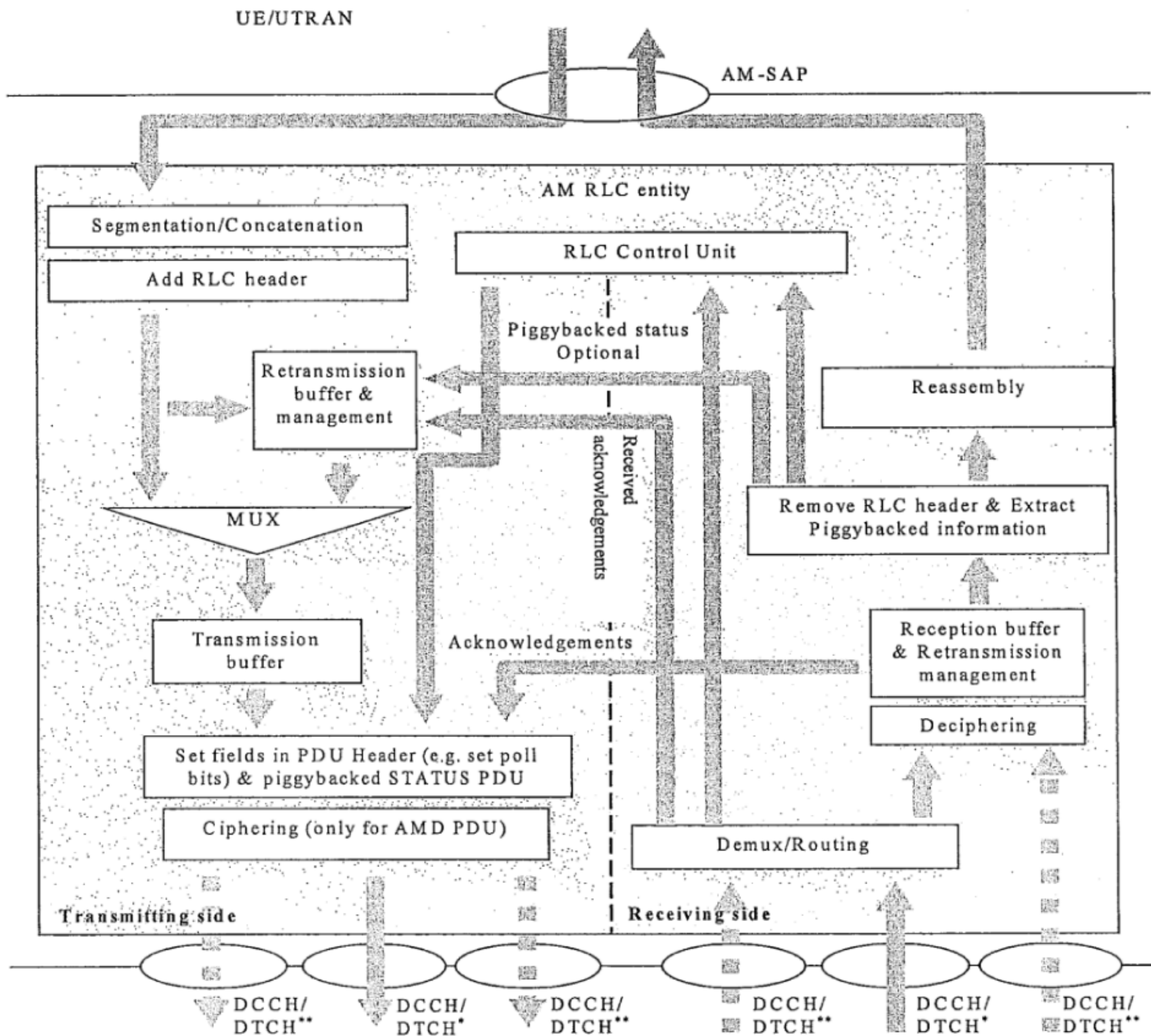


图5 确认模式实体的模型

4.2.4.1.1 发送侧

AM-RLC实体的发送侧通过AM-SAP从上层接收RLC SDU。

如果配置了固定RLC PDU大小，则以一个固定的长度对RLC SDU进行分段和/或者级联组成AMD PDU。当接收到的RLC SDU的长度大于AMD PDU中可用的空间时，对RLC SDU进行分段处理。上行AMD PDU的长度是半静态的参数，它是由上层进行配置的，并且只能够通过上层重建AM RLC实体来进行修改。

如果配置了可变RLC PDU大小，RLC SDU将被分段和/或级联以使创建的RLC PDU大于或等于最小UL RLC PDU大小并且小于或等于最大UL AMD PDU大小。如果待传数据的大小不足以创建一个最小尺

寸的AMD PDU, 那么也允许创建一个包含所有待传数据的AMD PDU, 虽然其尺寸小于最小UL RLC PDU大小。

注: 在下行方向, 如果配置了可变RLC PDU大小, 当SDU大于最大RLC PDU大小时, RLC SDU将被分段, 此外还可以使用级联, 只要形成的RLC PDU小于最大RLC PDU大小即可。

AMD PDU可能包含分段和/或者级联的RLC SDU。AMD PDU可能进行填充以确保具有有效的长度。如果配置了固定RLC PDU大小, 使用长度指示或HE域的一个特殊值来定义AMD PDU中RLC SDU的边界, 同时, 长度标识也用来定义是否在AMD PDU中包含了填充或者Piggybacked状态PDU。如果配置了可变RLC PDU大小, 长度指示的大小由高层配置。HE域的特殊值的使用由高层配置。

在经过分段和/或者级联以后, AMD PDU被放入重传缓冲区和MUX中。

根据从对等AM RLC实体接收到的状态PDU或者Piggybacked状态PDU中的状态报告, 重传缓冲区中的AMD PDU被删除或者重传。这个状态报告可能包含了关于对等AM RLC实体接收到的每一个AMD PDU的正面或者负面的确认信息。

MUX对重传缓冲区中需要重传的AMD PDU和来自分段/级联功能的新生成的AMD PDU进行复用。

PDU被发送到完成AMD PDU包头的功能模块, 该功能模块可能使用Piggybacked状态信息替换填充信息单元。为了匹配AMD PDU中的剩余空间Piggybacked状态PDU可以具有各种的长度。根据来自RLC控制单元的指示各个字段(例如Polling Bit)设置值的信息, 完成对AMD PDU包头的处理。如果需要的话, 该功能还进行对来自RLC控制单元(复位和复位确认PDU)以及来自接收缓冲区(Piggybacked状态和状态PDU)的控制PDU与AMD PDU的复用操作。

然后对AMD PDU进行加密(如果配置的话)操作。AMD PDU包头不进行加密。AMD PDU中的Piggybacked状态PDU和填充信息(如果有的话)被加密。控制PDU(例如状态PDU, 复位PDU和复位确认PDU)不进行加密。

AM RLC实体的发送侧通过一个或者两个DCCH或者DTCH逻辑信道向下层发送AMD PDU。

4.2.4.1.2 接收侧

AM-RLC实体的接收侧通过配置的逻辑信道从下层接收AMD和控制PDU。

如果配置了固定RLC PDU大小, AMD PDU的长度是半静态的参数, 它是由上层进行配置的, 并且只能够通过上层重建AM RLC实体来进行修改。当AMD PDU的长度没有被配置时, 它由第一个收到的AMD PDU的长度决定。上下行AMD PDU的长度可以不一致。

如果配置了灵活RLC PDU大小, AMD PDU大小就是可变的, 最大可以达到最大RLC PDU大小, 并且长度指示的大小由高层配置。

AMD PDU被送到解密单元, 在那里AMD PDU(不包括AMD PDU包头)被解密(如果配置并且启动了加密), 然后发送到接收缓冲区。

在一个RLC SDU被完全接收到之前, AMD PDU一直被放在接收缓冲区中。接收端通过它的发送侧向AM RLC对等实体发送一个或者多个状态PDU来指示成功接收或者要求重传丢失的AMD PDU。如果在AMD PDU中发现了Piggybacked状态PDU, 那么Piggybacked状态PDU将被发送到位于AM RLC实体发送侧的重传缓冲和管理单元, 用于清除已经得到正面确认的AMD PDU的缓冲区, 以及指示哪些AMD PDU需要重传。

一旦完全接收到一个RLC SDU, 重组单元将重组相关的AMD PDU并且通过AM-SAP向上层发送。

复位和复位确认PDU被发送到RLC控制单元以用于过程操作。如果需要向对等AM RLC实体进行响应, AM RLC实体发送侧的RLC控制单元将发送一个适当的控制PDU。接收到的状态PDU被发送到位于AM RLC实体发送侧的重传缓冲和管理单元, 用于清除已经得到正面确认的AMD PDU的缓冲区, 以及指示哪些AMD PDU需要重传。

5 功能

以下是RLC子层所支持的功能, 下列功能的详细描述请参见3GPP TS 25.301:

- 分段和重组;
- 级联;
- 填充;
- 用户数据的传送;
- 纠错;
- 按序发送高层 PDU;
- 重复检测;
- 流量控制;
- 序号检查;
- 协议错误检测和恢复;
- 加密;
- SDU 丢弃;
- 乱序递交;
- 重复避免和重排序。

6 提供给上层的业务

本节描述了RLC子层提供给上层的不同业务, 还包括RLC功能与不同RLC业务之间的映射。RLC业务的详细描述参见3GPP TS 25.301。

- 透明数据传送业务。

需要以下功能来支持透明数据传送:

- 分段和重组;
- 用户数据的传送;
- SDU 丢弃。

- 非确认数据传送业务。

需要以下功能来支持无确认数据传送:

- 分段和重组;
- 级联;
- 填充;
- 用户数据的传送;
- 加密;
- 序号检查;

- SDU 丢弃;
- 乱序递交;
- 重复避免和重排序;
- 序列号的提供。

— 确认数据传送业务。

需要以下功能来支持确认数据传送:

- 分段和重组;
- 级联;
- 填充;
- 用户数据的传送;
- 纠错;
- 按序传送上层 PDU;
- 副本检测;
- 流量控制;
- 协议错误检测和恢复;
- 加密;
- SDU 丢弃。

— 根据上层的定义进行 QoS 的维护。

— 不可恢复错误的通知。

6.1 业务/功能在逻辑信道上的映射

表1~表4表示了业务和功能在UL/DL和UE/UTRAN中逻辑信道上的适用性。在列中的“+”表示对于所描述的逻辑信道该业务/功能是可用的,“—”则表示该业务/功能不可用。

表1 UE 上行链路 RLC 模式和功能

业务	功能	CCCH	SHCCH	DCCH	DTCH
透明业务	适用性	+	+	—	+
	分段	—	—	—	+
	用户数据的传送	+	+	—	+
	SDU 丢弃	—	—	—	+
非确认业务	适用性	—	—	+	+
	分段	—	—	+	+
	级联	—	—	+	+
	填充	—	—	+	+
	用户数据的传送	—	—	+	+
	加密	—	—	+	+
	SDU 丢弃	—	—	+	+
确认业务	适用性	—	—	+	+
	分段	—	—	+	+
	级联	—	—	+	+
	填充	—	—	+	+

表 1 (续)

业务	功能	CCCH	SHCCH	DCCH	DTCH
确认业务	用户数据的传送	—	—	+	+
	流量控制	—	—	+	+
	纠错	—	—	+	+
	协议错误检测和恢复	—	—	+	+
	加密	—	—	+	+
	SDU 丢弃	—	—	+	+

表2 UE 侧下行链路 RLC 模式和功能

业务	功能	BCCH	PCCH	SHCCH	CCCH	DCCH	DTCH	CTCH	MCCH	MTCH	MSCH
透明业务	实用性	+	+	—	—	+	+	—	—	—	—
	重组	—	—	—	—	+	+	—	—	—	—
	用户数据传输	+	+	—	—	+	+	—	—	—	—
无确认业务	实用性	—	—	+	+	+	+	+	+	+	+
	重组	—	—	+	+	+	+	+	+	+	+
	解密	—	—	—	—	+	+	—	—	—	—
	序列号检查	—	—	+	+	+	+	+	+	+	+
	用户数据传输	—	—	+	+	+	+	+	+	+	+
	重复避免与重排序	—	—	—	+	—	—	—	—	+	—
	乱序 SDU 传递	—	—	—	—	—	—	—	+	—	—
确认业务	实用性	—	—	—	—	+	+	—	—	—	—
	重组	—	—	—	—	+	+	—	—	—	—
	纠错	—	—	—	—	+	+	—	—	—	—
	流量控制	—	—	—	—	+	+	—	—	—	—
	按序发送	—	—	—	—	+	+	—	—	—	—
	副本检测	—	—	—	—	+	+	—	—	—	—
	协议错误检测和恢复	—	—	—	—	+	+	—	—	—	—
	解密	—	—	—	—	+	+	—	—	—	—
	用户数据传输	—	—	—	—	+	+	—	—	—	—
	SDU 丢弃	—	—	—	—	+	+	—	—	—	—

表3 UTRAN 下行链路侧 RLC 模式和功能

业务	功能	BCCH	PCCH	CCCH	SHCCH	DCCH	DTCH	CTCH	MCCH	MTCH	MSCH
透明业务	适用性	+	+	—	—	+	+	—	—	—	—
	分段	—	—	—	—	+	+	—	—	—	—
	用户数据的传送	+	+	—	—	+	+	—	—	—	—
	SDU 丢弃	—	—	—	—	+	+	—	—	—	—
非确认业务	适用性	—	—	+	+	+	+	+	+	+	+
	分段	—	—	+	+	+	+	+	+	+	+
	级联	—	—	+	+	+	+	+	+	+	+
	填充	—	—	+	+	+	+	+	+	+	+
	加密	—	—	—	—	+	+	—	—	—	—
	用户数据传输	—	—	+	+	+	+	+	+	+	+
	SDU 丢弃	—	—	—	—	+	+	—	+	+	+

表3 (续)

业务	功能	BCCH	PCCH	CCCH	SHCCH	DCCH	DTCH	CTCH	MCCH	MTCH	MSCH
确认业务	适用性	—	—	—	—	+	+	—	—	—	—
	分段	—	—	—	—	+	+	—	—	—	—
	级联	—	—	—	—	+	+	—	—	—	—
	填充	—	—	—	—	+	+	—	—	—	—
	用户数据传送	—	—	—	—	+	+	—	—	—	—
	流量控制	—	—	—	—	+	+	—	—	—	—
	纠错	—	—	—	—	+	+	—	—	—	—
	协议错误检测和恢复	—	—	—	—	+	+	—	—	—	—
	加密	—	—	—	—	+	+	—	—	—	—
	SDU 丢弃	—	—	—	—	+	+	—	—	—	—

表4 UTRAN 上行链路侧 RLC 模式和功能

业务	功能	CCCH	SHCCH	DCCH	DTCH
透明业务	适用性	+	+	—	+
	重组	—	—	—	+
	用户数据传输	+	+	—	+
无确认业务	适用性	—	—	+	+
	重组	—	—	+	+
	解密	—	—	+	+
	序列号检查	—	—	+	+
	用户数据传输	—	—	+	+
确认业务	适用性	—	—	+	+
	重组	—	—	+	+
	纠错	—	—	+	+
	流量控制	—	—	+	+
	按序发送	—	—	+	+
	副本检测	—	—	+	+
	协议错误检测和恢复	—	—	+	+
	解密	—	—	+	+
	用户数据传输	—	—	+	+
	SDU 丢弃	—	—	+	+

7 期望从 MAC 获得的业务

MAC子层向上层提供的业务的详细描述参见3GPP TS 25.301。

— 数据传输。

8 层与层通信的单元

RLC层与其他层间的交互用原语来描述，这里原语表示RLC子层与其他层之间控制与信息的逻辑交换。原语不对具体实现进行说明或限制。

8.1 RLC 和上层之间的原语

RLC和上层之间的原语见表5。其中，Req. (Requirement) 为请求，Ind. (Indication) 为指示，Resp. (response) 为响应，Conf. (confirmation) 为确认。

表5 RLC 和上层之间的原语

名称	参数			
	Req.	Ind.	Resp.	Conf.
RLC-AM-DATA	Data, CNF, DiscardReq, MUI, UE-ID type indicator	Data, DiscardInfo	没有定义	Status, MUI
RLC-UM-DATA	Data, UE-ID type indicator, DiscardReq, MUI	Data, Sequence_Number	没有定义	MUI
RLC-TM-DATA	Data, UE-ID type indicator, DiscardReq, MUI	Data, Error_Indicator	没有定义	MUI
CRLC-CONFIG	E/R, Stop (UM/AM only), Continue (UM/AM only), Ciphering Elements (UM/AM only), TM_parameters (TM only), UM_parameters (UM only), AM_parameters (AM only), SN_Delivery (UM only)	没有定义	没有定义	没有定义
CRLC-SUSPEND (UM/AM only)	N	没有定义	没有定义	VT (US) (UM only), VT (S) (AM only)
CRLC-RESUME (UM/AM only)	没有参数	没有定义	没有定义	没有定义
CRLC-STATUS	没有定义	EVC	没有定义	没有定义

每个原语的定义如下:

RLC-AM-DATA-Req/Ind/Conf

- RLC-AM-DATA-Req, 在确认模式下, 高层使用该原语请求传输 RLC PDU。
- RLC-AM-DATA-Ind, AM RLC 实体使用该原语向上层传送在确认模式下发送的 RLC SDU, 以及向上层指示对等 RLC AM 实体丢弃的 RLC SDU。
- RLC-AM-DATA-Conf, AM RLC 实体使用该原语向上层确认对等 RLC AM 实体对 RLC SDU 的接收或向上层指示一个被丢弃的 SDU。

RLC-UM-DATA-Req/Ind/Conf

- RLC-UM-DATA-Req, 在非确认模式下, 上层使用该原语请求传输 RLC SDU。
- RLC-UM-DATA-Ind, UM RLC 实体使用该原语向上层传送在非确认模式下发送的 RLC SDU。
- RLC-UM DATA-Conf, UM RLC 实体使用该原语来向上层指示一个被丢弃的 SDU。如果配置了 SN_Delivery, UM RLC 实体会还使用 RLC-UM-DATA-Ind 原语向上层递交 Sequence_Number。

RLC-TM-DATA-Req/Ind/Conf

- RLC-TM-DATA-Req, 在透明模式下, 上层使用该原语请求传输 RLC SDU。
- RLC-TM-DATA-Ind, TM RLC 实体使用该原语向上层传送在透明模式下发送的 RLC SDU。
- RLC-TM-DATA-Conf, TM RLC 实体使用该原语来向上层指示一个被丢弃的 SDU。

CRLC-CONFIG-Req

上层使用该原语用于建立、重建、释放、停止、继续或者更改 RLC。对于 UM 和 AM 操作, 原语的作用范围包括加密单元。

CRLC-SUSPEND-Req/Conf

- CRLC-SUSPEND-Req, 上层使用该原语挂起 UM 或者 AM RLC 实体。
- CRLC-SUSPEND-Conf, UM 或者 AM RLC 实体使用该原语确认实体已经被挂起。

CRLC-RESUME-Req

在 UM 或者 AM RLC 实体被挂起后, 上层使用该原语进行实体的恢复。

CRLC-STATUS-Ind

RLC 实体使用该原语向上层发送状态信息。

8.2 原语参数

在原语中, 用到下列参数:

1) 参数“data”是 RLC SDU, 它被映射到 RLC PDU 中的数据字段中。在使用 AM 或者 UM RLC 实体的情况下, “data”参数的长度是 8bit 的整数倍; 否则(TM RLC 实体)“data”参数的长度是一个比特流, 它的长度不一定是 8bit 的整数倍。

2) 参数“确认请求(CNF)”指示了 AM RLC 实体的发送侧是否需要确认对等 RLC AM 实体已收到该 RLC SDU。如果需要, 那么一旦得到接收 AM RLC 实体关于组成一个 RLC SDU 的所有 AMD PDU 的正面确认信息, 发送 AM RLC 实体将通知上层。

3) 参数“消息单元标识(MUI)”是 RLC SDU 的标识, 用来指示 RLC-AM-DATA-Conf.原语所确认, 或者 RLC-AM/UM/TM-DATA-Conf.原语所丢弃的是哪个 RLC SDU。

4) 参数“E/R”指示了 RLC 实体的建立、重建、释放或修改, 其中重建只适用于 AM 和 UM RLC 实体。如果要求的是重建, 那么状态变量和可配置参数将根据 9.7.7 节进行初始化。如果要求的是释放, 那么所有协议参数、变量和计数器将被释放, RLC 实体进入 NULL 状态。如果要求的是修改, 那么仅对上层所指示的协议参数(例如加密参数)进行修改, 而保持其他协议参数, 例如协议变量、协议定时器和协议状态不变。如果上行或下行 AMD PDU 的大小发生改变, 那么 AM RLC 实体将被重建。其他协议参数的修改不需要重建。

5) 参数“事件编码(EVC)”指示了 CRLC-STATUS-ind 的原因, 例如像数据链路层丢失这样的不可恢复的错误, 或者诸如复位此类的可恢复状态事件。

6) 参数“加密单元”只适用于 UM 和 AM 操作。这些参数是加密模式、密钥、传送激活时间(在传送端激活一个新的加密配置的 SN), 接收激活时间(在接收端激活一个新的加密配置的 SN)和 HFN(超帧号)。

7) 参数“AM_parameters”仅适用于 AM 操作。它包括可以设置为固定值或可变大小的 AMD PDU 大小、最大 UL AMD PDU 大小(仅用于高层配置了灵活 PDU 大小的情况, 见 9.2.2.8 节)、长度指示大小、顺序传送标识(指示 RLC SDU 是按顺序传送到上层还是可以无序传送到上层)、定时器值(见 9.5 节)、HE 域特殊值的使用、协议参数值(见 9.6 节)、轮询触发器(见 9.7.1 节)、状态触发器(见 9.7.2 节)、周期性状态阻塞配置(见 9.7.2 节)、SDU 丢弃模式(见 9.7.3 节)、最小 WSN(见 9.2.2.11.3)和发送 MRW。最小 WSN 总是大于或等于最小传输块集的传输块数目。发送 MRW 指示关于每一个被丢弃的 RLC SDU 的信息都被发送到接收端, 并且即使要丢弃的 RLC SDU 没有任何分段被发送给下层, MRW SUFI 也将被发送给接收端。

8) 参数“删除信息(DiscardInfo)”向上层指示了在对等 RLC AM 实体中删除的 RLC SDU。它只适用于配置了顺序传送的情况, 当上层要求可靠的数据传输时使用到它。

9) 参数“终止 (Stop)”仅适用于 AM 和 UM RLC 实体, 用于指示 RLC 实体 (见 9.7.6 节) 不发送和接收任何 RLC PDU。

10) 参数“继续 (continue)”仅适用于 AM 和 UM RLC 实体, 用于指示 RLC 实体继续发送和接收 RLC PDU。

11) UM 参数只适用于 UM 操作。它包括 Timer_Discard 值 (见 9.5 节)、使用 E-bit 的备用解释 (见 9.2.2.5 节)、最大 UL UMD PDU 尺寸 (见 9.2.2.8 节)、下行 RLC UMLI 大小 (见 9.2.2.8 节) 和 SN_Delivery。对于用户侧的 UM RLC 接收实体, 需要包含一个额外的参数用于指示是否使用 SDU 乱序递交功能 (见 11.2.3.2 节)。若需要使用 SDU 乱序递交功能, 那么还需要增加配置参数 OSD_Window_Size (见 9.6 节) 以及定时器 Timer_OSD (见 9.5 节) 超时值。对于用户侧的 UM RLC 接收实体, 需要包含一个额外的参数用于指示是否使用重复避免和重排序功能 (见 9.7.10 节), 如果使用了重复避免和重排序功能, 还需要包含参数 DAR_Window_Size (见 9.6 节) 和定时器值 Timer_DAR (见 9.5 节)。如果配置支持乱序, 需要包含参数 Configured_Rx_Window_Size。如果配置了 SN_Delivery 参数, 接收 RLC 实体将携带 Sequence_Number 的 RLC SDU 递交给高层。当配置了 SN_Delivery 参数, 下行发送 RLC 实体对 RLC SDU 既不级联也不分段, 上行发送 RLC 实体也不对 RLC SDU 进行级联, 但是可以分段。

12) TM 参数只适用于 TM 操作。它包含例如: 分段指示 (见 9.2.2.9 和 11.1.2.1 节)、Timer_Discard 值 (见 9.5 节) 和错误 SDU 指示的发送 (见 11.1.3 节)。

13) 参数“N”指示: 对于 AM, RLC 实体不会发送“Sequence Number” \geq VT(S)+N 的 PDU; 对于 UM, RLC 实体不会发送“Sequence Number” \geq VT(US)+N 的 PDU, 其中 N 是非负整数。

14) 参数“VT(S)”用于在 AM 情况下指示发送状态变量的值。

15) 参数“VT(US)”用于在 UM 情况下指示 UM 数据状态变量的值。

16) 参数“Error_Indicator”指示 RLC SDU 是否错误 (见 11.1.3 节)。

17) 参数“UE-ID type indicator”指示相关 RLC SDU 使用的 RNTI 类型 (U-RNTI 或者 C-RNTI)。在 UE 上不要求这个参数。

18) 参数“DiscardReq”指示 RLC 传输实体是否需要向上层通知被丢弃的 RLC SDU。如果需要, 当 SDU 被丢弃时, RLC 传输实体通知上层。

19) 参数“Status”仅应用于 AM 操作。这一参数指示了一个 RLC SDU 已被传输成功或已被丢弃。

20) 参数 Sequence_Number 是 RLC PDU 中的 Sequence Number (SN) 域的值, 只在为 UMD RLC 实体配置了 SN_Delivery 的情况下用于 UM 操作。

21) 参数 Minimum UL PDU size 用于上行 AM 和 UM 操作。Minimum UL PDU size 决定 RLC AMD 或 RLC UMD PDU 在分段或级联后的最小尺寸 (见 9.2.2.9 节)。如果待传数据量不足以填满一个最小尺寸的 PDU, 允许传输一个包含所有数据的 PDU, 虽然其尺寸小于 Minimum UL RLC PDU size。

9 对等层通信元素

9.1 协议数据单元

本节中定义的结构是标准化的。

9.1.1 数据 PDU

a) TMD PDU (透明模式数据 PDU)

TMD PDU 是用来传送不加任何 RLC 开销的 RLC SDU 数据。当 RLC 处于透明模式时使用 TMD PDU。

b) UMD PDU (非确认模式数据PDU)

UMD PDU 是用来传送包含 RLC SDU 数据的顺序编号的 PDU。当 RLC 被配置为无确认数据传送时使用 UMD PDU。

c) AMD PDU (确认模式数据 PDU)

AMD PDU 是用来传送包含 RLC SDU 数据的顺序编号的 PDU。当 RLC 被配置为确认数据传送时使用 AMD PDU。

9.1.2 控制 PDU

控制 PDU 仅用于确认模式，详见表 6。

a) 状态 PDU 和 Piggybacked 状态 PDU

状态 PDU 和 Piggybacked 状态 PDU 在下列情况下被使用：

- 接收端用来通知发送端在接收端丢失的和收到的 AMD PDU；
- 接收端用来通知发送端所允许的传输窗口长度；
- 发送端用来请求接收端移动接收窗口；
- 接收端向发送端确认收到移动接收窗口的请求。

b) 复位 PDU

复位 PDU 用于复位所有协议状态、协议变量以及对等 RLC 实体的协议定时器，以此来同步两个对等实体。它是由发送端向接收端发送的。

c) 复位确认 (RESET ACK) PDU

RESET ACK PDU 是对复位 PDU 的一个确认。它是由接收端向发送端发送的。

表6 RLC PDU 名称和描述

数据传送模式	PDU 名称	描述
透明	TMD	透明模式数据
无确认	UMD	顺序非确认模式数据
确认	AMD	顺序确认模式数据
	STATUS	请求或无请求状态报告、改变窗口大小命令、SDU 丢弃命令、或者 SDU 丢弃确认
	Piggybacked STATUS	Piggybacked 请求或无请求状态报告、改变窗口大小命令、SDU 丢弃命令、或者 SDU 丢弃确认
	RESET	复位命令
	RESET ACK	复位确认

9.2 格式和参数

本节中定义的 RLC PDU 和它们的参数的格式是标准化的。

9.2.1 格式

本节详细说明了 RLC PDU 的格式。每个 RLC PDU 的参数在 9.2.2 节中解释。

9.2.1.1 概述

RLC PDU 是一个比特流。在 9.2 节中的图中，比特流是用表格来表示的，在这个表格中，第一个比特是表格中第一行最左边的比特，最后一个比特是表格中最后一行最右边的比特。通常，比特流是逐行从左到右来读取的。

按照所提供的业务，RLC SDU是非空的比特流，或者是长度为8bit整数倍的比特流。将RLC SDU从第一个比特开始放入RLC PDU。

9.2.1.2 TMD PDU

当RLC在透明模式下工作时，使用如图6所示的TMD PDU传送用户数据。RLC不对SDU附加任何开销。没有对数据长度进行整数字节的限制。

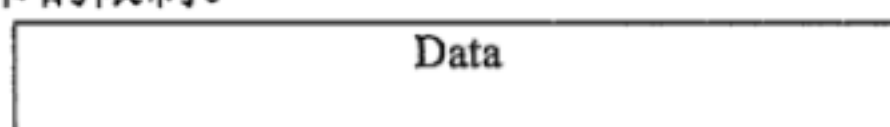
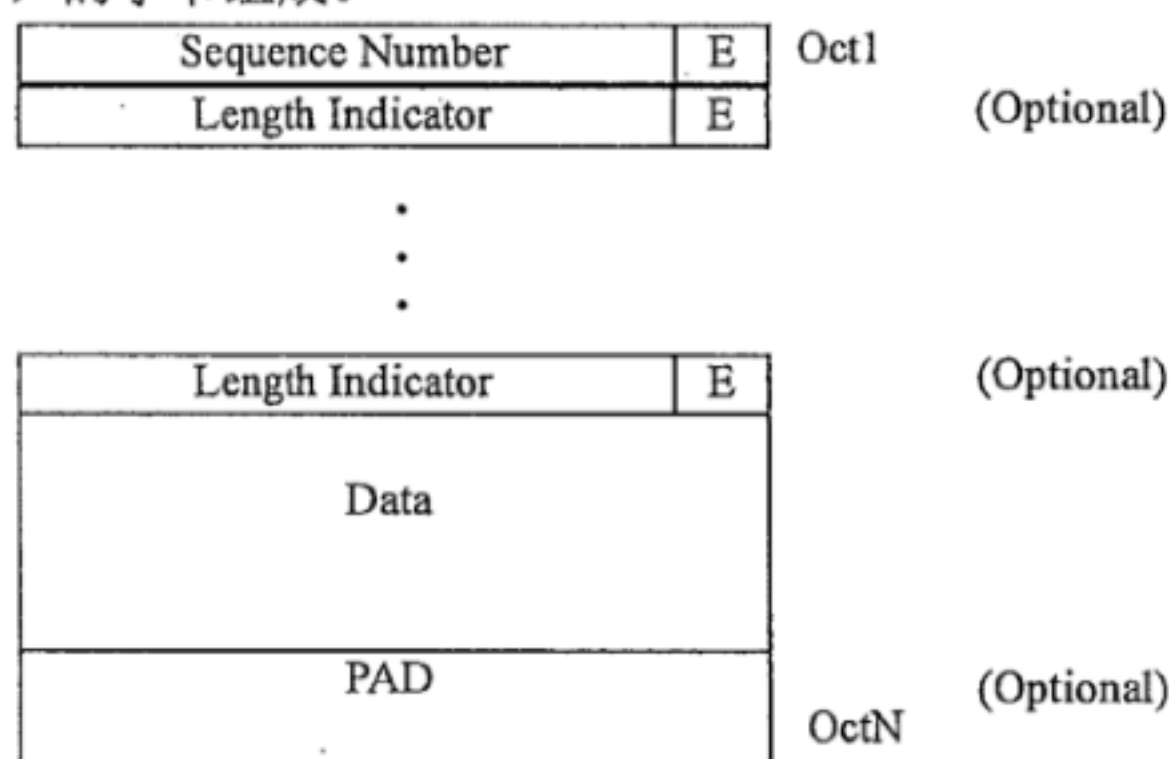


图6 TMD PDU

9.2.1.3 UMD PDU

当RLC工作在非确认模式下时，使用如图7所示的UMD PDU 传送用户数据。数据部分的长度应该是整数字节。UMD PDU的包头包括第一个字节，这个字节包含了序号。RLC包头由第一个字节和所有包含长度标识（Length Indicator）的字节组成。

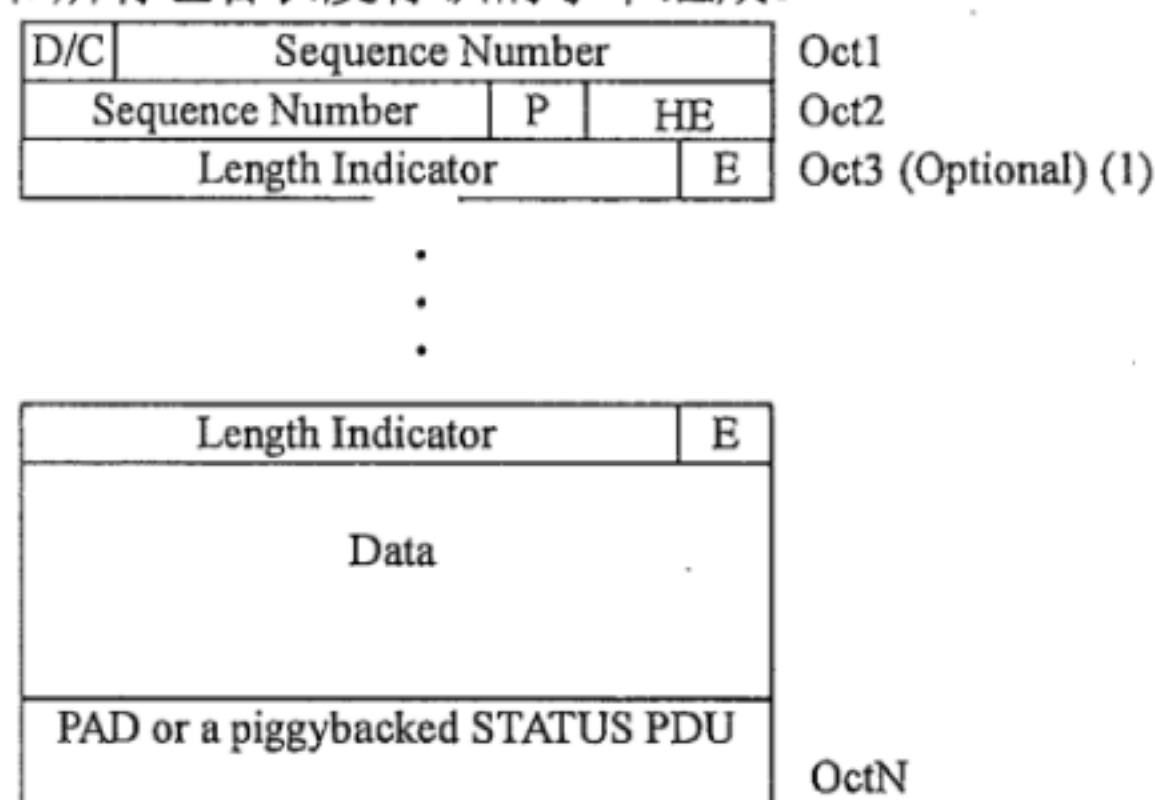


注：“Length Indicator”可能是15bit。

图7 UMD PDU

9.2.1.4 AMD PDU

当RLC操作在确认模式时，使用如图8所示的AMD PDU传送用户数据、piggybacked 状态信息和Polling比特。数据部分的长度应该是整数字节。AMD PDU的头部包含前两个字节，这两个字节中包括序号。RLC包头由前两个字节和所有包含长度标识的字节组成。



注：“Length Indicator”可能是15bit。

图8 AMD PDU

9.2.1.5 STATUS PDU

STATUS PDU用来在两个RLC AM实体之间交换状态信息。

STATUS PDU的格式如图9所示。每个超字段（SUFI）的长度取决于它的类型和内容。

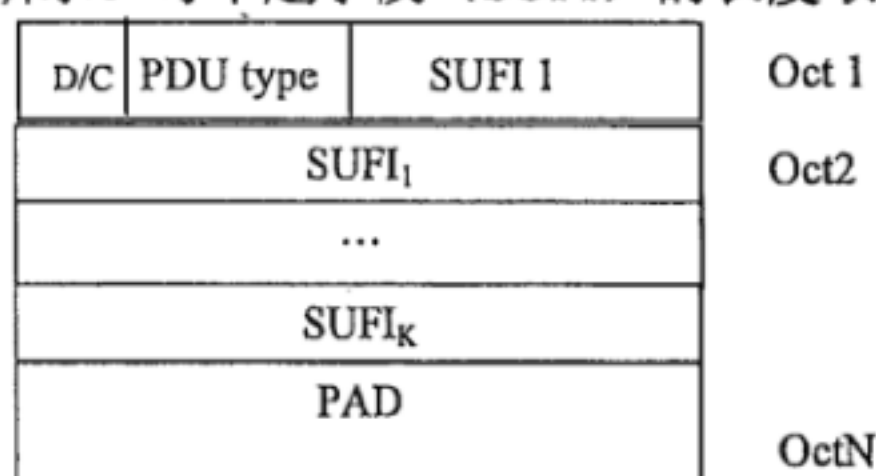


图9 状态 PDU (STATUS PDU)

状态PDU可以包含不同类型的超字段。状态PDU的长度是可变的，其上限是由发送控制PDU的逻辑信道所使用的最大RLC PDU长度所确定。通过填充位使之符合发送控制PDU的逻辑信道所采用的PDU大小。STATUS PDU的长度应该是整数字节。

9.2.1.6 Piggybacked STATUS PDU

除了D/C字段被保留位（R2）替代外，如图10所示的piggybacked STATUS PDU的格式与状态PDU相同。如果数据不能填满整个AMD PDU，则该PDU可以填充入AMD PDU。PDU类型字段被设置为“000”，其他值在这个版本的协议中是无效的。

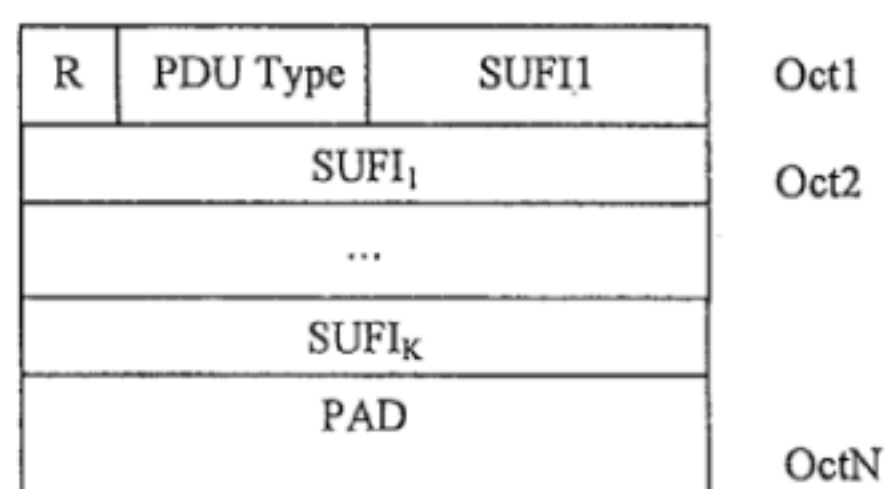


图10 Piggybacked 状态 PDU

9.2.1.7 复位, 复位确认PDU

RESET PDU中包含一个1bit的顺序号字段（RSN）。响应发送的RESET ACK PDU通过携带这个比特值来告诉对等实体它所响应的是哪一个RESET PDU，如图11所示。

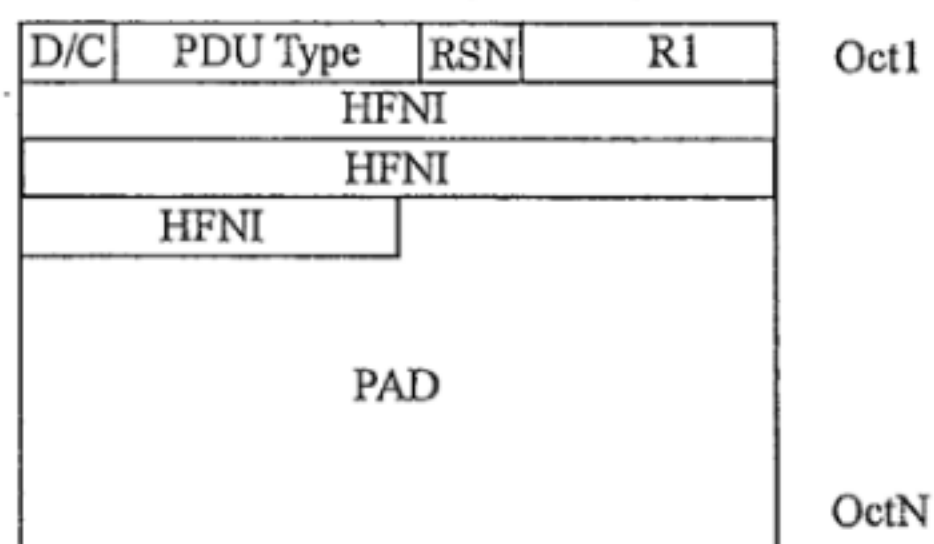


图11 RESET, RESET ACK PDU

RESET或者RESET ACK PDU的大小是可变的，其上限由发送控制PDU的逻辑信道所使用的最大RLC PDU大小所确定。如果上层配置了固定RLC PDU大小，通过填充位使之符合发送控制PDU的逻辑信道所使用的某个PDU尺寸。RESET或者RESET ACK PDU的长度应该是8bit的整数倍。如果高层配置了灵活RLC PDU大小，填充位只用于使得RESET和RESET ACK PDU的长度成为8bit的整数倍。

9.2.2 参数

如果在每个字段的定义中没有另外说明，那么参数中的比特应该被如下解释：最左边的比特流是第一个并且是最高有效位，最右边的比特是最后一个并且是最低有效位。

除非另外说明，整数将使用无符号整数标准的二进制编码方式来进行编码。在所有情况下，包括如表中所示的数值扩展超过一个字节的的情况，对RLC PDU的读取从MSB到LSB的方向进行。

9.2.2.1 D/C字段

长度：1bit。

D/C字段指示了AM PDU的类型。数据PDU或者控制PDU。

Bit	描述
0	控制 PDU
1	数据 PDU

9.2.2.2 PDU类型

长度：3bit。

PDU类型字段指示了控制PDU的类型。

Bit	PDU 类型
000	状态
001	复位 (RESET)
010	复位确认 (RESET ACK)
011~111	保留 (在这个版本的协议中带有这种编码的 PDU 会被丢弃)

9.2.2.3 序列号 (Sequence Number, SN)

该字段指示了按二进制编码的RLC PDU的序列号。

PDU 类型	长度	注
AMD PDU	12 bits	用于重传和重组
UMD PDU	7 bits	用于重组

9.2.2.4 Polling比特 (P)

长度：1bit。

该字段用来请求从接收端得到状态报告（一个或者多个STATUS PDU）。

Bit	描述
0	不请求状态报告
1	请求一状态报告

9.2.2.5 扩展比特 (E)

长度：1bit。

该比特的解释依赖于RLC模式及高层配置：

— 在 UMD PDU 中，第一个字节中的扩展比特可以用正常 E-bit 解释或扩展 E-bit 解释，依赖于高层配置。所有其他字节中的扩展比特使用正常 E-bit 解释。

— 在 AMD PDU 中，扩展比特一直使用正常 E-bit 解释。

正常 E-bit 解释：

Bit	描述
0	下一个字段是数据、piggybacked STATUS PDU 或者填充
1	下一个字段是长度指示和 E 比特

扩展 E-bit 解释：

Bit	描述
0	下一个字段是一个完整的 SDU，没有被分段、级联或填充
1	下一个字段是长度指示或 E-bit

9.2.2.6 保留1 (R1)

长度: 3 bits。

RESET PDU和RESET ACK PDU中的该字段用来完成八位组对齐。其编码为“000”。该字段的其他值被保留，并且在这个协议版本中将被认为是无效的。

9.2.2.7 包头扩展类型 (HE)

长度: 2 bits。

该两比特字段指示了下一个字节是数据还是“Length Indicator”和E比特。

取值	描述
00	之后的八位组包含数据
01	之后的八位组包含长度标识和 E 比特
10	该值是 HE 字段的特殊值，用于指示其后的八位组包含数据并且 PDU 的最后一个八位组是 SDU 的最后一个八位组
11	保留（在这个版本协议中，这种编码的 PDU 将被丢弃）

9.2.2.8 长度标识 (LI)

除非有扩展比特指示UMD PDU包含一个没有被分段、级联、填充的完整的SDU，或者HE字段指示AMD PDU包含RLC SDU的最后一个字节，“Length Indicator”用于指示在PDU内终结的每一个RLC SDU的最后一个字节。如果扩展比特指示UMD PDU包含一个没有被分段、级联、填充的完整的SDU，或者HE字段指示AMD PDU包含RLC SDU的最后一个字节，则RLC PDU中不包含LI字段。

除了为特殊目的预先定义并且在下表中列出的数值，“长度指示”应该：

- 设为RLC包头结束直到包括一个RLC SDU分段的最后一个字节之间所包含的字节数。
- 包含在它们所指示的PDU中。

“Length Indicator”的大小可能是7 bit或者15 bit。“Length Indicator”的大小上下行可以独立取值。“长度指示”的值不应该超过在11.2.4.2节和11.3.4.5节中分别为UMD和AMD PDU定义的值。

指示相同 PDU 的“Length Indicator”应该：

- 在重传的情况下不被记录。
- 与它们所指示的RLC SDU有相同的顺序。

对于AM上行和AM下行，当配置了固定RLC PDU大小时：

- 如果“AMD PDU 大小” ≤ 126 字节：
 - 应该使用 7 bit 的“Length Indicator”。
- 否则：
 - 应该使用 15 bit 的“Length Indicator”。
- 对于一个 RLC 实体，所有 AMD PDU 的“Length Indicator”的尺寸大小总是相同的。

对于AM上行和AM下行，当配置了灵活RLC PDU大小时：

- 如果“Length Indicator size”设为 7 bit：
 - 使用 7 bit 的“Length Indicator”

— 否则:

- 使用 15 bit 的 “Length Indicator”。

对于上行UM:

— 如果 “最大 UMD PDU 大小” ≤ 125 字节:

- 应该使用 7 bit 的 “Length Indicator”。

— 否则

- 应该使用 15 bit 的 “Length Indicator”。

对于下行UM:

— “Length Indicator” 取值为 “DL RLC UM LI size” 中指定的值。

对于UM:

— 在 “最大 UMD PDU 大小” 的各个修改之间, “Length Indicator” 的尺寸大小对于所有 UMD PDU 相同。

— 如果 RLC SDU 在 RLC PDU 开始处开始; 并且

— 如果在上行链路上传送该 RLC SDU; 并且

— 如果没有 “Length Indicator” 指示 RLC SDU 正好终结在上一个 RLC PDU 的结尾处, 或者差一个字节 (仅当使用 15 bit “Length Indicator” 的情况下) 填满上一个 RLC PDU。

— 如果扩展比特没有指示 UMD PDU 包含一个未被分段、级联、填充的完整的 SDU, 并且

— 如果长度指示没有指明该 RLC PDU 的第一个数据字节是 RLC SDU 的第一个数据字节而且该 RLC PDU 中的最后一个字节是相同 RLC SDU 的最后一个字节不存在, 并且

— 如果长度指示指明该 RLC PDU 的第一个数据字节是一个 SDU 的第一个字节而且相同的 RLC SDU 比正好填满 PDU 差一个字节 (仅当使用 15 bit 的长度指示时) 不存在:

- 如果使用 7 bit “Length Indicator”, 则

◆ 应该使用 “111 1100” 作为 “Length Indicator” 的值。

- 如果使用 15 bit “Length Indicator”, 则

◆ 应该使用 “111 1111 1111 1100” 作为 “Length Indicator” 的值。

— 在下行链路:

- 如果使用 7 bit “Length Indicator”, 则

◆ 接收方应准备接收值为 “111 1100” 的 “Length Indicator”;

◆ 无论值为 “111 1100” 的 “Length Indicator” 出现与否, 接收方应遵循 11.2.3 的丢弃准则。

- 如果使用 15 bit “Length Indicator”, 则

◆ 接收方应准备接收值为 “111 1111 1111 1100” 的 “Length Indicator”;

◆ 无论值为 “111 1111 1111 1100” 的 “Length Indicator” 出现与否, 接收方应遵循 11.2.3 的丢弃准则。

如果 RLC SDU 的最后一个分段正好在 PDU 结束处终结并且没有指示 RLC SDU 结尾的 “Length Indicator”, 而且 PDU 的 HE 字段未指示 AMD PDU 的最后一个字节是一个 SDU 的最后一个字节, 而且下一个 PDU 的扩展比特字段未指示 UMD PDU 包含一个未被分段、级联、填充的完整 SDU, 而且下一个 PDU 的长度指示字段未指示该 PDU 的第一个数据字节是一个 SDU 的第一个字节并且该 PDU 的最后一个字节是相

同SDU的最后一个字节，而且下一个PDU的长度指示字段没有指明该RLC PDU的第一个数据字节是一个SDU的第一个字节并且相同RLC SDU比恰好填满PDU短一个字节（仅当使用15bit长度指示时）：

— 如果使用 7bit 的 “Length Indicator”，则

- 在下一个 PDU 中应该使用值为“000 0000”的“Length Indicator”作为第一个“Length Indicator”。

— 如果使用 15bit 的 “Length Indicator”，则

- 在下一个 PDU 中应该使用值为“000 0000 0000 0000”的“Length Indicator”作为第一个“Length Indicator”。

在PDU包含一个指示RLC SDU差一个字节填满PDU的15bit “Length Indicator”的情况下，该PDU的最后一个字节应该：

— 由发送端进行填充，并且被接收端所忽略，虽然没有“Length Indicator”指示存在填充；并且

— 不使用下一个 RLC SDU 数据的第一个字节进行填充。

如果PDU中使用15bit “Length Indicator”并且RLC SDU的最后一个分段正好差一个字节填满PDU，而且没有“Length Indicator”指明RLD SDU的结束：

— 如果在下列 PDU 中使用 15bit “Length Indicator”：

- 在下一个 PDU 中应该使用值为“111 1111 1111 1011”的长度标识作为第一个“Length Indicator”。
- 当前 PDU 中剩余的一个字节应该由发送端填充，并且被接收端忽略，虽然没有“Length Indicator”指示存在填充。

— 如果为下一个 PDU 配置了 7bit “Length Indicator”：

- 如果 RLC 被配置为 UM 模式：

◆ 如果该 PDU 的扩展比特未指示 UMD PDU 包含一个未被分段、级联、填充的完整 SDU，而且该 PDU 的长度指示未指明该 PDU 的第一个数据字节是一个 SDU 的首字节并且该 PDU 的最后一个字节是相同 SDU 的最后一个字节：

— 取值为“000 0000”的长度指示将作为下一个PDU的第一个长度指示；

- 在发送之前应该将它的序号加 2。

对于UM和AM RLC：

— 如果在 RLC PDU 中使用 7bit “Length Indicator”，并且 RLC PDU 中在最后一个 RLC SDU 结束之后进行了一个或多个字节的填充：

- 通过使用值为“111 1111”的“Length Indicator”作为 PDU 的最后一个“Length Indicator”来指示使用了填充。

— 如果在 RLC PDU 中使用 15bit “Length Indicator”，并且 RLC PDU 中在最后一个 RLC SDU 结束之后进行了一个或多个字节的填充：

- 通过使用值为“111 1111 1111 1111”的“Length Indicator”作为 PDU 的最后一个“Length Indicator”来指示使用了填充。

注：即使“Length Indicator”指示在RLC PDU中使用了填充，填充的长度可能是零。

如果为UM RLC配置了扩展E-bit解释，而且RLC PDU包含一个SDU的分段，而这个分段既不是SDU的首字节又不是最后一个字节：

— 如果使用了7bit长度指示：

- 取值为“111 1110”的长度指示将被使用。
- 如果使用了15bit长度指示:
- 取值为“111 1111 1111 1110”的长度指示将被使用。

如果为UM RLC配置了扩展E-bit解释, 而且该RLC PDU的第一个数据字节是一个SDU的首字节, 该RLC PDU的最后一个字节是相同SDU的最后一个字节:

- 如果使用了7bit长度指示:
- 取值为“111 1101”的长度指示将被使用。
- 如果使用了15bit长度指示:
- 取值为“111 1111 1111 1101”的长度指示将被使用。

如果为UM RLC配置了扩展E-bit解释, 而且该RLC PDU的第一个数据字节是一个SDU的首字节, 并且相同的RLC SDU比恰好填满PDU短一个字节, 而且使用了15bit的长度指示:

- 取值为“111 1111 1111 1010”的长度指示将被使用。

如果一个“Length Indicator”还在等待发送并且没有可用的RLC SDU, 那么可能发送一个由这个“Length Indicator”、适当的填充“Length Indicator”以及填充组成的RLC PDU。

使用预先定义的“Length Indicator”值来指示填充。在下面的图标中, 按照“Length Indicator”的尺寸大小列出了保留作为特殊目的的数值。只有预先定义的“Length Indicator”才能够指示填充空间。这些值应该放在PDU的其他所有“Length Indicator”后面。

可以通过使用部分或者全部的填充空间将状态PDU附在AMD PDU上。使用预先定义的“Length Indicator”来指示piggybacked状态PDU的存在。这个“Length Indicator”替换了填充“Length Indicator”。piggybacked状态PDU应该紧跟在PDU数据后。如果只有一部分填充空间被使用, 那么将由SUFI字段NO_MORE或者ACK中的一个来指示piggybacked状态PDU的结束。因此不需要另外的“Length Indicator”来指示在AMD PDU中仍然包含填充。

如果配置了“SDU discard with explicit signalling”:

- 一个AMD PDU最多可以包含15个“Length Indicator”, 相应的指示15个SDU的结尾; 并且
- 剩余的AMD PDU空间可以用作填充或者piggybacked状态PDU。

长度: 7bit。

Bit	描述
0000000	一个RLC SDU的最后一个分段正好填满了前一个RLC PDU并且在上一个RLC PDU中没有指示RLC SDU结尾的“Length Indicator”
1111100	UMD PDU: 这个RLC PDU中的第一个字节是一个RLC SDU的第一字节。 AMD PDU: 保留(在这个协议版本中, 这种编码的PDU将被丢弃)
1111101	UMD PDU: 该RLC PDU的第一个数据字节是一个RLC SDU的第一个字节并且该RLC PDU的最后一个字节是相同RLC SDU的最后一个字节。 AMD PDU: 保留(在这个协议版本中, 这种编码的PDU将被丢弃)
1111110	AMD PDU: RLC PDU的剩余部分包括一个piggybacked STATUS PDU。 UMD PDU: 保留(在这个协议版本中, 这种编码的PDU将被丢弃)
1111111	RLC PDU的剩余部分是填充部分, 填充长度可以为0

长度: 15bit。

Bit	描述
000000000000000	一个 RLC SDU 的最后一个分段正好填满了前一个 RLC PDU, 并且在上一个 RLC PDU 中没有指示 RLC SDU 结尾的 “Length Indicator”
111111111111010	UMD PDU: 该 RLC PDU 的第一个数据字节是一个 RLC SDU 的第一个字节并且该 RLC PDU 的倒数第二个字节是相同 RLC SDU 的最后一个字节, 该 RLC PDU 剩余的一个字节忽略
111111111111011	RLC SDU 的最后一个分段差一个字节填满上一个 RLC PDU, 并且在上一个 RLC PDU 中没有指示 RLC SDU 结尾的 “Length Indicator”。上一个 RLC PDU 中剩余的一个字节被忽略
111111111111100	UMD PDU: 这个 RLC PDU 的第一个字节是一个 RLC SDU 的第一字节。AMD PDU: 保留 (在这个协议版本中, 这种编码的 PDU 将被丢弃)
111111111111101	UMD PDU: 该 RLC PDU 的第一个数据字节是一个 RLC SDU 的第一个字节并且该 RLC PDU 的最后一个字节是相同 RLC SDU 的最后一个字节。 AMD PDU: 保留 (在这个协议版本中, 这种编码的 PDU 将被丢弃)
111111111111110	AMD PDU: RLC PDU 的剩余部分包括一个 piggybacked STATUS PDU。UMD PDU: 保留 (在这个协议版本中, 这种编码的 PDU 将被丢弃)
111111111111111	RLC PDU 的剩余部分是填充, 填充长度可以为 0

9.2.2.9 数据字段

在透明、非确认和确认模式下, RLC SDU或者RLC SDU的分段被映射到这个字段。

透明模式数据:

- 对RLC SDU的长度不作8bit的整数倍的限制。
- 如果配置了“分段”：
 - 承载一个 RLC SDU 的分段的所有 RLC PDU 应该在一个 TTI 内发送。
 - 一个 TTI 中只能传输承载单一 RLC SDU 的分段的 RLC PDU。
- 否则 (没有配置“分段”)：
 - 一个 TTI 内 TMD PDU 的大小固定并且等于 RLC SDU 的大小。

非确认模式数据和确认模式数据:

- RLC SDU的长度为8bit的整数倍。
 - 如果配置了固定 RLC PDU 大小:
 - ◆ 为了完全填满数据字段以及避免不必要的填充, 除非如 9.2.2.8 节或 11.2.2.2 节特别指明, 一个 RLC SDU 的最后一个分段应该与下一个 RLC SDU 的第一个分段相连接。使用 “Length Indicator” 来指示 RLC SDU 间的边界 (见 9.2.2.8 节)。
 - 如果配置了灵活 RLC PDU 大小:
 - ◆ 在下行方向, RLC SDU 的最后一个分段可以与下一个 RLC SDU 进行级联以使 RLC PDU 的数据域填到最大尺寸, “长度指示” 域用于指示 RLC SDU 的边界 (见 9.2.2.8 节)。
 - ◆ 在上行方向, RLC SDU 的最后一个分段将与下一个 RLC SDU 的第一个分段进行级联以使 RLC PDU 至少达到最小 UL RLC PDU 尺寸, 对于 AM 模式允许级联到最大 UL AMD PDU 尺寸, 对于 UM 模式允许级联到最大 UMD PDU 尺寸。长度指示域用于指明 RLC SDU 之间的边界 (见 9.2.2.8 节)。如果待传数据的大小不足以填满一个最小尺寸的 UMD PDU, 那么也允许创建一个包含所有待传数据的 UMD PDU, 虽然其尺寸小于最小 UL RLC PDU 大小。

- 在上行方向, 如果配置了 MAC-i/is:

- ◆ 如果 UE 为随后的 TTI 预生成了一些 RLC PDU:

— 如果 UE 有足量的数据待传输, RLC PDU 数据域大小的选取应使得复用到 MAC-i/is PDU 的每个 RLC PDU 满足当前 TTI 可用授权资源 (调度或非调度) 允许传输的最大数据量。

— RLC PDU 仅在当该逻辑信道上待传的预生成的 RLC PDU 的数据量小于或等于当前 TTI 可用授权资源 (调度或非调度) 允许最大数据量的四倍时才会被生成。

- ◆ 否则如果 UE 为随后的 TTI 预生成了一些 RLC PDUs 并且 FDD 有两个激活的上行频率或者 1.28 Mcps TDD 有不只一个被调度的上行频率:

— 假设 UE 有充足的数据要传输, 那么 RLC PDU 数据域大小的选取将使得复用到 MAC-i/is PDU 的每个 RLC PDU 与下面内容的最小值相匹配:

— 对于 FDD,

— 当前 TTI 上行主载波上的当前可用的 (调度或者非调度) 授权允许传输的最大数据量。

— 当前 TTI 上行辅载波上的当前可用的 (调度) 授权允许传输的最大数据量。

— 对于 1.28 Mcps TDD,

— 当前 TTI 在每个调度上行载波上当前可用的授权允许传输的最大数据量。

— 对于 FDD, 如果这个逻辑信道的为解决的预生成 RLC PDUs 中的数据量小于或者等于下面值的最小值的 8 倍, 那么 RLC PDUs 可能只能预生成:

— 当前 TTI 上行主载波上的当前可用的 (调度或者非调度) 授权允许传输的最大数据量。

— 当前 TTI 上行辅载波上的当前可用的 (调度) 授权允许传输的最大数据量

— 对于 1.28 Mcps TDD, 如果这个逻辑信道的为解决的预生成 RLC PDUs 中的数据量小于或者等于下面值的最小值的 4N 倍, 那么 RLC PDUs 可能只能预生成:

— 当前 TTI 在每个调度上行载波上当前可用的授权允许传输的最大数据量。

注: N 是配置的上行载波数。

— 否则:

— RLC PDU 大小的选取将使得 RLC PDU 满足当前 E-TFC 选择过程为该逻辑信道请求的数据量。

— 如果配置了灵活 RLC PDU 大小, RLC PDU 数据域的最大长度为 1503 字节。

9.2.2.10 填充 (PAD)

PDU 中所有未使用的空间应该位于 PDU 的结尾部分并且被作为填充。填充有一个长度, 这样整个 PDU 有一个预先定义的总长度。

填充可能为任意值, 接收端和发送端应该对其忽略。

9.2.2.11 SUFI

使用哪一个 SUFI 字段由具体实现而定, 但是当 STATUS PDU 包含关于哪些 AMD PDU 已经被接收到以及哪些已经被发现丢失的信息时, 它将不包含关于 "Sequence Number" $\geq VR(H)$ 或 "Sequence Number" $\geq VR(MR)$ 的 AMD PDU 的信息, 也就是说还没有到达接收端的 AMD PDU 或超出接收窗口的 AMD PDU。除非在为了使用 BITMAP SUFI 而必须做的情况下, 不应该包含关于 "Sequence Number" $< VR(R)$ 的 AMD PDU 的信息。见 9.2.2.11.5 节。

长度: 比特数可变。

SUFI可以包括三个子-字段：类型信息（SUFI的类型，例如列表、比特位图、确认等）、长度信息（提供在随后的值域中的可变长度域的长度）及一个值。

图12表示了SUFI的结构。类型子字段的长度是非零的，但其他子字段的长度可以是零。

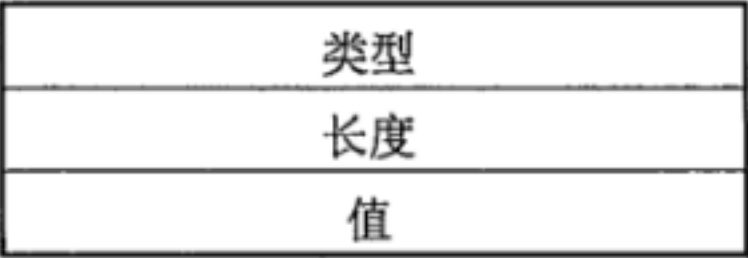


图12 SUFI 的结构

类型字段的长度为4bit，它可以是下列值中的任意一个。

比特	描述
0000	不再有数据 (NO_MORE)
0001	窗口大小 (WINDOW)
0010	确认 (ACK)
0011	列表 (LIST)
0100	比特位图 (BITMAP)
0101	相关列表 (Rlist)
0110	移动接收窗口 (MRW)
0111	移动接收窗口确认 (MRW_ACK)
1000	轮询 (POLL)
1001~1111	保留 (在这个协议版本中，这种编码的 PDU 是无效的)

是否存在“长度”和“值”以及它们的大小都取决于SUFI类型，并且在每一个SUFI中单独定义。

9.2.2.11.1 S不再有数据SUFI

“不再有数据”SUFI指示了STATUS PDU中数据部分的结尾，如图13所示。如果在STATUS PDU中包括了“不再有数据”SUFI，那么它将总是作为最后一个SUFI。该SUFI之后的所有数据都将被当作填充并且被忽略。

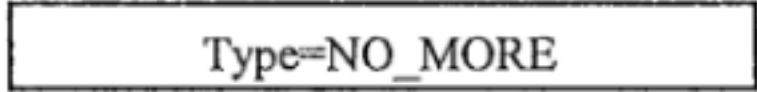


图13 状态 PDU 中的 NO_MORE 字段

9.2.2.11.2 确认SUFI

“确认”SUFI由一个类型标识字段（ACK）和一个序列号（LSN）组成，如图14所示。确认SUFI同样也指示STATUS PDU中数据部分的结尾。因此，当STATUS PDU中存在“ACK”SUFI时，就不再需要“NO_MORE”SUFI。当STATUS PDU中包含ACK SUFI时，应该将它作为最后一个SUFI。这个SUFI之后的所有数据将被认为是填充并且被忽略。

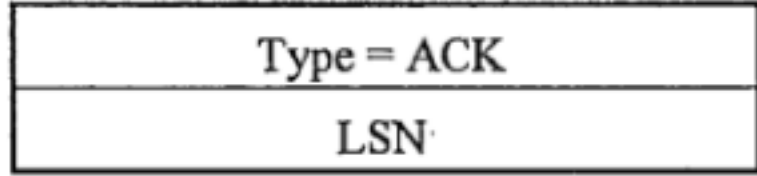


图14 状态 PDU 中的 ACK 字段

LSN

长度：12 bits

确认“Sequence Number” < LSN（最后一个序列号）的所有AMD PDU都被正确接收，这些序列号的PDU在STATUS PDU的先前部分中没有被指示为错误。其含义是，如果LSN设置为大于VR（R）的值，那么所有错误的AMD PDU应该被包含在同一个的STATUS PDU中；如果LSN设置为等于VR（R），那么

错误的AMD PDU可以被分散到多个STATUS PDU中。在发送端，如果LSN的值 \leq STATUS PDU中指示的第一个错误的值，VT (A) 将根据LSN进行更新；否则，VT (A) 将根据STATUS PDU中指示的第一个错误进行更新。VT (A) 只能够基于包含ACK SUFI (或者 MRW_ACK SUFI) 的STATUS PDU进行更新。LSN不能被设置为 $>$ VR (H) 或者 $<$ VR (R) 的值。

9.2.2.11.3 窗口尺寸SUF

窗口尺寸SUF由一个类型标识 (窗口) 和一个窗口尺寸 (WSN) 组成，如图15所示。在一个连接期间，允许接收端改变对等实体的发送窗口的大小，但是所允许的最小和最大值由上层配置所给定。接收端的接收窗口大小不发生改变。

Type = WINDOW
WSN

图15 状态 PDU 中的窗口字段

WSN

长度：12 bits。

允许发送端使用的VT (WS) 的值。WSN的范围是 $[0, 2^{12}-1]$ 。VT (WS) 的最小值是1。如果WSN为0，那么在这个协议版本中，SUF将被丢弃。一旦接收到这个SUF，变量VT (WS) 的值将被设为等于WSN。如果WSN大于Configured_Tx_Window_Size，VT (WS) 应该被设为等于Configured_Tx_Window_Size。

由于WINDOW SUFI存在丢失的可能，需要提供方案解决这一问题，具体方案参见11.7节。

9.2.2.11.4 列表SUF

列表SUF由一个类型标识字段 (LIST)、一个列表长度字段 (LENGTH) 和LENGTH数目的成对的列表组成，如图16所示。

Type = LIST
LENGTH
SN ₁
L ₁
SN ₂
L ₂
...
SN _{LENGTH}
L _{LENGTH}

图16 状态 PDU 中的列表字段

长度

长度：4 bits。

LIST类型的SUF中 (SN_i, L_i) -对的数目。值“0000”是无效的，STATUS PDU将被丢弃。

SN_i

长度：12 bits。

没有正确接收的AMD PDU的序列号。

L_i

长度： 4 bits。

序列号为SN_i的PDU之后没有正确接收的连续AMD PDU的数目。

9.2.2.11.5 位图super-field

位图super-field

位图SUF由一个类型标识字段 (BITMAP)、一个位图长度字段 (LENGTH)、一个开始顺序号 (FSN) 和一个位图组成, 如图17所示。

Type = BITMAP
LENGTH
FSN
Bitmap

图17 状态 PDU 中的 Bitmap 字段

LENGTH

长度: 4 bits。

按八位字节计算的位图的长度等于LENGTH+1, 也就是, LENGTH=“0000”意味着位图的大小是1字节; LENGTH=“1111”给出了最大的位图尺寸, 16字节。

FSN

长度: 12 bits。

位图中第一个比特的顺序号。当接收窗口尺寸小于最大RLC AM “Sequence Number” 的一半时, FSN不能够被设置为小于VR (R) -7的值。如果接收窗口尺寸大于最大RLC AM “Sequence Number” 的一半时, FSN不能够被设置为小于VR (R) 的值。

Bitmap

长度: 字节数目可变, 由LENGTH字段给出。

在位图中指示的间隔[FSN, FSN + LENGTH×8-1]中的 “Sequence Number” 字段的状态, 这里, 每个位置 (从左到右) 都可以有两个不同的值 (0和1), 分别具有下列含义 (bit_position∈[0,LENGTH×8-1]):

1: SN = (FSN + bit_position) 已经正确接收。

0: SN = (FSN + bit_position) 没有正确接收。

UE可以在发送端删除已经被收到的BITMAP SUFI 指示正确接收的AMD PDUs。

注: 发送窗口不会基于BITMAP SUFI移动, 参见9.4节。

9.2.2.11.6 相关列表SUF

相关列表SUF由一个类型标识字段 (RLIST)、一个列表长度字段 (LENGTH)、起始序列号 (FSN) 和LENGTH数目的码字 (CW) 的列表组成, 如图18所示。

Type = RLIST
LENGTH
FSN
CW ₁
CW ₂
...
CW _{LENGTH}

图18 状态 PDU 中的相关列表字段

LENGTH

长度：4 bits。

RLIST类型的SUF I中的码字（CW）数。

FSN

长度：12 bits。

RLIST中第一个错误AMD PDU的序列号，也就是说，LENGTH=“0000”表示SUF I中只有FSN。

CW

长度：4 bits。

CW由4bit组成，前三个比特是一个数的一部分，最后一个比特是状态标识，解释如下：

码字	描述
$X_1X_2X_3\ 0$	数的下三个比特是 $X_1X_2X_3$ 并且数在下一个 CW 中继续。在这个 CW 中，最高有效位是 X_1
$X_1X_2X_3\ 1$	数的下三个比特是 $X_1X_2X_3$ 并且数被终结。在这个 CW 中，最高有效位是 X_1 。这是数中的最高有效码字

作为默认情况，由CW给出的数代表从先前指示了错误的AMD PDU到一个错误AMD PDU的间隔，且包括下一个错误的AMD PDU。

在这里定义了一个特殊的CW值：

000 1 “错误突发标识”

错误突发标识表示下一个CW将代表连续错误AMD PDU的数目（不算已经指示了的错误位置）。在一个突发中的错误数目用XXX 1进行终结之后，下一个码字在缺省情况下，又将成为到下一个错误的间隔的最低有效位（LSB）

9.2.2.11.7 移动接收窗口确认SUF I

“移动接收窗口确认” SUF I用来确认MRW_SUF I的接收。它的格式如图19所示。

Type=MRW_ACK
N
SN_ACK

图19 状态 PDU 中的 MRW_ACK 字段

N

长度：4 bits。

如果SN_ACK字段等于SN_MRWLENGTH字段，那么N字段应该被设为等于接收到的MRW SUF I中的NLENGTH字段。否则，N被设为0。

通过这个字段与SN_ACK字段相结合，可以确定MRW_ACK是否对上一个被传送的MRW_SUF I有响应。

SN_ACK

长度：12bits。

SN_ACK字段用来指示在接收到MRW SUF I后VR（R）的更新值。通过这个字段与N字段的相结合，可以确定MRW_ACK是否对上一个被传送的MRW_SUF I有响应。

9.2.2.11.8 移动接收窗口（MRW）SUF I

“移动接收窗口” SUF I用来请求接收端移动它的接收窗口，并且可选的，作为发送端丢弃一个RLC SDU的结果，指示被丢弃RLC SDU的集合，格式如图20所示。

Type = MRW
LENGTH
SN_MRW ₁
SN_MRW ₂
...
SN_MRW _{LENGTH}
N _{LENGTH}

图20 状态 PDU 中的 MRW 字段

LENGTH

长度：4 bits

在MRW类型的SUFi中的SN_MRW字段的数量。

“0001”到“1111”分别指示1到15的SN_MRW_i。“0000”指示只有一个SN_MRW_i在使用，并且接收端将要丢弃的RLC SDU扩展到超出了发送端配置的发送窗口。

SN_MRW_i

长度：12bits。

在配置了“Send MRW”的情况下，应该使用一个SN_MRW_i来表示每一个被丢弃的RLC SDU的结尾，也就是说，SN_MRW_i字段的数目应该等于被该MRW SUFI丢弃的RLC SDU的数目。在没有配置“Send MRW”的情况下，使用SN_MRW_i字段来指示在接收端被丢弃的最后一个RLC SDU的结尾，另外的可能可选的用于指示其他被丢弃的RLC SDU的结尾。SN_MRW_i表示AMD PDU 的“Sequence Number”，该AMD PDU包含在接收端被丢弃的第i个RLC SDU的“Length Indicator”（除了在N_{LENGTH}=0时的SN_MRW_{LENGTH}，参见N_{LENGTH}的定义）。SN_MRW_i的顺序与它们所指示的RLC SDU的顺序相同。

此外，SN_MRW_{LENGTH}还请求接收端丢弃所有未成功接收的具有分段SDUs的或者“Length Indicators”指示SDUs结束的AMD PDU的“Sequence Number” < SN_MRW_{LENGTH}的SDUs，并相应的移动接收窗口。另外，当N_{LENGTH}>0时，接收端必须丢弃所有未成功接收的具有分段SDUs的或者“Length Indicators”指示SDUs结束的AMD PDU的且有PDU分段落在的序号等于“Sequence Number” < SN_MRW_{LENGTH}中的SDU及并且包含第N_{LENGTH}个“Length Indicator”指示的序号为 SN_MRW_{LENGTH}的AMD PDU中相应数据字节的SDUs。

N_{LENGTH}

长度：4 bits。

N_{LENGTH}与SN_MRW_{LENGTH}共同使用来指示在接收端被丢弃的最后一个RLC SDU的结尾。

N_{LENGTH}指示“序号”SN_MRW_{LENGTH}的AMD PDU中哪一个“Length Indicator”与在接收端被丢弃的最后一个RLC SDU相对应。N_{LENGTH} = 0指示最后一个RLC SDU的结束在“Sequence Number” SN_MRW_{LENGTH}-1的AMD PDU中指示，并且“Sequence Number” SN_MRW_{LENGTH}的AMD PDU中的第一个数据字节是将被重传的第一个数据字节。

9.2.2.11.9 轮询（POLL）SUFi

轮询超字段用于向接收端请求一个状态报告（一个或者多个STATUS PDU）。下行轮询超字段仅当下行链路配置了可变RLC大小时使用，上行轮询超字段仅当上行链路配置了可变RLC大小时使用。格式如图21所示。

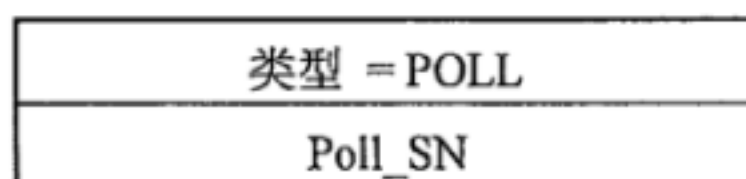


图21 状态 PDU 中的 POLL 字段

Poll_SN

长度: 12bit。

当配置了轮询时, Poll_SN字段将被设置为VT(S)-1。

9.2.2.12 保留2 (R2)

长度: 1 bit。

Piggybacked STATUS PDU中的这个比特用于使Piggybacked STATUS PDU在长度上成为8bit的整数倍, 基于这个目的, 它被设为0。否则PDU将被认为是无效的, 并且因此在这个版本的协议中将被丢弃。

9.2.2.13 复位序号 (RSN)

长度: 1 bit。

这个字段用于指示传送的RESET PDU的序号。如果这个RESET PDU是一个原始RESET PDU的重传, 那么重传的RESET PDU和原始的RESET PDU具有相同的RSN值。否则, 它将等于下一个RSN值。这个字段的初始值是0。当RLC重建时, 该字段的值被重新初始化。当RLC复位时这个字段不会重新初始化。

9.2.2.14 超帧号标识 (HFNI)

长度: 20 bit。

这个字段用于向对等实体指示超帧号 (HFN)。借助于这个字段, 可以实现UE和UTRAN中超帧号的同步。

9.3 协议状态

本节中所描述的内容仅为了支持RLC 协议状态的定义, 不对协议的实现进行定义或者限制。

9.3.1 透明模式实体的状态模型

透明模式RLC实体的状态模型 (传送和接收) 如图22所示。透明模式实体可以处于下列状态之一。

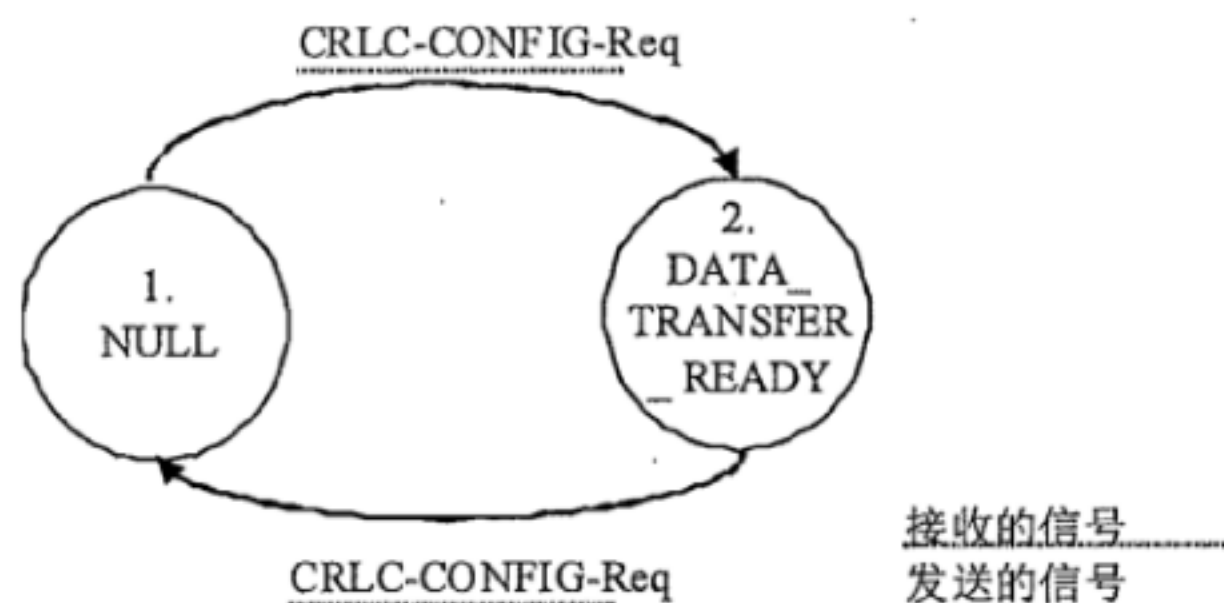


图22 透明模式实体的状态模型

9.3.1.1 NULL状态

在NULL状态下, RLC实体不存在, 因此不可能通过其传送任何数据。

一旦从上层接收到一个CRLC-CONFIG-Req指示建立, RLC实体:

- 被产生; 并且
- 进入DATA_TRANSFER_READY状态。

9.3.1.2 DATA_TRANSFER_READY状态

在DATA_TRANSFER_READY状态时, 可以根据11.1节在两个实体之间进行透明模式数据的交换。
当从上层接收到一个CRLC-CONFIG-Req指示释放, RLC实体:

- 进入NULL状态; 并且
- 认为是已被终结。

9.3.2 非确认模式实体的状态模型

非确认模式RLC实体(发送和接收)的状态模型如图23所示。一个非确认模式实体可以处于以下状态之一。

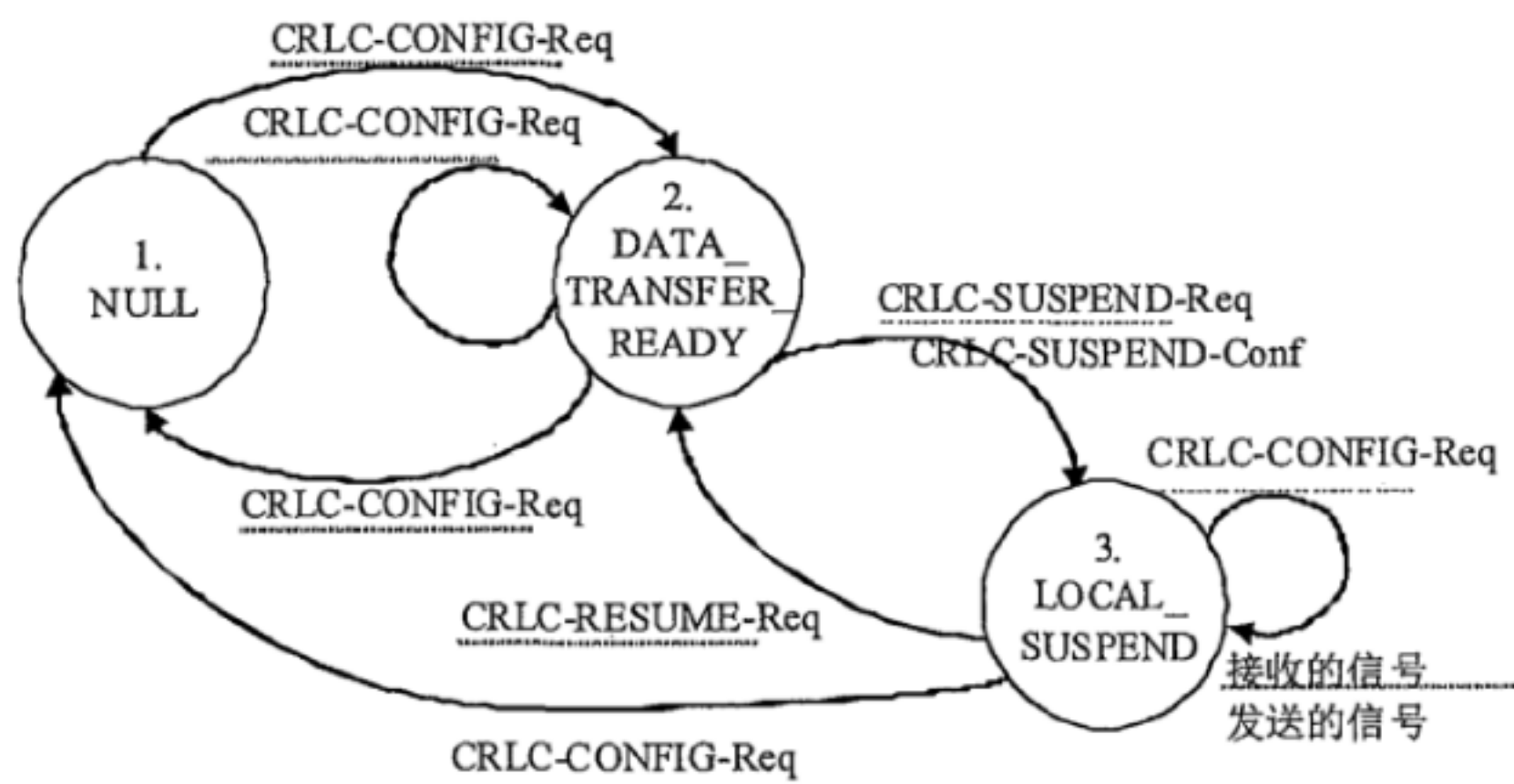


图23 非确认模式实体的状态模型

9.3.2.1 NULL状态

在NULL状态下, RLC实体不存在, 因此不可能通过它来传送任何数据。

当从上层接收到一个CRLC-CONFIG-Req指示建立, RLC实体:

- 被产生; 并且
- 进入DATA_TRANSFER_READY状态。

9.3.2.2 DATA_TRANSFER_READY状态

在DATA_TRANSFER_READY状态下, 可以根据11.2节在实体之间进行非确认模式数据的交换。

当从上层接收到一个CRLC-CONFIG-Req指示释放, RLC实体:

- 进入NULL状态; 并且
- 认为是已被终结。

当从上层接收到一个CRLC-CONFIG-Req指示修改, RLC实体:

- 处于DATA_TRANSFER_READY状态;
- 仅修改上层指示的协议参数和定时器。

当从上层接收到一个CRLC-SUSPEND-Req, RLC实体:

- 进入LOCAL_SUSPEND状态。

9.3.2.3 LOCAL_SUSPEND状态

在LOCAL_SUSPEND状态, RLC实体被挂起, 也就是说, 它不发送“Sequence Number”大于或者等于某个特定值的UMD PDU(见9.7.5节)。

当从上层接收到一个CRLC-CONFIG-Req指示释放时候, RLC实体:

- 进入NULL状态; 并且

— 被认为结束。

当从上层接收到一个CRLC-RESUME-Req, RLC实体:

- 进入DATA_TRANSFER_READY状态; 并且
- 恢复数据的传输。

当从上层接收到一个CRLC-CONFIG-Req指示修改, RLC实体:

- 处于LOCAL_SUSPEND状态;
- 仅修改上层指示的协议参数和定时器。

9.3.3 确认模式实体的状态模型

确认模式RLC实体（发送和接收）的状态模型如图24所示。一个确认模式实体可以处于以下状态之一。

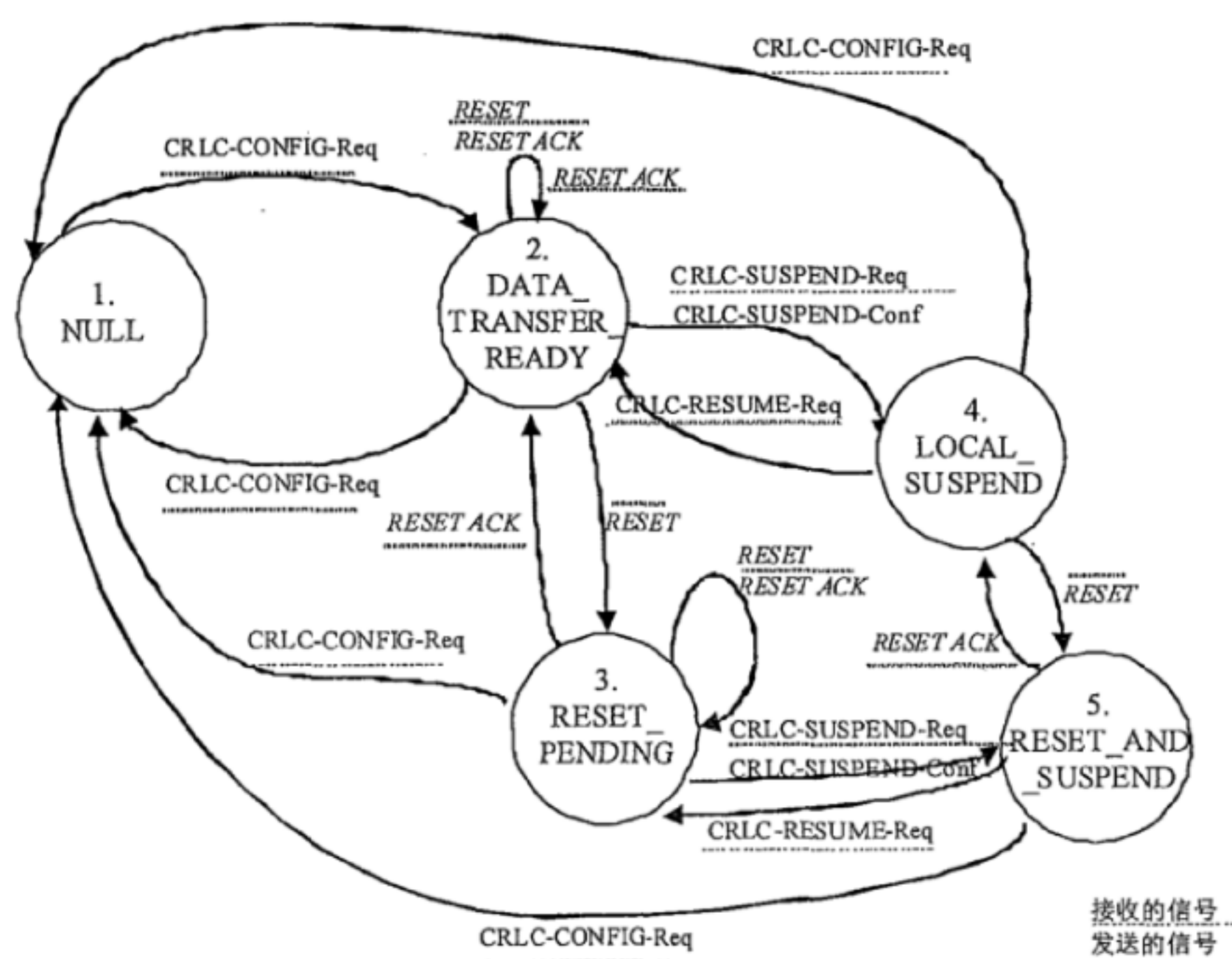


图24 确认模式实体的状态模型

9.3.3.1 NULL状态

在NULL状态下, RLC实体不存在, 因此不能通过它传送任何数据。

当从上层接收到一个CRLC-CONFIG-Req指示建立, RLC实体:

- 被产生; 并且
- 进入DATA_TRANSFER_READY状态。

9.3.3.2 DATA_TRANSFER_READY状态

在DATA_TRANSFER_READY状态下, 可以根据11.3节在实体之间进行确认模式数据的交换。

当从上层接收到一个CRLC-CONFIG-Req指示释放, RLC实体:

- 进入NULL状态; 并且
- 认为已被终结。

当检测到11.4.2节中描述的RLC复位过程的初始条件时, RLC实体:

- 初始化 RLC 复位过程（见 11.4 节）；并且
- 进入 RESET_PENDING 状态。

当接收到一个 RESET PDU 时，RLC 实体根据 11.4.3 节进行响应。

当接收到一个 RESET ACK PDU 时，RLC 实体不作任何动作。

当从上层接收到一个 CRLC-SUSPEND-Req 时，RLC 实体被挂起并且进入 LOCAL_SUSPEND 状态。

9.3.3.3 RESET_PENDING 状态

在 RESET_PENDING 状态下，实体等待对等实体的响应，此时在实体之间不能进行数据的交换。

当从上层接收到一个 CRLC-CONFIG-Req 指示释放，RLC 实体：

- 进入 NULL 状态；并且
- 认为是已经被终结。

当接收到具有与相对应的 RESET PDU 相同 RSN 值的 RESET ACK PDU 时，RLC 实体：

- 根据 11.4.4 节进行动作；并且
- 进入 DATA_TRANSFER_READY 状态。

当接收到具有与相对应的 RESET PDU 不同 RSN 值的 RESET ACK PDU 时，RLC 实体：

- 丢弃 RESET ACK PDU（见 11.4.4 节）；并且
- 处于 RESET_PENDING 状态。

当接收到 RESET PDU 时，RLC 实体：

- 根据 11.4.3 节进行响应；并且
- 处于 RESET_PENDING 状态。

当从上层接收到 CRLC-SUSPEND-Req 时，RLC 实体：

- 进入 RESET_AND_SUSPEND 状态。

9.3.3.4 LOCAL_SUSPEND 状态

在 LOCAL_SUSPEND 状态时，RLC 实体被挂起，也就是说，它不发送“Sequence Number”大于或者等于某个特定值的 AMD PDU（见 9.7.5）。

当从上层接收到 CRLC-RESUME-Req 时，RLC 实体：

- 恢复数据的传输；并且
- 进入 DATA_TRANSFER_READY 状态。

当从上层接收到 CRLC-CONFIG-Req 指示释放时，RLC 实体：

- 进入 NULL 状态；并且
- 认为已经被终结。

当检测到 11.4.2 节中描述的 RLC 复位过程的初始条件时，RLC 实体：

- 初始化 RLC 复位过程（见 11.4 节）；并且
- 进入 RESET_AND_SUSPEND 状态。

9.3.3.5 RESET_AND_SUSPEND 状态

在 RESET_AND_SUSPEND 状态时，实体等待对等实体的响应或者是来自上层的原语（CRLC-RESUME-Req），此时在实体间无法进行数据的交互。

当从上层接收到 CRLC-CONFIG-Req 指示释放时，RLC 实体：

- 进入 NULL 状态；并且
- 认为是已经被终结。

当接收到与相应的RESET PDU具有相同RSN值的RESET ACK PDU时，RLC实体：

- 根据 11.4.4 进行响应；并且
- 进入 LOCAL_SUSPEND 状态。

当从上层接收到CRLC-RESUME-Req时，RLC实体：

- 恢复，也就是，释放挂起限制；并且
- 进入 RESET_PENDING 状态。

9.4 状态变量

本节中定义的状态变量是标准化的。

本节中描述了用于定义点对点协议而在AM和UM中使用的状态变量。所有的状态变量都是非负整数。UMD和AMD PDU模整数序列号(SN)进行编号，对于AM，取模后的编号范围0到 $2^{12}-1$ ；对于UM，取模后的编号范围0到 2^7-1 。本文中描述的所有与VT(S)，VT(A)，VT(MS)，VR(R)，VR(H)和VR(MR)相关的算术运算都受AM模数的影响，所有与VT(US)，VR(US)，VR(UDH)，VR(UDR)，VR(UOH)和VR(UM)相关的算术运算都受到UM模数的影响。所有状态变量或者序列号值的算术比较运算都需要用到模数基。首先所有相关的数值减去这个模数基（在适当的范围内），然后进行一个绝对值的比较。在发送端，VT(A)和VT(US)分别作为AM和UM的模数基；在接收端，VR(R)作为AM模式下的模数基，VR(US)作为UM模式下没有配置SDU乱序递交和重复避免与重排序功能时的模数基；若配置了SDU乱序递交功能， $(VR(UOH) - OSD_Window_Size + 1)$ 将作为UM接收模式的模数基；若配置了和重复避免与重排序功能， $(VR(UDH) - DAR_Window_Size + 1)$ 将作为UM接收模式的模数基。

在发送端，RLC维护下列状态变量：

a) VT(S) ——发送状态变量

这个状态变量包含了下一次传送的 AMD PDU 的“Sequence Number”，这里指的是第一次传送的 AMD PDU（也就是，不包含重传的 PDU）。在前面提到的 AMD PDU 或者一个包括 $SN_MRW_LENGTH > VT(S)$ （见 11.6 节）的 MRW SUFI 传送之后，它将被更新。该变量的初始值为 0。

b) VT(A) ——确认状态变量

这个状态变量包含了最后一个被按顺序确认的 AMD PDU 的“Sequence Number”后的“Sequence Number”，其形成了可接收确认的发送窗口的下届。当接收到一个包括 ACK（见 9.2.2.11.2 小节）和/或者 MRW_ACK SUFI 的状态 PDU 后，VT(A) 将基于它进行更新。

该变量的初始值为 0。为了进行协议的初始化，这个值应该被认为是最后一个被按顺序确认的 AMD PDU 后的第一个“Sequence Number”。

c) VT(DAT)

该状态变量对一个 AMD PDU 被安排发送的次数进行计数。对每个 PDU 都有一个 VT(DAT)，并且 AMD PDU 每被安排发送一次，或者每次有一个包含 POLL SUFI 的 STATUS PDU 被安排发送而且相应 AMD PDU 的序列号等于 $VT(S) - 1$ 时，相应的变量值就加增加。

该变量的初始值为 0。

d) VT (MS) ——最大发送状态变量

这个状态变量包含了可以被对等接收端拒绝的第一个 AMD PDU 的 “Sequence Number”， $VT (MS) = VT (A) + VT (WS)$ 。该值表示发送窗口的上界。除非 $VT (S) \geq VT (MS)$ ，发送端不发送 “Sequence Number” $\geq VT (MS)$ 的 AMD PDU。在那种情况下，“Sequence Number” 等于 $VT (S) - 1$ 的 AMD PDU 也可以被发送。当 VT (A) 或 VT (WS) 更新时，VT (MS) 也将被更新。

这个变量的初始值等于 Configured_Tx_Window_size。

e) VT (US) ——UM 数据状态变量

该状态变量给出了下一个要传送的 UMD PDU 的序列号。UMD PDU 每传送一次，该变量就加 1。

该变量的初始值为 0。

注：在 UTRAN 侧，该变量的初始值可以不等于 0。

f) VT (PDU)

在配置了 “poll every Poll_PDU PDU” 轮询触发器的情况下，使用这个状态变量。每个 AMD PDU 被传送时（包括新的和重传的 AMD PDU），该变量就加 1。当它等于 Poll_PDU 时，发送一个新的轮询送，并且变量被设为 0。

该变量的初始值为 0。

g) VT (SDU)

在配置了 “poll every Poll_PDU PDU” 轮询触发器的情况下，使用这个状态变量。每当承载一个特定 SDU 的第一个分段的 AMD PDU 被首次发送后，与该 SDU 相对应的变量值增加 1。当它等于 Poll_SDU 时，发送一个新的轮询并且状态变量被设为 0。当包含一个 RLC SDU 的最后一个分段的 AMD PDU 第一次发送时，“Polling bit” 被设为 1（通过长度指示指明 SDU 结束或者通过 HE 字段的特殊值指示）。

该变量的初始值为 0。

h) VT (RST) ——复位状态变量

这个状态变量用来计数在复位过程完成以前，一个复位 PDU 被安排传送的次数。根据 11.4.2 和 11.4.5.1 节复位 PDU 每被安排传送一次，VT (RST) 增加 1。仅当接收到 RESET ACK PDU 时，VT (RST) 被复位（也就是说当对等 RLC 实体初始化 RLC 复位时，VT (RST) 不进行复位），除非 9.7.7 节说明的其他情况出现。

该变量的初始值为 0。

i) VT (MRW) ——MRW 命令发送状态变量

这个状态变量用来计数 MRW 命令被发送的次数。每当定时器 Timer_MRW 超时，VT (MRW) 就增加 1。当明确的信令过程指示的 SDU 丢弃终止时，VT (MRW) 被复位。这个变量的初始值是 0。

j) VT (WS) ——发送窗口大小状态变量

这个状态变量包含了发送窗口的大小。当发送端收到一个包含 WINDOW SUFI 的 STATUS PDU 时，VT (WS) 被设为等于 WSN 字段。

这个变量的初始值是 Configured_Tx_Window_size。

在接收端，RLC 维护以下状态变量：

a) VR (R) ——接收状态变量

这个状态变量包含按顺序接收到的最后一个 AMD PDU 的下一个“Sequence Number”。当接收到具有与 VR(R) 相等的“Sequence Number”的 AMD PDU 时, 它将被更新。

该变量的初始值是 0。为了协议的初始化, 这个值应该被认为是按顺序接收到的最后一个 AMD PDU 后的第一个“Sequence Number”。

b) VR(H) ——最高期望状态变量

这个状态变量包含接收到的或者确定丢失的最高 AMD PDU “Sequence Number” 后的序列号。当接收到“Sequence Number” x 或者收到的轮询超字段的 POLL_SN= x 满足 $VR(H) \leq x < VR(MR)$ 的 AMD PDU 时, 这个状态变量应该被设为 $x+1$ 。

c) VR(MR) ——最大可接收状态变量

这个状态变量包含应该被接收端拒绝的第一个 AMD PDU 的“Sequence Number”, $VR(MR) = VR(R) + \text{Configured_Rx_Window_Size}$ 。

d) VR(US) ——接收者发送顺序状态变量

这个状态变量仅在未配置乱序 SDU 传递时使用, 该状态变量包含接收缓冲区接收到的最后一个 UMD PDU 后的“Sequence Number”(参见第 4 章图 3 与图 4)。当接收缓冲区接收到“Sequence Number”等于 x 的 UMD PDU 时, 该状态变量被设为 $x+1$ 。

该变量的初始值为 0。

e) VR(UOH) ——UM SDU 乱序递交的最高接收状态变量

这个状态变量包含接收到的最近一次收到的 UMD PDU 的序号。

该变量初始值设置见 11.2.3.2 节。

f) VR(UDR) ——UM 重复避免与重排序发送状态变量

这个状态变量包含期望顺序接收到的下一个 UMD PDU 的序号, 其取值设置见 9.7.10 节。

该变量初始值设置见 9.7.10 节。

g) VR(UDH) ——UM 重复避免与重排序最高接收状态变量

这个状态变量包含由重复避免与重排序功能接收到的编号最高的 UMD PDU 的序号。

该变量初始值设置见 9.7.10 节。

h) VR(UDT) ——UM 重复避免与重排序定时器状态变量

这个状态变量包含当定时器 Timer_DAR 运行时与该定时器关联的 UMD PDU 的序号, 其取值设置见 9.7.10 节。

i) VR(UM) ——最大允许接收状态变量

这个状态变量包含将被接收端拒绝的第一个 UMD PDU 的序号, $VR(UM) = VR(US) + \text{Configured_Rx_Window_Size}$, 该状态变量仅当高层配置了乱序接收时才使用。

9.5 定时器

本节中定义的定时器是标准化的。从被启动时开始, 直到它们超时或者被停止为止, 定时器是激活的。

a) Timer_Poll

只上层进行了相关配置时, 才使用该定时器。定时器的值由上层信令指示。在 UE 端, 当下层指示一个包含轮询信息的 AMD PDU 被成功或者不成功发送时, 该定时器启动(或重启)。在 UTRAN 端, 当一

个轮询被发送给下层时, 该定时器启动。如果 x 是轮询信息被发送给下层后状态变量 VT(S) 的值, 那么一旦接收到下列消息, 定时器应该被停止:

- 对于所有 “Sequence Number” 小于或者等于 $x-1$ 的 AMD PDU 的正面确认; 或者
- 对于 “Sequence Number” = $x-1$ 的 AMD PDU 的反面确认。

如果定时器超时, 并且没有收到满足上述条件的 STATUS PDU。

- 应该再一次向接收端发起轮询;
- 定时器应该重新启动; 并且
- VT(S) 的新值应该被保存。

如果在定时器激活期间发送了一个新的轮询信息, 那么应该在上面定义的时间重新启动定时器, 并且 VT(S) 的值应该被保存。

b) Timer_Poll_Prohibit

只有在上层进行了相关配置的情况下才使用该定时器。它用于在一段时间内禁止轮询信息的传输。定时器的值由上层信令指示。

在 UE 端, 当下层指示一个包含轮询信息的 AMD PDU 被成功或者不成功发送时, 该定时器启动。在 UTRAN 端, 当一个轮询被发送给下层时, 该定时器启动。

从轮询被触发开始, 直到定时器超时, 轮询被禁止。如果在轮询禁止期间, 有另一个轮询被触发, 那么它应该被延时到定时器超时后发送 (见 9.7.1 节)。即使期间有多个轮询被触发, 当定时器超时的時候也只发送一个轮询。这个定时器不应该受到 STATUS PDU 的接收的影响。

当上层没有配置 Timer_Poll_Prohibit 时, 轮询从不被禁止。

c) 无效

d) Timer_Discard

只有在上层配置了基于定时器的 SDU 丢弃时, 才使用该定时器。定时器的值由上层信令指示。在发送端, 当从上层接收到一个 SDU 后, 一个新的定时器被启动。

UM/TM 时, 如果在相应的 SDU 被发送给下层之前定时器超时, 那么对 11.2.4.3 和 11.1.4.2 节中定义的 “没有明确信令指示情况下的 SDU 丢弃” 进行初始化。在 AM 时, 如果在相应的 SDU 被确认之前定时器超时, 那么对 11.6 节中定义的 “没有明确信令指示情况下的 SDU 丢弃” 进行初始化。

e) Timer_Poll_Periodic

只有在上层配置了 “基于定时器轮询” 时, 才使用该定时器。定时器的值由上层信令指示。当 RLC 实体建立时, 该定时器启动。当定时器超时后, RLC 实体:

- 重新启动定时器
- 如果存在 AMD PDU 可用于传输或者重传 (还没有被确认), 那么:
 - ◆ 触发一个轮询。

f) Timer_Status_Prohibit

— 只有上层进行了相应配置的情况下, 才使用该定时器。它用于禁止接收端发送连续的确认状态报告。如果状态报告包括 SUFIs、LIST、BITMAP、RLIST 或 ACK 中的任何一个的话, 那么它就是一个确认状态报告。定时器的值由上层信令指示。

— 在 UE 端, 当下层指示一个确认状态报告的最后一个 STATUS PDU 被成功或者不成功发送时, 该定时器启动。在 UTRAN 端, 当一个确认状态报告的最后一个 STATUS PDU 被发送给下层时, 该定时器启动。

— 从确认状态报告被触发开始, 直到 Timer_Status_Prohibit timer 超时, 确认被禁止。如果在确认被禁止的时候有另一个这样的状态报告被触发, 那么它应该被推迟到定时器超时后发送 (见 9.7.2 节)。在此期间, 状态报告可能被更改。SUFIs MRW, MRW_ACK, WINDOW or NO_MORE 的发送不被限制。

— 在上层没有配置 Timer_Status_Prohibit 的情况下, 确认不被禁止。

g) Timer_Status_Periodic

只有在上层配置了基于定时器的状态发送的情况下, 才使用该定时器。

当 RLC 实体建立时, 该定时器启动。当定时器超时, 触发一个状态报告的发送并且定时器重新启动。该定时器可以被上层阻塞。当上层指示定时器不再被阻塞时, 定时器应该重新启动。

h) Timer_RST

该定时器用来处理对等 RLC 实体的 RESET PDU 的丢失, 或者是来自对等实体的 RESET ACK PDU 的丢失。定时器的初始值由上层信令指示。

在 UE 端, 当下层指示 RESET PDU 被成功或者不成功发送时, 该定时器启动 (或重启)。在 UTRAN 端, 当 RESET PDU 被发送给下层时, 该定时器启动。

只有在接收到 RESET ACK PDU (与 RESET PDU 具有相同的 RSN 值) 时, Timer_RST 才被停止, 也就是说当对等 RLC 实体初始化一个 RLC 复位时, 该定时器不应该被停止。如果定时器超时, 应该重新发送 RESET PDU。

i) Timer_MRW.

该定时器用来触发一个包含 MRW SUFI 字段的状态报告的重传。定时器的初始值由上层指示。

在 UE 端, 当下层指示包含 MRW SUFI 的 STATUS PDU 被成功或者不成功发送时, 该定时器启动 (或重启)。在 UTRAN 端, 当包含 MRW SUFI 的 STATUS PDU 被发送给下层时, 该定时器启动。

每当定时器超时, 就重传 MRW SUFI 并且重新启动定时器。当满足某一个“明确信令指示的 SDU 丢弃”的终止条件时 (见 11.6.4 节), 定时器被停止。

j) Timer_OSD.

这个定时器用于 UM 模式下的 SDU 乱序递交功能。使用该定时器触发存储的 PDU 的删除。

定时器的启动和停止条件参见 11.2.3.2 节。

k) Timer_DAR

这个定时器用于 UM 模式下的重复避免与重排序功能, 用来触发将 PDU 传输给下一个按序传输的 UM RLC 接收功能。

定时器的启动和停止条件参见 9.7.10 节。

9.6 协议参数

本节中所定义的行为是标准化的。本节中所定义的状态变量的值由上层信令指示。

a) MaxDAT

一个 AMD PDU 的最大传输次数等于 MaxDAT-1。该协议参数是状态变量 VT (DAT) 的上限。当 VT (DAT) 的值达到 MaxDAT 时, 将根据上层的配置开始 RLC RESET 过程或 SDU 丢弃过程的初始化。

b) Poll_PDU

该协议参数指示了在上层配置了“polling every Poll_PDU PDU”的情况下,发送端向接收端轮询的频率。它是状态变量 VT(PDU) 的上限,当 VT(PDU) 等于 Poll_PDU 时,向对等实体发送一个轮询。

c) Poll_SDU

该协议参数指示了在上层配置了“polling every Poll_PDU PDU”的情况下,发送端向接收端轮询的频率。它是状态变量 VT(SDU) 的上限,当 VT(SDU) 等于 Poll_SDU 时,向对等实体发送一个轮询。

d) Poll_Window

该协议参数指示了在上层配置了“基于窗口的轮询”的情况下,发送端在什么时间向接收端轮询。当: $J \geq \text{Poll_Window}$ 时,对于每一个 AMD PDU 触发一个轮询,这里 J 是由以下公式定义的发送窗口百分比。

$$J = \frac{(4096 + VT(S) + 1 - VT(A)) \bmod 4096}{VT(WS)} \times 100$$

其中常数 4096 是 9.4 节中描述的 AM 的模数,VT(S) 是 AMD PDU 发送到下层之前的变量值。

e) MaxRST

RESET PDU 传输的最大次数等于 MaxRST-1。该协议参数是状态变量 VT(RST) 的上限。当 VT(RST) 等于 MaxRST 时,将向上层指示不可恢复的错误。

f) Configured_Tx_Window_Size

这个协议参数指示了允许的最大发送窗口尺寸和 VT(WS) 状态变量的值。

g) Configured_Rx_Window_Size

这个协议参数指示了接收窗口尺寸。

h) MaxMRW

MRW 命令的最大传输次数等于 MaxMRW。这个协议参数是状态变量 VT(MRW) 的上限。当 VT(MRW) 等于 MaxMRW 时,开始 RLC RESET 过程的初始化。

i) OSD_Window_Size

这个协议参数指示了 SDU 乱序递交存储窗的大小。

j) DAR_Window_Size

这个协议参数指示了重复避免与重排序窗的大小。

9.7 指定功能

本节中所定义的功能是标准化的。

9.7.1 确认模式的轮询功能

轮询功能用于发送端向对等RLC实体请求状态报告。AMD PDU中的“轮询比特”或者POLL SUFI指示了轮询请求。对于轮询的初始化存在多个触发器。对于每一个RLC实体,由上层配置具体使用哪一个触发器。可以配置以下的触发器。

a) 缓冲区中最后的 PDU

当向下层传送第一次被发送的 AMD PDU 时,发送端应该:

— 如果该 AMD PDU 是被安排发送的最后一个 AMD PDU,根据 11.3.2 节(也就是说,没有从上层接收到的,还需要被分段到 AMD PDU 中的数据了);或者

— 如果该 AMD PDU 是允许发送的最后一个 AMD PDU, 根据 11.3.2.2 节, 那么:

- 触发一个对于该 AMD PDU 的轮询。

b) 重传缓冲区中的最后一个 PDU

当向下层发送一个重传 AMD PDU 时, 发送端应该:

— 如果该 AMD PDU 是最后一个被安排重传的 AMD PDU, 根据 11.3.2 节; 或者

— 如果该 AMD PDU 是根据 11.3.2.2 节允许发送的, 最后一个被安排重传的 AMD PDU, 那么;

- 触发一个该 AMD PDU 的轮询。

c) 轮询定时器

定时器 Timer_Poll 根据 9.5 节 a) 启动和停止。当定时器 Timer_Poll 超时, 发送端触发轮询功能。

d) 每个 Poll_PDU PDU

发送端对于每个 Poll_PDU PDU 触发轮询功能。重传和新的 AMD PDU 都将被计数。

e) Every Poll_SDU SDU

发送端对于每个 Poll_SDU SDU 触发轮询功能。对于包含一个 RLC SDU 的最后一个分段的 AMD PDU 的首次传输 (通过 "Length Indicator" 指示 RLC SDU 的结束或者通过 HE 字段的特殊取值指示), 触发轮询。

f) 基于窗口

当 9.6 节 d) ("Poll_Window") 中描述的条件满足时, 发送端触发轮询功能。

g) 基于定时器

发送端周期性触发轮询功能。

UTRAN 应该配置 RLC 以避免死锁。

发送端使用轮询禁止功能来推迟轮询功能的初始化。轮询禁止功能的使用由上层配置。轮询禁止功能包括启动定时器 Timer_Poll_Prohibit (根据 9.5 节 b)), 以及根据以下规则延迟轮询功能:

当轮询功能被触发时, 发送端应该:

— 如果轮询没有被禁止 (见 9.5 节 b)); 并且

— 如果有一个或者多个 AMD PDU 将要发送, 或者存在未被接收端确认的 AMD PDU:

- 根据 11.3.2.1.1 设置 "轮询比特" 来初始化轮询功能。

— 否则 (没有将要发送的 PDU 并且所有的 PDU 都已经被确认):

- 不初始化轮询功能。

当定时器 Timer_Poll_Prohibit 超时, 发送端应该:

— 如果在定时器 Timer_Poll_Prohibit 激活期间, 轮询功能至少被触发了一次; 并且

— 如果有一个或者多个 AMD PDU 将要发送, 或者存在未被接收端确认的 AMD PDU:

- 根据 11.3.2.1.1 节设置 "轮询比特" 初始化一次轮询功能。

— 否则 (如果没有将要发送的 PDU 并且所有的 PDU 都已经被确认):

- 不初始化轮询功能。

注1: 在下行方向, 当配置了下行灵活 RLC PDU 大小时 UTRAN 可以根据 9.2.2.11.9 节通过装配 POLL SUFI 来发起轮询功能。

注2: 在上行方向, 当配置了上行灵活 RLC PDU 大小时 UE 可以根据 9.2.2.11.9 节通过装配 POLL SUFI 来发起轮询功能。

9.7.2 确认模式的状态传送

为了通知发送端哪个PDU收到哪个PDU没有收到,接收端向发送端发送状态报告。每个状态报告包含一个或多个STATUS PDU。当收到轮询请求时,接收端触发一次状态报告的发送。另外上层还可以配置下列发送状态报告的触发条件:

a) 检测到 PDU 的丢失

如果接收端检测到一个或多个 AMD PDU 丢失,那么它触发一次向发送端的状态报告的发送。

b) 基于定时器的状态报告传输

接收端周期性地触发向发送端的状态报告的发送。时间周期由定时器 `Timer_Status_Periodic` 控制,根据 9.5 节 g)。当上层配置了“周期性状态阻塞”时,触发器不应该被激活。

c) 无效

有两种功能可以禁止接收端发送包含SUFIs LIST, BITMAP, RLIST 或者 ACK的状态报告。包含其他SUFIs的状态报告不被禁止。上层控制对于每个RLC实体应该使用哪个功能。如果使用了下列功能,那么状态报告的发送应该被推迟,即使在满足上面提到的触发条件的情况下:

a) STATUS 禁止

根据 9.5 节 f) 启动定时器 `Timer_Status_Prohibit`。在确认被禁止的情况下,不允许接收端发送状态报告(见 9.5 节 f))。如果在此期间触发了一个状态报告,那么当定时器 `Timer_Status_Prohibit` 超时的時候发送该状态报告,具体描述如下:

b) 无效

当触发一个状态报告时,接收端应该:

— 如果功能“STATUS 禁止”没有禁止状态报告的发送,那么:

- 构造并且向发送端发送状态报告,如 11.5.2.2 和 11.5.2.3 节中所定义的。

— 否则(如果状态报告至少被功能“STATUS 禁止”所禁止)

- 如果在状态报告中要求 MRW、MRW_ACK、WINDOW 或者 POLL SUFIs,那么:

- ◆ 立刻发送一个不包含 ACK, LIST, BITMAP, 和 RLIST SUFIs 的状态报告。

- 如果在状态报告中要求 ACK, LIST, BITMAP, 或者 RLIST SUFIs, 那么:

- ◆ 推迟发送这些 SUFIs, 直到禁止功能结束。

当定时器 `Timer_Status_Prohibit` 超时的時候,接收端应该:

— 如果在状态报告传输禁止期间至少有一个状态报告被触发,并且由于禁止而无法发送;并且

— 如果状态报告的发送不再被功能“STATUS 禁止”所禁止,那么:

- 向发送端发送一个状态报告,使用 11.5.2.3 节中描述的过程。

9.7.3 确认模式、非确认模式和透明模式的 SDU 丢弃功能

当经过一段时间或者多次不能成功传送RLC PDU时,发送端使用SDU丢弃功能从RLC PDU缓存区中丢弃RLC PDU。SDU丢弃功能可以避免缓存器溢出。RLC PDU丢弃功能具有几种可选的操作模式。上层控制每一个RLC实体应该使用的丢弃功能。

表7列出了RLC SDU丢弃功能的一系列操作模式,在后续的章节中将对它们进行详细的描述。

表7 控制何时执行 SDU 删除的条件列表

操作模式	出现
基于定时器的丢弃, 具有明确的信令	网络控制
基于定时器的丢弃, 没有明确的信令	网络控制
重传 MaxDAT 次之后, SDU 丢弃	网络控制
重传 MaxDAT 次之后, 没有丢弃	网络控制

9.7.3.1 基于定时器的丢弃, 具有明确的信令

在 RESET_AND_SUSPEND 状态时, 实体等待对等实体的响应或者是来自上层的原语 (CRLC-RESUME-Req), 此时在实体间无法进行数据的交互。

当从上层接收到CRLC-CONFIG-Req指示释放时, RLC实体:

- 进入 NULL 状态; 并且
- 认为是已经被终结。

当接收到与相应的RESET PDU具有相同RSN值的RESET ACK PDU时, RLC实体:

- 根据 11.4.4 进行响应; 并且
- 进入 LOCAL_SUSPEND 状态。

当从上层接收到CRLC-RESUME-Req时, RLC实体:

- 恢复, 也就是, 释放挂起限制; 并且
- 进入 RESET_PENDING 状态。

9.7.3.2 基于定时器的丢弃, 没有明确的信令

该可选择的方法仅适用于工作在非确认模式或者透明模式的RLC实体。它使用如同9.7.3.1节中所描述的基于定时器 (Timer_Discard) 的SDU丢弃触发机制。所不同的是该丢弃方法不使用任何对等信令。

对于每个从上层接收到的SDU, 发送端应该:

- 启动 SDU 传输时间的定时器监视。

当传输时间超过SDU的配置值时, 发送端应该:

- 没有明确信令指示的情况下丢弃该 SDU (对于工作在非确认模式的 RLC 实体, 使用 11.2.4.3 节描述的内容进行状态变量的更改)。

9.7.3.3 重传MaxDAT次之后, SDU丢弃

该可供选择的方法使用了重传次数作为SDU丢弃的触发机制, 因此该方法只适用于确认模式RLC。这使得SDU丢弃功能与信道速率相关。同时, 这个不同的SDU丢弃功能尽可能的保持连接的SDU丢失率恒定, 以变化的延时为代价。

如果 AMD PDU 被安排传输的次数等于 MaxDAT, 那么发送端应该:

- 丢弃所有包含在 AMD PDU 中的有 SDU 分段或"Length Indicators"指示 SDU 结束的所有 SDU; 并且
- 根据 11.6 节, 使用明确的信令通知接收端。

9.7.3.4 重传MaxDAT次之后, 没有丢弃

该可供选择的方法使用了重传次数, 因此仅适用于确认模式RLC。

如果 AMD PDU 被安排发送的次数等于 MaxDAT, 那么发送端应该:

- 初始化 RLC 复位过程 (见 11.3.4.4 节)。

9.7.3.5 没有配置SDU丢弃

如果非确认RLC实体没有配置SDU丢失，那么除非发送缓冲区满，否则发送端的SDU不应该被丢弃。

— 如果被丢弃的 SDU 的分段已经被发送给下层：

- 根据 11.2.4.3 节，在没有明确信令指示的情况下丢弃 SDU。

— 否则，如果被丢弃的 SDU 没有分段已经被发送给下层：

- 不使用任何丢弃过程的情况下从发送缓冲区中删除 SDU。

如果对透明模式的 RLC 实体没有配置 SDU 丢弃，那么发送端从上层接收到新的 SDU 后应该：

— 丢弃所有在以前 TTI 中从上层接收到的，并且还没有被发送给下层的 SDU。

— 在第一个可能的 TTI 中发送新的 SDU。

对于确认模式RLC实体，总是配置SDU丢弃模式。

9.7.4 无效

9.7.5 确认模式和非确认模式传输的本地挂起功能

高层可能将RLC实体挂起。

当工作在非确认模式的 RLC 实体被上层使用参数 N 挂起时，RLC 实体应该：

— 使用包含 VT (US) 当前值的确认信息来对挂起请求进行确认。

— 不发送“Sequence Number” $SN \geq VT (US) + N$ 的 UMD PDU。

当工作在确认模式的 RLC 实体被上层使用参数 N 挂起时，RLC 实体应该：

— 使用包含 VT (S) 当前值的确认信息来对挂起请求进行确认。

— 不发送“Sequence Number” $SN \geq VT (S) + N$ 的 AMD PDU。

当工作在非确认模式的 RLC 实体被上层回复时，RLC 实体应该：

— 恢复数据传输过程。

当工作在确认模式的 RLC 实体被上层回复时，RLC 实体应该：

— 如果 RLC 实体被挂起，并且没有 RLC 复位过程正在进行，那么：

- 恢复数据传输过程。

— 否则，如果 RLC 实体被挂起，并且 RLC 复位过程正在进行，那么：

- 删除挂起限制；
- 根据 11.4 节恢复 RLC 复位过程。

9.7.6 确认和非确认模式的 RLC 停止, RLC 继续功能

上层可能停止RLC实体。

当RLC实体被停止时，RLC定时器不受影响。

当 RLC 实体被上层停止时，RLC 实体应该：

— 不向下层发送任何 RLC PDU，或者接收任何 RLC PDU。

— 延迟触发轮询功能或者状态传输，直到 RLC 实体被继续。

注：如果通过向MAC发送RLC实体消息参数进行了TFC选择交换的初始化，那么RLC实体可能延迟停止功能，直到下一个TTI结束。

当上层继续一个 RLC 实体时，RLC 实体应该：

— 如果 RLC 实体被停止，那么：

- 继续数据传输和接收;
- 进行触发轮询功能和状态传输。
- 否则, 如果 RLC 实体没有被停止, 那么:
 - 不做任何动作。

9.7.7 确认和非确认模式的 RLC 重建功能

RLC重建由高层触发。

RLC重建功能适用于AM RLC或UM RLC。对UM RLC, 重建整个RLC实体; 对于AM RLC, 高层可以请求重建发送端、或接收端或整个RLC实体。

当上层重建 UM RLC 实体时, RLC 实体应该:

- 将状态变量复位成它们的初始值。
- 将可配置参数设置为它们的配置值。
- 按照上层配置的值设置超帧号 (HFN)。
- 如果 RLC 实体工作在非确认模式, 那么:
 - 如果它是一个接收 UM RLC 实体, 那么:
 - ◆ 丢弃所有 UMD PDU;
 - ◆ 停止所有定时器。
 - 如果它是一个发送 UM RLC 实体, 那么:
 - ◆ 丢弃一个或多个分段已经被发送给下层的 RLC SDU。
 - ◆ 如果上层请求
 - 通知上层丢弃的 SDU。
- 如果 RLC SDU 没有丢弃, 那么不停止 Timer_Discard。

当上层重建AM RLC实体的发送端和/或接收端时, RLC实体应该:

- 如果 AM RLC 接收端被重建:
 - 将状态变量按 9.4 节复位成它们的初始值。
 - 将接收端的可配置参数按 9.6 节设置为它们的配置值。
 - 按照上层配置的值设置接收端的超帧号 (HFN)。
 - 丢弃接收端已经接收的 AMD PDU。
 - 若只有接收端的 RLC 实体重建:
 - ◆ 丢弃所有 (包括捎带类型) 状态 PDU 中的确认状态报告、WINDOW SUFI 和 MRW_ACK_SUFI 信息, 丢弃 RESET PDU 和 RESETACK PDU (即丢弃所有的控制 PDU 除了 MRW_SUFI);
 - ◆ 若 RLC 复位过程正在进行 (例如 Timer_RST 在运行):
 - 停止 Timer_RST;
 - 中断正在进行的 RLC 复位过程;
 - 复位 VT (RST);
 - 在 RLC 重建完成后重新开始一个新的 RLC 复位过程。
 - ◆ 停止 Timer_Status_Prohibit 定时器;
 - ◆ 对于在重建之前已经被调度用于传输但是还没有发送给底层的剩余的控制 PDU 和 SUFI:

- 使用配置后的 RLC PDU 大小重新将没有被丢弃的控制 PDU 和 SUFI 组成一个新的控制 PDU;
- 调度新的控制 PDU 进行发送;
- 如果 AM RLC 发送端被重建:
 - 将状态变量按 9.4 节复位成它们的初始值。
 - 将接收端的可配置参数按 9.6 节设置为它们的配置值。
 - 按照上层配置的值设置发送端的超帧号 (HFN)。
 - 丢弃发送端没有被全部发送的 SDU (即丢弃的 AMD PDU 中包含了分段的 SDU, 以及指示 SDU 结尾的 LI)。
- ◆ 用户也可以丢弃部分传输的 SDU (至少是 1 个包含了 SDU 分段的 AMD PDU 被传输)。
- 对于没有丢弃的 SDU, 按照配置后的 RLC PDU 大小重新分段组成 AMD PDU (这里配置后的 RLC PDU 大小可能和重配置前的不同)。
- 如果仅仅 AM RLC 发送端被重建:
 - ◆ 丢弃所有状态 PDU 和捎带的状态 PDU 中的 MRW SUFI, 丢弃 RESET PDU 和 RESET ACK PDU; (例如, 丢弃所有的控制 PDU, 除了确认的状态报告、WINDOW SUFI 和 MRW ACK SUFI);
 - ◆ 对于重建之前准备发送但是还没有发送给底层的剩余的控制 PDU 和 SUFI:
 - 按照配置后的 RLC PDU 大小, 将没有丢弃的控制 PDU 和 SUFI 组成新的控制 PDU;
 - 将新的控制 PDU 发送。
 - ◆ 若 RLC 复位过程正在进行:
 - 停止 Timer_RST;
 - 中断正在进行的 RLC 复位过程;
 - 在 RLC 重建之后重新开始一个新的 RLC 复位过程;
 - ◆ 停止定时器 Timer_Poll、Timer_MRW 和 Timer_Poll_Prohibit;
 - 如果 AM RLC 发送端和接收端同时被重建。
 - ◆ 接收端和发送端同时丢弃控制 PDU, 发送端丢弃 AMD PDU;
 - ◆ 如果 RLC 复位过程正在进行 (比如 Timer_RST 正在运行)
 - 中断正在进行的 RLC 复位过程;
 - ◆ 停止 9.5 节描述的所有定时器, 除了 Timer_Poll_Periodic 和 Timer_Status_Periodic、以及还有 SDU 没有丢弃时候运行的 Timer_Discard。
 - 如果上层请求,
 - ◆ 通知上层丢弃的 SDU。

注: 如果通过向 MAC 发送 RLC 实体信息参数进行了 TFC 选择交换的初始化, 那么 RLC 实体可能延迟重建功能, 直到下一个 TTI 结束。

9.7.8 确认和非确认模式的加密

如果无线承载使用非透明 RLC 模式 (AM 或者 UM), 那么根据下列规则在 RLC 使用加密功能。数据单元根据下面描述的传输模式进行加密。

- 对于 RLC UM 模式, 除了 UMD PDU 的第一个字节以外, 其他属于加密单元, 也就是说除了 UMD PDU 包头, 如图 25 所示。

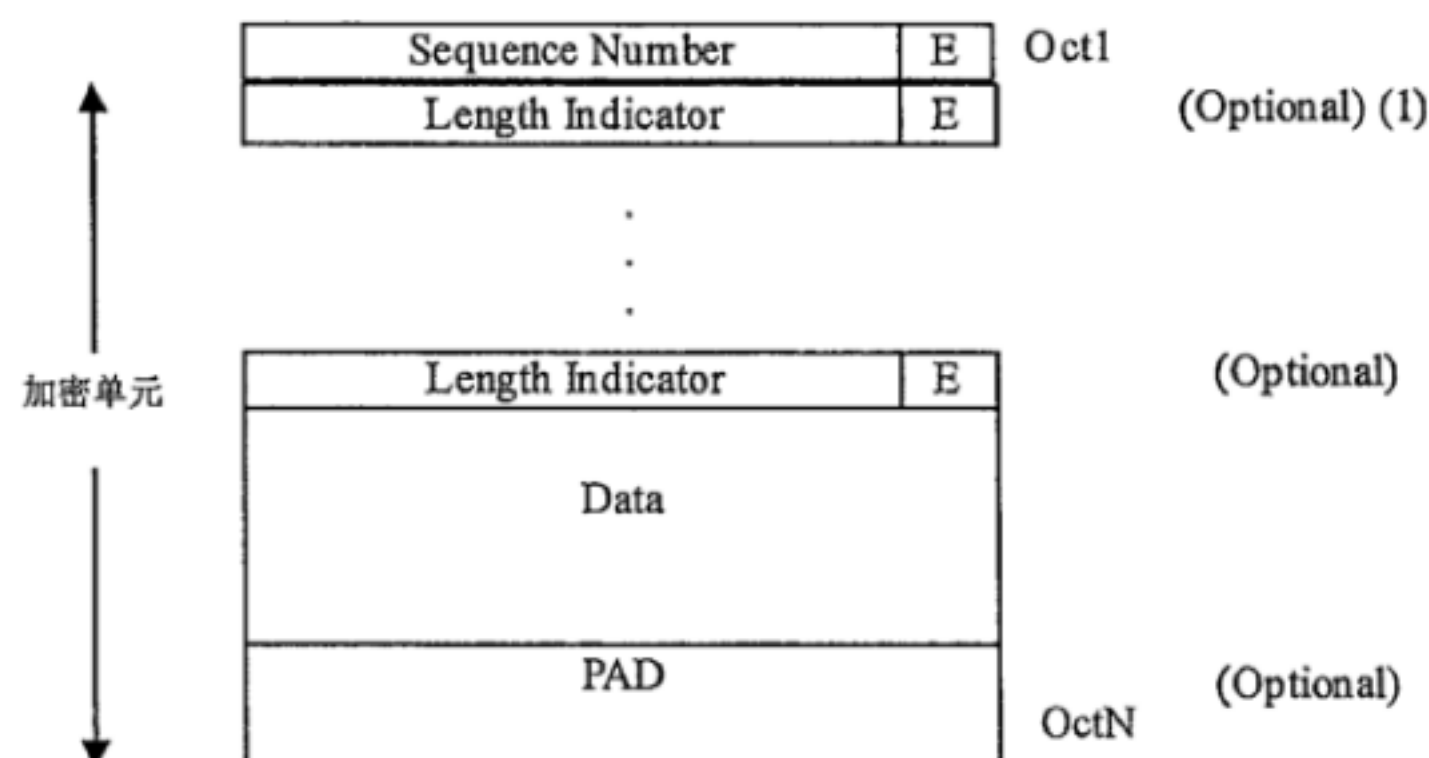


图25 UMD PDU 的加密单元

对于RLC AM 模式,除了AMD PDU的前两个字节以外,其他属于加密单元,也就是说除了AMD PDU 包头,如图26所示。

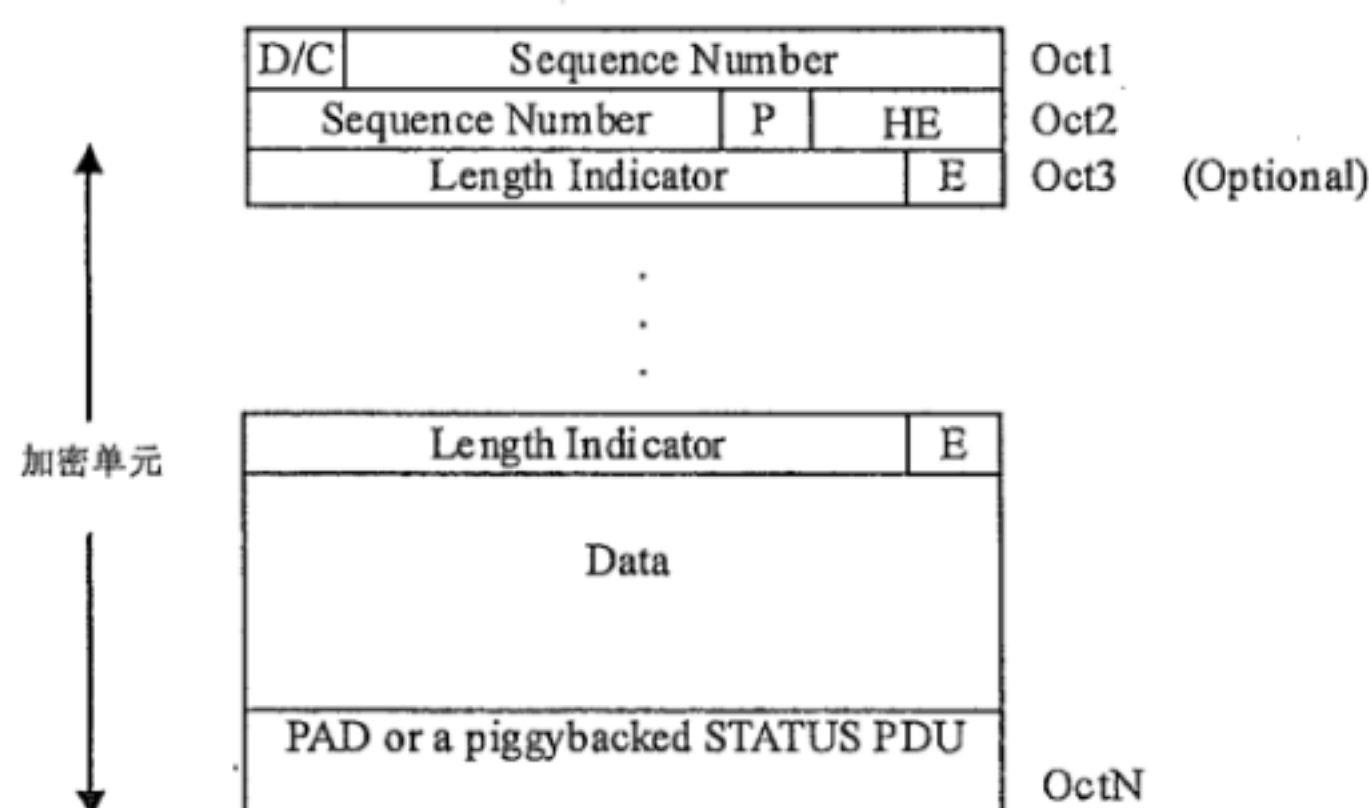


图26 AMD PDU 的加密单元

由上层配置加密算法和加密键(参见3GPP TS.25.331)并且如3GPP TS 33.102中定义的使用加密方法。

3GPP TS 33.102中定义了RLC所要求的用于加密的参数,并且作为加密算法的输入。下面列出了RLC 所要求的,由上层提供的参数:

- RLC AM HFN (映射到RLC AM 的无线承载的超帧号);
- RLC UM HFN (映射到RLC AM 的无线承载的超帧号);
- BEARER (定义为3GPP TS 33.102 中所描述的无线承载标识。它将使用3GPP TS 25.331 中所描述的RB 标识-1);
- CK (加密键)。

9.7.9 高层重配置RLC 参数

RLC参数可以被高层重配置(修改)。

当RLC参数被高层重配置,UE应该:

- 使用重配置的RLC 参数。

如果Configured_Rx_Window_Size 参数被重配置:

- UE 将更新状态变量VR (UM) (见9.4 节);
- UE 将更新状态变量VR (MR) (见9.4 节);
- 对"Sequence Number" 为 x 的AMD PDU, 其中 $VR (MR) \leq x < VR (H)$:
- UE 将丢弃这些AMD PDU, 同时将相应地更新状态变量VR (H);

- 认为被丢弃的 AMD PDU 没有被成功接收。

如果 Configured_Tx_Window_Size 参数被重配置:

- UE 将状态变量 VT (WS) 设为 Configured_Tx_Window_Size;
- UE 更新状态变量 VT (MS) (见 9.4 节);
- 对 "Sequence Number" 为 x 的 AMD PDU, 其中 $VT (MS) \leq x < VT (S)$:

- UE 将不丢弃任何没有收到肯定应答的 AMD PDU;
- UE 可以丢弃已经得到肯定应答的 AMD PDU。

当发送窗口大小或接收窗口大小被重配置时, 所需要的内存可能会暂时超过可用内存 (见 11.3.4.9)。

如果参数 OSD_Window_Size 被重配置:

- 用户需将存储窗口 ($VR (UOH) \geq SN > VR (UOH) - OSD_Window_Size$) 外的 SN 对应的 PDU 删除;

如果参数 DAR_Window_Size 被重配置:

- UE 将从存储中移除所有序列号在窗外的 PDU 并将其递交给上层 RLC 功能进行重装配, 如 11.2.3 节所述, 并根据 9.7.10 节更新状态变量 VR (UDR)。

- UE 将更新状态变量 VR (UDT) 并根据 9.7.10 节按需对定时器 Timer_DAR 进行操作。

9.7.10 非确认模式的重复避免与重排序

重复避免与重排序功能可以在 UE 的接收 UM RLC 实体中配置。该功能将从多个数据源收到的 PDU 序列进行组合和/或将从一个数据源收到的多次传输进行重复, 形成一个顺序传输的 PDU 序列递交给头移除与重装配功能。它根据 UM PDU 序列号完成重复检测、丢弃及重排序功能。UM RLC 从多个数据源接收输入数据, 这些输入可以被增加或移除, 而不改变缓冲区内容、与重复避免与重排序功能关联的定时器、状态变量以及任何随后的 UM RLC 功能。

重复避免与重排序功能使用状态变量 VR (UDR) 和一个接收窗口, 该窗口是从 $VR (UDH) - DAR_Window_Size + 1$ 到 $VR (UDH)$ 并包括边界。对于重排序该功能使用一个缓冲区来临时存储 PDU。

对接收到的每个 PDU, 重复避免与重排序功能将 (下面用 SN 表示每个 PDU 的序列号):

设置初始状态变量:

- 如果该 PDU 是建立或重建之后重复避免与重排序功能收到的第一个 PDU:

- VR (UDH) 赋值为 SN;
- VR (UDR) 赋值为 $VR (UDH) - DAR_Window_Size + 1$ 。

重复避免与重排序:

- 如果 SN 在接收窗口内:

- 如果 $SN < VR (UDR)$ 或者已经有序列号为 SN 的 PDU 存在缓存中:
- ◆ 该 PDU 将被丢弃。

- 否则:

- ◆ 该 PDU 将被放入缓存中。

- 如果 SN 在接收窗口外:

- 该 PDU 将被放入缓存中;
- VR (UDH) 将被赋值为 SN, 据此将接收窗口提前;

- 对于任何存储的序列号 $< VR(UDH) - DAR_Window_Size + 1$ 的 PDU, 也就是在位置更新之后落在窗外的 PDU, 将这些 PDU 从缓存中移除并递交给上层 RLC 功能以执行 11.2.3 节规定的行为。

- 如果 $VR(UDR) < VR(UDH) - DAR_Window_Size + 1$, 也就是 $VR(UDR)$ 落在了更新的接收窗之外:

- ◆ $VR(UDR)$ 赋值为 $VR(UDH) - DAR_Window_Size + 1$ 。

- 如果序列号为 $VR(UDR)$ 的 PDU 保存在缓存中:

- 对于该 PDU 以及序列号从 $VR(UDR) + 1$ 开始的任何连续的 PDU 序列, 将这些 PDU 从缓存中移除并递交给上层 RLC 功能以执行 11.2.3 节规定的行为

- $VR(UDR)$ 赋值为 $x + 1$, 这里 x 是提交给上层 RLC 功能的序列号最高的 PDU 的序列号。

定时器操作:

- 如果重复避免与重排序功能存储的 PDU 的 $SN > VR(UDR)$:

- 启动 $Timer_DAR$;

- $VR(UDT)$ 赋值为 PDU 的序列号。

- $Timer_DAR$ 停止条件:

- 如果序列号为 $VR(UDT)$ 的 PDU 在 $Timer_DAR$ 超时前被移除。

- 如果 $Timer_DAR$ 超时:

- 对于保存的所有序列号小于或等于 $VR(UDT)$ 的 PDU, 以及对于序列号从 $VR(UDT) + 1$ 开始的任何连续的 PDU 序列, 将这些 PDU 从缓存中移除并递交给上层 RLC 功能以执行 11.2.3 节规定的行为;

- $VR(UDR)$ 赋值为 $x + 1$, 这里 x 是提交给上层 RLC 功能的序列号最高的 PDU 的序列号。

- 当 $Timer_DAR$ 停止或超时, 并且仍然有 PDU 在重复避免与重排序功能保存:

- $Timer_DAR$ 将被启动;

- $VR(UDT)$ 将被赋值为保存的编号最高的 PDU 的序列号。

10 未知、未预见和错误协议数据的处理

10.1 错误序列号

当出现下列情况之一, 一个 STATUS PDU 或 Piggybacked STATUS PDU 被认为携带了“错误的序列号”:

- 一个 LIST、BITMAP 或 RLIST SUFI 中, 至少有一个被否定确认的 AMD PDU 的序列号不在“确认状态变量” $VT(A)$ 和“发送状态变量” $VT(S)$ 的值之间; 或者

- 一个 ACK 的 LSN 不在“确认状态变量” $VT(A)$ 和“发送状态变量” $VT(S)$ 之间。

如果一个 AM RLC 实体收到了一个包含错误序列号的 STATUS PDU 或 Piggybacked STATUS PDU, 将丢弃该 PDU, 并发起 RLC 重启过程 (参见 11.4)。

10.2 不一致的状态指示

如果 AM RLC 实体接收到一个 STATUS PDU 或 Piggybacked STATUS PDU 指示了同一 AMD PDU 的不同状态, 这个 STATUS PDU 或 Piggybacked STATUS PDU 将被丢弃。

10.3 无效的 PDU 格式

如果一个 UM 或 AM RLC 实体接收到一个 RLC PDU, 包含保留或无效值, 该 RLC PDU 将会被丢弃。

10.4 CRC 错误的 RLC PDU

如果UM或AM RLC实体接收到有错误指示RLC PDU，则：

— 丢弃该 RLC PDU。

如果 TM RLC 实体接收到有错误指示的 RLC PDU，则：

— 如果"Delivery of Erroneous SDUs"已配置：

• 按 11.1.3 节处理该 RLC PDU。

— 否则：

• 丢弃该 RLC PDU。

11 基本过程

11.1 透明模式数据（TMD）传送过程

11.1.1 概述

透明模式数据传送过程是用于透明模式下的两个RLC对等实体之间的数据传送。该过程仅适用于处于传送准备状态（DATA_TRANSFER_READY）下的RLC实体。透明模式数据传送的基本过程如图27所示。发送实体可以是UE或网络，接收实体可以是网络或UE。

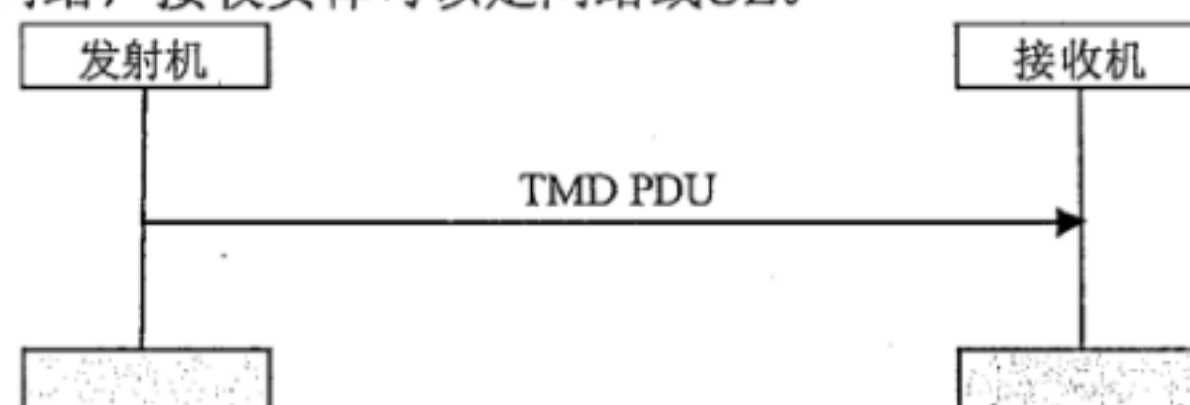


图27 透明模式数据传送过程

DTCH、CCCH（仅用于上行链路）、SHCCH（仅用于上行链路）、BCCH、PCCH信道可以使用透明模式数据传送。逻辑信道的类型取决于RLC实体是处于用户平面（DTCH）还是控制平面（CCCH/SHCCH/BCCH/PCCH）。

11.1.2 TMD PDU 传送

当有来自高层的透明模式数据传送的请求，发送实体即启动此过程。当发送实体处于传送准备状态时，它将把从高层接收的数据单元SDU放入TMD PDU中。如果需要，RLC将完成分段的任务。

在每个传输时间间隔（TTI）可以传送一个或几个PDU。对于每个TTI，由MAC决定应用哪种PDU大小（当使用了分段时应用）和可以传送多少个PDU。不能在一个TTI内传输的SDU将会根据RRC设置的丢弃配置进行缓冲。

如果上层没有配置丢弃设置，为了从上层接收新的SDU，RLC实体将丢弃在以前TTI中收到的SDU单元。

如果应用了基于定时器的SDU丢弃配置（非显式信令的SDU丢弃），将会为从高层接收的每一个SDU应用一个定时器Timer_Discard，在接收SDU时启动。

RLC实体对要传送RLC SDU进行排队，通知下一层MAC准备接收数据，并完成向下一层MAC的TMD PDU传送。

11.1.2.1 TMD PDU内容设置和发送

TMD PDU包括一个完整的SDU或SDU的一部分。如何进行分段是在业务建立时决定的。没有附加开销或报头。相反，分段是基于要使用的传输信道的传输格式的。发送实体将设定一个特殊的传输格式通知接受方执行了分段。

当一个或多个按计划发送的 RLC SDU 准备就绪, 发送实体按如下过程操作:

- a) 如果进行分段操作, 发送方将通知下一层要发送的 SDU 的大小, 并根据下层指示的 PDU 的大小将 SDU 分段, 放入多个 TMD PDU 中;
- b) 否则, 发送方将把 SDU 放入 TMD PDU, 并通知 MAC 将要发送的 SDU 的数量和大小;
- c) 根据 MAC 层请求的 TMD PDU 数量把 TMD PDU 发送给 MAC;
- d) 不能在一个 TTI 内传输的 SDU 将会根据 RRC 设置的丢弃配置进行缓冲。

11.1.3 TMD PDU 的接收

当接收PLC实体在一个TTI中接收到MAC传送的TMD PDU, 接收实体将把PDU重新组合(如果进行了分段)成RLC SDU。RLC通过TM-SAP将RLC SDU转发给高层。

如果使用了分段操作, 当组成一个SDU的一个或多个TMD PDU出错, 则认为该SDU出错; 若不使用分段操作, 当一个TMD PDU出错, 则认为相应的SDU出错。

- a) 如果“错误的 SDU 传送 (Delivery of Erroneous SDUs)”设置为“No”, RLC 将只向高层传送正确的 RLC;
- b) 如果设为“No Detect”, RLC 将把全部接收的 RLC_SDU 向高层传送;
- c) 如果设为“Yes”, RLC 除将把全部接收的 RLC_SDU 向高层传送外, 还会对每一个接收错误的 SDU 向高层提供一个错误指示。

11.1.4 异常情况

11.1.5 没有显式信令的 SDU 丢弃

当发送方的Timer_Discard定时器超时后, 发送方应当丢弃所联结的SDU, 如果被要求的话, 将这一被丢弃的SDU通知给上层。在由于向MAC发送RLC实体信息参数而启动了TFC选择交换的情况下, UE需要等待到它提供给MAC所请求的PDU之后, 才能丢弃该SDU。

11.2 非确认模式数据 (UMD) 传送过程

11.2.1 概述

非确认模式数据传送过程是用于在非确认模式下两个RLC对等实体之间的数据传送。该过程仅适用于处于传送准备状态 (DATA_TRANSFER_READY) 或本地暂停状态 (LOCAL_SUSPEND) 下的RLC实体。用于非确认模式数据传送的基本过程如图28所示。发送实体可以是UE或网络, 接收实体可以是网络或UE。

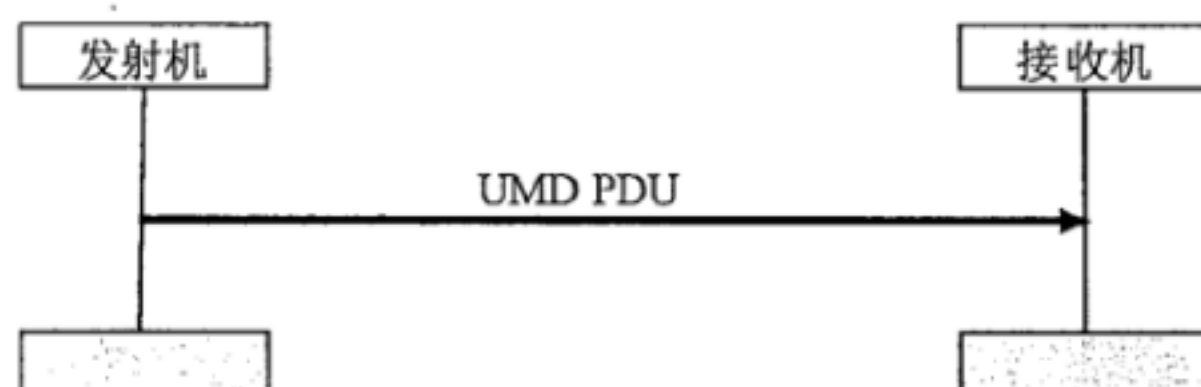


图28 非确认模式数据传送过程

可用的信道为DTCH、DCCH、CCCH (仅用于下行链路)、SHCCH (仅用于下行链路)、MCCH、MSCH、MTCH。逻辑信道的类型取决于RLC实体是位于用户平面 (DTCH/CTCH/MTCH) 还是位于控制平面 (DCCH/CCCH/SHCCH (仅用于下行) /MCCH/MSCH)。在每个传输时间间隔 (TTI) 可以传送一个或几个PDU。对于每一个TTI来说, 由MAC决定哪一个PDU的大小可以使用和可以同时传送多少个PDU。

11.2.2 UMD PDU 传送

当有来自高层的非确认模式数据传送的请求, 发送实体即启动此过程。

当发送实体处于传送就绪状态时, 它将把从高层接收的数据进行分段并且如果可能还要进行连接, 之后放入 PDU 中。

如果上层没有配置丢弃设置, RLC 实体将在发送缓冲区满的条件下丢弃 SDU 单元。

如果应用了基于定时器的 SDU 丢弃配置(非显式信令的 SDU 丢弃), 将会为从高层接收的每一个 SDU 应用一个定时器 `Timer_Discard`, 在接收 SDU 时启动。

RLC 实体对要传送 RLC SDU 进行排队, 通知下一层准备接收数据, 并完成向下一层的 UMD PDU 传送。

如果一个 UMD PDU 只包含一个带有一个长度指示 (Length Indicator) 的 RLC 头 (指示 PDU 的其余部分是填充), 那么这个 PDU 会被认为是一个填充 PDU。

11.2.2.1 UMD PDU 内容设置

发送实体将完成如下设置:

- a) 序列号 (SN) 字段将设置为 VT (US)。
- b) 对于 RLC 头中的每个扩展比特域 (Extension bit), 如果下一个字段是一个长度指示字段, 则扩展比特设置为 1, 否则将设置为 0。
- c) 对于每个结束于此 PDU 中 SDU 设置一个长度指示字段。LI 的设置见 9.2.2.8 节。

11.2.2.2 UMD PDU 的传送

当要传送的 SDU 准备就绪, 发送实体将:

- a) 通知 MAC 将要发送的 SDU 的数量和大小;
- b) 如果配置了 SN_Delivery, 则进行分段, 但是不进行级联; 否则根据下层指示的 PDU 的大小将 SDU 分段或连接 (见 9.2.2.9 节);
- c) 根据请求的数量, 把相应的 UMD PDU 发送给 MAC;
- d) 对于每个被传输的 UMD PDU, 更新 VT (US) 状态变量的值;
- e) 不能在一个 TTI 内传输的 SDU 将会根据 RRC 设置的丢弃配置进行缓冲 (参见 9.7.3)。

11.2.3 UMD PDU 的接收

当接收到一个 UMD PDU, 接收实体将:

— 如果配置了 SDU 乱序递交功能:

- 执行操作参见 11.2.3.2 节;

— 否则:

- 执行操作参见 11.2.3.1 节。

11.2.3.1 SDU 丢弃和重组

当接收到底层或者重复避免与重排序子实体传来的 UMD PDU 时候, 接收端应当:

— 如果配置了乱序接收并且 $SN \geq VR(UM)$:

- 丢弃 UMD PDU。

— 否则:

- 丢弃 UMD PDU;
- 根据每个接收到的 UMD PDU 更新 VR (US) (参见 9.4 节);

- 若 VR (US) 的更新步长不等于 1 (可能有一个或多个 UMD PDU 丢失):
 - ◆ 将丢弃的 UMD PDU 中包含分段或者 LI 指示 SDU 结尾的 SDU 丢弃, 具体操作参见 9.2.2.8 节和 9.2.2.9 节。
- 若特殊的 LI ("1111 100" or "1111 1111 1111 100") 指示为接收的 UMD PDU 的第一个 LI, 下行方向:
 - ◆ 将该 UMD PDU 的第一个数据字节认为是一个 RLC SDU 的第一个字节。
- 若扩展比特指示 UMD PDU 包含一个未被分段、级联、填充的完整 SDU:
 - ◆ 将该 UMD PDU 中的数据部分认为是一个完整的 RLC SDU。
- 若特殊的 LI 指示 ("1111 101"或"1111 1111 1111 101") 为下行方向接收的 UMD PDU 的第一个 LI:
 - ◆ 将该 UMD PDU 的第一个数据字节认为是一个 RLC SDU 的第一个字节, 并将最后一个字节认为是相同 RLC SDU 的最后一个字节。
- 若特殊的 LI 指示 ("1111 1111 1111 010") 为下行方向接收的 UMD PDU 的第一个 LI:
 - ◆ 将该 UMD PDU 的第一个数据字节认为是一个 RLC SDU 的第一个字节, 并将倒数第二个字节认为是相同 RLC SDU 的最后一个字节。
- 将接收的 UMD PDU 重组为 RLC SDU;
- 将 RLC SDU 通过 UM-SAP 递交到高层。

11.2.3.2 SDU 乱序递交

为了保证从 UMD PDU 中重组出 SDU, 接收端需要存储 PDU 直到和该 PDU 相关的所有的 SDU 都可以被重组出来, 或者直到这些 PDU 由于下面描述的原因被丢弃掉。当可以根据 PDU 和表示 SDU 结束的长度指示重组出 SDU 的时候, 需要尽快将重组后的 PDU 传递到高层。

基于从底层接收到的一组 UMD PDU, 接收方对每个 PDU 应当作如下处理 (设收到的 PDU 的序号为 SN):

- 如果 PDU 是第一个接收到的 PDU (接收实体建立或重建后或者 Timer_OSD 超时后):
 - VR (UOH) 取值设置为 SN-1。
- 如果 $VR (UOH) > SN > VR (UOH) - OSD_Window_Size$, 那么:
 - 如果该 SN 的 PDU 已经保存:
 - ◆ 丢弃该 PDU。
 - 否则:
 - ◆ 按序号顺序存储该 PDU。
- 否则:
 - $VR (UOH) = SN$, 移动存储窗口;
 - 按序号顺序存储该 PDU;
 - 将序号超出存储窗口 (存储窗口 $VR (UOH) > SN > VR (UOH) - OSD_Window_Size$) 的 PDU 从存储队列中删除;
 - 如果定时器 Timer_OSD 激活, 则停止该定时器;
 - 将定时器 Timer_OSD 开启。

— 如果序号 SN 对应的 PDU 被存储:

- 遍历存储队列, 并查找序号连续存储的 PDU 以及使用的 LI 的取值信息, 如果有连续的 PDU 存储, 则:

- ◆ 将连续的 PDU 重组为 SDU;
- ◆ 将生成的 RLC SDU 通过 UM-SAP 递交到高层;
- ◆ 将不包含任何 SDU 分组的 PDU 以及不包含特殊 LI 的 PDU 从存储队列中删除。这里指的特殊 LI 指示用于指示还没有被重组的 SDU 的结尾, 包括“0000 000”, “0000 0000 0000 000” or “1111 1111 1111 011”。

注 1: 如果在 SDU 重组后 PDU 从存储队列中删除, 那么重传的 PDU 可能会造成重复的 SDU 发送给高层。

— 如果 Timer_OSD 超时:

- 将所有存储的 PDU 从存储队列中删除。

注 2: 当配置了 SDU 乱序传递功能, 如果发送状态变量 VT (US) 在 Timer_OSD 超时之前大于等于 $(128 + SN - OSD_Window_Size + 1)$ (这里的 SN 是在定时器超时前发送或者重发的所有 PDU 中序号最低的), 发送端应当考虑到连续 $(128 - OSD_Window_Size)$ 个 PDU 被丢失的情况可能会导致接收端出现协议不能检测到的错误,

注 3: 发送方不能在包含 MBMS 接入信息的 PDU、SDUs 或者分段的 SDUs 中串接其他 MCCH 消息类型的 SDU 或者 SDU 分段。

注 4: SDU 包含在连续序号的 PDU 中。为了保证包含 MBMS 接入信息消息的 SDU 在对应发送的时间点发送, 发送端可以不按照序号顺序发送该 PDU。

注 5: 发送端不能在一个 PDU 中单独发送包含特殊长度指示为“0000 000”, “0000 0000 0000 000”及“1111 1111 1111 011”的 MBMS 接入信息消息的 SDU 或者 SDU 分段。

11.2.4 异常情况

11.2.4.1 长度指示值为 UMD PDU 的保留值

当接收到一个 UMD PDU, 该 UMD PDU 所包含的长度指示的值为该协议版本中 UMD PDU 的保留值, 接收实体将丢弃该 UMD PDU。

11.2.4.2 无效长度指示值

如果一个 PDU 的长度指示的长度大于该 PDU 的长度减去 RLC 头部大小, 并且不是 9.2.2.8 中预定义的值中的一个, 该 PDU 将被丢弃。

11.2.4.3 没有显式信令丢弃 SDU

当发送方的 Timer_Discard 定时器超时后, 发送方应当丢弃所联结的 SDU; 如果被要求的话, 将这一被丢弃的 SDU 通知给上层。在由于向 MAC 发送 RLC 实体信息参数而启动了 TFC 选择交换的情况下, UE 需要等待到它提供给 MAC 所请求的 PDU 之后, 才能丢弃该 SDU。

为了指示在当前的 RLC PDU 之前的 RLC SDU 被丢弃, 对于丢弃后传送的第一个 UMD PDU, 发送 RLC 实体将增加 VT (US), 使得 UMD PDU 的序列号相比以前的 UMD PDU 增加 2 步。同时用一个 RLC SDU 的首字节填充该 PDU 的第一个数据字节。如果扩展比特没有指示 UMD PDU 包含一个未被分段、级联或填充的完整 SDU, 为了阻止接收端丢弃额外的 SDU, 在丢弃操作之后, 设置第一个发送的 PDU 的 LI 字段, 指示前一个 SDU 刚好填满上一个 PDU 的值。

11.2.4.4 无效的 PDU 大小

在UE侧, 如果"DL RLC UM LI size"配置为7 bit, 如果收到UMD PDU 的大小超过125 字节, 则接收端丢弃该UMD PDU。

11.3 确认模式数据 (AMD) 传送过程

11.3.1 概述

确认模式数据传送过程用于确认模式下两个RLC对等层实体之间数据的传送。该过程仅适用于处于传送准备状态 (DATA_TRANSFER_READY) 或本地暂停状态 (LOCAL_SUSPEND) 下的RLC实体。确认模式数据传送的基本过程如图29所示。发送实体可以是UE或网络, 接收实体可以是网络或UE。

如果发送实体位于控制平面, AMD PDU将在DCCH信道上传送, 如果位于用户平面, AMD PDU将在DTCH信道上传送。在每个传输时间间隔 (TTI) 可以传送一个或几个PDU。对于每一个TTI来说, 由MAC决定哪一个PDU的大小可以使用和在可以传送多少个PDU。

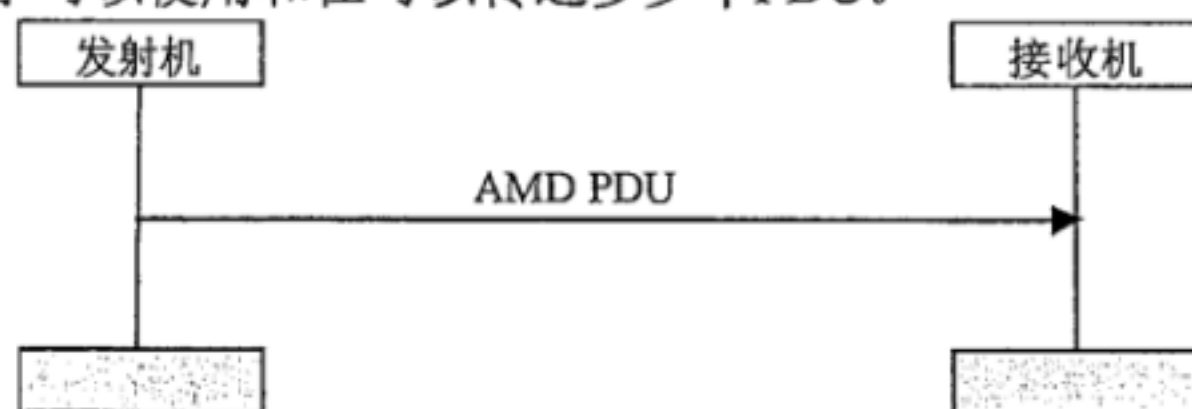


图29 确认模式数据传送过程

11.3.2 AMD PDU 的传送

当有来自高层的确认模式数据传送或 PDU 的重传的请求, 发送实体将:

— 当发送实体从高层收到了 RLC SDU:

- 如果配置了固定 RLC PDU 大小, 则根据高层配置的 PDU 的固定大小将 RLC SDU 分段或级连放入多个 AMD PDU 中 (见 9.2.2.9 节)。

- 如果 PDU 的最后一个字节是 SDU 的最后一个字节而且在 PDU 内没有 SDU 串联, 并且高层配置了 HE 域特殊值的使用, 则设置 HE 域来指示 PDU 的最后一个字节是 SDU 的最后一个字节 (见 9.2.2.7 节)。

- 如果配置了可变 RLC PDU 大小, 则 RLC SDU 的最后一个分段将于下一个 RLC SDU 的第一个分段进行级联以将数据域至少填至达到最小 UL RLC PDU 大小; 如果待传数据量不足以填满一个最小尺寸的 AMD PDU, 则允许创建一个包含所有待传数据的 AMD PDU, 即使形成的 RLC PDU 大小小于最小 UL RLC PDU 大小。

- 根据 9.2.2.8 节为在一个 PDU 中结束的每个 SDU 设置一个长度指示字段 (LI), 已经根据 9.2.2.7 节用 HE 域指示 SDU 结束的 SDU 除外。

- 如果应用了基于定时器的显式信令的 SDU 丢弃配置, 将会为从高层接收的每一个 SDU 应用一个定时器 Timer_Discard (见 9.7.3 节)。

- 调度 AMD PDU 的传输。

— 对于每一个已经被否定应答的 AMD PDU (见 11.5.3 节), 如果 AMD PDU 的序列号小于 VT (MS), 则调度该 AMD PDU 进行重传。

— 如果轮询被配置的轮询功能触发 (见 9.7.1 节), 并且

— 如果轮询功能没有被禁止 (见 9.5 节), 并且

— 如果没有 AMD PDU 被调度用于首传或重传, 并且

— 如果至少一个AMD PDU已发送, 而且还未被丢弃而且没有收到确认。

- 如果 Configured_Tx_Window_Size 大于等于 2048, 选择序列号为 $VT(S) - 1$ 的 PDU 重传, 或者当配置了“可变 RLC PDU 大小”时根据 9.2.2.11.9 小节装配一个 POLL SUFI;

- 否则如果 Configured_Tx_Window_Size 小于 2048, 则选择序列号为 $VT(S) - 1$ 的 PDU 或还没有被确认、没有被丢弃的 PDU 来重传, 或者当配置了“可变 RLC PDU 大小”时根据 9.2.2.11.9 小节装配一个 POLL SUFI。

- 如果一个 AMD PDU 被选择, 则为重传调度选择的 AMD PDU (为了发送一个轮询), 或者如果装配了一个 POLL SUFI, 则调度一个包含 POLL SUFI 的 STATUS PDU 进行传输, 进一步如果配置了定时器 Timer_Poll, 则根据 9.5 节启动 Timer_Poll。

注 1: 在下行方向, 如果配置了可变 RLC PDU 大小, UTRAN 将会对 RLC SDU 进行分段甚至级联, 形成不大于最大 RLC PDU 大小的 AMD PDU。

注 2: 在下行方向, 如果配置了下行可变 RLC PDU 大小, UTRAN 可以通过装配 POLL SUFI 来发起轮询功能。如果装配了一个 POLL SUFI, UTRAN 应向底层调度并递交一个状态 PDU, 其中包含 POLL SUFI。

注 3: 在上行方向, 如果配置了上行可变 RLC PDU 大小, UE 可以通过装配 POLL SUFI 来发起轮询功能。如果装配了一个 POLL SUFI, UE 应向底层调度并递交一个状态 PDU, 其中包含 POLL SUFI。

每一个 AMD PDU 传输时, 发送实体将增加它的状态变量 $VT(DAT)$ 值。如果 $VT(DAT) < MaxDAT$, 发送实体通知 MAC 待传送数据已准备就绪, 并把相应的 PDU 传送给 MAC 层。如果 $VT(DAT) = MaxDAT$, 操作参见 11.3.3a。

在 AM 模式下, 如果一个 PDU 有以下情况会被认为是一个填充 PDU:

- a) 只包含带有一个长度指示字段 (用来指示 PDU 的其余部分是填充) 的 RLC 头部和填充比特。
- b) 只包含一个 NO_MORE_SUFI 字段的 STATUS PDU。

11.3.2.1 AMD PDU 内容设置

如果 PDU 是第一次传送:

— “Sequence Number” 字段将被设置为 $VT(S)$ 且 $VT(S)$ 将被更新。

— 如果 PDU 的最后一个字节是 SDU 的最后一个字节而且在 PDU 内部没有 SDU 级联, 并且高层已经配置了 HE 字段特殊值的使用, 则设置 HE 字段指示 PDU 的最后一个字节是 SDU 的最后一个字节 (见 9.2.2.7 节)。

— 根据 9.2.2.8 为 AMD PDU 结尾的每个 SDU 设置一个长度指示字段, 除非已经根据 9.2.2.7 节通过 HE 字段指示 SDU 结束。

— 轮询检测比特的设置在 11.3.2.1.1 节中规定。

如果是重传:

— 重传 PDU 的 “Sequence Number” 和第一次传送时相同;

— 如果一个 piggybacked STATUS PDU 包括在该 AMD PDU 中或是包括在上次传送的 AMD PDU 中, 会造成长度指示字段值的修改。LI 字段的设置参见 9.2.2.8。

— 轮询检测比特的设置在 11.3.2.1.1 节中规定。

11.3.2.1.1 轮询检测比特的设置

如果轮询检测被触发并且未被禁止 (见 9.5 节), 发送实体将把轮询检测比特设置为 1。否则设为 0。

11.3.2.1.2 SDU 的分段和连接

当接收到一个SDU, RLC将对SDU进行分段以适合一个PDU的固定大小。分段被插入PDU的数据段中。把一个长度指示器加到每一个包含SDU最后一个分的PDU中, 即如果一个PDU不含有长度指示器, SDU将延续到下一个PDU。长度指示器指示了边界在PDU中的位置。所指示的边界后面的数据可以是一个新的SDU、填充或piggybacked STATUS信息。如果增加了填充或piggybacked STATUS, 则增加另外一个长度指示器除非填充大小是一个带有15-bit LI的PDU的一个字节, 见9.2.2.8和9.2.2.9。

11.3.2.2 AMD PDU的发送

当待传送或重传的 AMD PDU 准备就绪, 如果 AMD PDU 的“Sequence Number”小于 VT (MS) 或等于 VT (S) - 1, 并且该 PDU 未被本地暂停功能限制, 则这些 PDU 将允许向 MAC 传送。

- a) 发送实体通知 MAC 允许传送或重传的 PDU 的数量。
- b) AMD PDU 的内容根据 11.3.2.1 设置。
- c) 发送给 MAC 层请求数量的 AMD PDU。
- d) 重传 PDU 的优先权高于首次传送的 PDU。
- e) 根据 9.4 节的要求为每个 AMD PDU 更新除 VT (DAT) 外的其他状态变量的值。
- f) 如果轮询检测比特设为 1, 并且配置了定时器 Timer_Poll, 则根据 9.5 节要求开启定时器。

g) 不能在一个 TTI 内传输的 PDU (即: MAC 已经指出一些可用的 PDU 不能被传送), 应当根据 RRC 设置的丢弃配置进行缓冲。

11.3.3 AMD PDU 的接收

当接收到一个 AMD PDU (在 UE 侧, 如果“downlink AMD PDU size”没有设置, 则“downlink AMD PDU size”设置为该接收到 PDU 大小) 而且 UE 侧配置了固定 RLC PDU 大小, 接收实体将根据接收的 PDU 对 VR (R)、VR (H) 和 VR (MR) 进行更新 (参见 9.4 节)。

如果所接收的 PDU 包含一个设置为 1 的轮询检测比特, 接收实体将发起 STATUS PDU 传送过程。

如果配置了丢失 PDU 指示 (Missing PDU Indicator) 并且接收实体检测出丢失了一个 PDU, 则将启动 STATUS PDU 传输过程。

接收实体把接收到的 AMD PDU 重组为 RLC SDU。

如果配置了“按序传送”, 接收实体将通过 AM-SAP 将 RLC SDU 按发送顺序发送给高层。否则按照随机顺序发送 RLC SDU。

11.3.4 VT (DAT) 等于 MaxDAT

当 VT (DAT) 等于 MaxDAT, 如果配置了“No_discard after MaxDAT number of transmission”, 发送实体将根据 11.4 节发起 RLC 复位流程。如果配置了“SDU_discard after MaxDAT number of transmission”, 发送实体将根据 11.6 节发起显式信令的 SDU 丢弃流程 (SDU_discard with explicit signalling)。

11.3.5 异常情况

11.3.5.1 接收到一个接收窗口之外的 PDU

当接收到一个 PDU, 如果它的序列号 SN 不在 [VR (R), VR (MR)] 范围之内, 接收实体将丢弃 PDU。如果所接收的 PDU 包含一个设置为 1 的轮询检测比特, 接收实体将发起 STATUS PDU 传送过程。

11.3.5.2 定时器 Timer_Discard 超时

11.3.5.2.1 使用显式信令的 SDU 丢弃

当定时器超Timer_Discard时,发送实体将利用显式信令过程启动SDU丢弃功能。具体流程参见11.6.2。在由于向MAC发送RLC实体信息参数而启动了TFC选择交换的情况下,UE需要等待到它提供给MAC所请求的PDU之后,才能丢弃该SDU。

11.3.5.3 无效长度指示值

如果一个PDU的长度指示值大于PDU长度减去RLC头部大小,并且不是9.2.2.8中列出的预定义的值,接收实体将丢弃该PDU并且按丢失PDU处理。

11.3.5.4 AMD PDU保留的长度指示值

在本版本的协议中,如果递交给低层的AMD PDU中包含有保留值的长度指示,收端应该:

- 对此PDU不进行处理。

11.3.5.5 在接收窗内接收到同一AMD PDU多于一次时的处理(复制处理)

一旦接受到"Sequence Number"落在 $VR(R) \leq SN < VR(MR)$ 范围内的AMD PDU,而且此"Sequence Number"的AMD PDU已经收到过,接收端应该:

- 丢弃该 the AMD PDU;
- 在下一个状态报告中应该认为此AMD PDU已经成功接收;
- 如果被丢弃的AMD PDU的"polling bit"设置为"1":
- 发起STATUS PDU传输过程。

11.3.5.6 Buffer满

在某些情况下,Buffer满可以被预见,比如,当窗口被重新配置时,UE有存储器限制。

当Buffer满时,

- a) UE不再分段SDU组装成AMD PDU;
- b) UE将继续处理要到达的AMD PDU,尤其是SN等于VR(R)的AMD PDU,并进行正常的相关的协议过程,比如发送状态报告或重传等;
- c) UE可能会丢弃一些已收到的处于接收窗口内的AMD PDU,并认为这些被丢弃的AMD PDU未被收到。

11.3.5.7 无效

11.3.5.8 接收窗内多次收到相同的AMD PDU(重复检测处理)

当接收到AMD PDU,序号在 $VR(R) \leq SN < VR(MR)$ 范围内,而该AMD PDU之前已经收到过,接收方将:

- 丢弃该AMD PDU;
- 认为含有该序号的AMD PDU在下一个要发送的状态报告中会标注为正确接收;
- 若被丢弃的AMD PDU中轮询比特设置为1:
 - 初始传输STATUS PDU的过程;
 - 若该AMD PDU中捎带了STATUS PDU,那么按照11.5.3节内容执行操作。

11.3.5.9 满缓冲区行为

可以预料在某些条件下,例如当窗长被重配置时,UE会有存储限制。

在缓冲区满的情况下:

- 不要求UE根据11.3.2将RLC SDU分段成ADM PDU;

— UE 将:

- 能够处理到来的 AMD PDU (尤其能够处理并保存序列号=VR (R) 的 AMD PDU);
- 根据正常协议处理, 例如处理状态报告并执行重传。

— UE 可以将接收到的序列号处于接收窗口内的 ADM PDU 丢弃并认为丢弃的 ADM PDU 没有被接收到。

11.3.5.10 无效PDU大小

在 UE 侧, 如果配置了固定 RLC PDU 大小而且收到的 AMD PDU 大小与配置的下行 AMD PDU 大小不同, 接收端将: 忽略该 AMD PDU。

11.3.6 POLL SUFI 的发送

每次当包含 POLL SUFI 的 STATUS PDU 被调度发送时, 发送端将:

- 将序列号等于 VT (S) - 1 的 AMD PDU 的相应 VT (DAT) 值递增;
- 如果 VT (DAT) = MaxDAT: 则执行 11.3.3a 小节的动作。
- 否则: 通知底层 STATUS PDU 已经可以用于传输。

11.4 RLC 复位过程

11.4.1 概述

RLC 复位过程用来复位在确认模式下运行的两个对等 RLC 实体。RLC 复位的基本过程如图 30 所示。在复位过程中, 在 UTRAN 和 UE 的超帧数目 (HFN) 是同步的。两个用来加密的 HFN, 下行的 DL HFN 和上行的 UL HFN 需要同步。在复位过程中, RLC 传送实体使用最大的 UL HFN 和 DL HFN 在 UE 和 UTRAN 之间交换。如果有 AMD PDU 传送, HFN 和 “Sequence Number” 为 VT (S) - 1 的 AMD PDU 相关, 如果没有 AMD PDU 传送, “Sequence Number” 为 0。

RESET PDU 和 RESET ACK PDU 的优先级高于数据 AMD PDU。

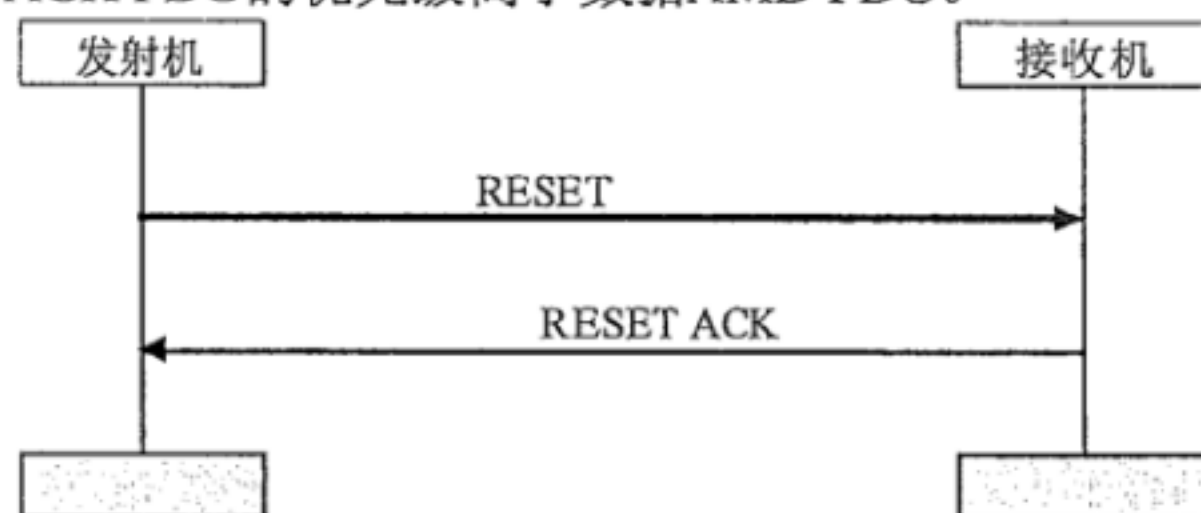


图30 RLC 复位过程

11.4.2 复位发起

发送端将会在检测到如下的任一复位触发条件时, 将启动相关的复位过程:

- 如果配置了 “No_discard after MaxDAT number of transmission” 功能, 且此时 VT (DAT) = MaxDAT;
- VT (MRW) = MaxMRW;
- 发端收到包含错误的序列号的 STATUS PDU/Piggybacked STATUS PDU (详见 10.1)

此时发送端将会:

- 停止发送任何 AMD PDU 或 STATUS PDU (即此时发送端只发送 RESET 或 RESET ACK PDU);
- 忽略掉收到的任何 AMD, 或者 STATUS/piggybacked STATUS PDU (即此时发送端只 “认识” RESET 或 RESET ACK PDU);
- 将 VT (RST) 加 1;

d) 如果此时 $VT(RST) < MaxRST$, 发端将向底层发送一个 RESET PDU 并同时启动定时器 Timer_RST。

e) 如果此时 $VT(RST) = MaxRST$, 发送端将中止正在进行的复位过程, 停止定时器 Timer_RST 并向高层报告不可恢复错误。

注: VT(RST) 是复位过程结束前 RESET PDU 被调度发送的次数。初始值为 0, VT(RST) 在每次发送 RESET PDU 时加 1, 在收到 RESET ACK PDU 时才重新置为 0。

注意: 如果此时 TFC 选择交换过程已启动将会把复位过程延迟到下一个 TTI 结束时再进行。

仅在如下情况时, 一个复位过程才会结束:

a) 发送端收到接收端发来的与发送的 RESET PDU 有相同 RSN 的 RESET ACK PDU;

b) 收到高层的重建或者释放 RLC 实体请求, 重建请求是为了请求重建 (用于收发端 RLC 实体, 或者发送端或接收端 RLC 实体)。

在复位过程时, 若发送端收到来自其对等实体的 RESET PDU (此种情况属于 Abnormal case, 详见 11.4.5), 此时不会阻止该复位过程。

11.4.2.1 RESET PDU 内容设置

RESET PDU 的大小将等于所允许的 PDU 大小中的一个。超帧数字段 (HFN) 设置成当前使用的最大 HFN (当由 UTRAN 发送 RESET 时用 DL HFN, 而当 UE 发送时用 UL HFN) RSN 字段应当设为 RESET PDU 的 “Sequence Number”。每次 AM 实体建立或重建后 (对于所有的 RLC 实体, 也包括只有发送端或者只有接收端重建的 RLC 实体) 的第一个 RESET PDU 的 “Sequence Number” 为 0。每发送一个新的 RESET PDU, “Sequence Number” 加 1, 但是重发 RESET PDU 时不增加顺序号。

11.4.3 接收方对 RESET PDU 的接收

如果 RESET PDU 不是 AM 实体建立或者重建之后收到的第一个 RESET PDU, 并且 RESET PDU 中的 RSN 取值和上一次收到的 RESET PDU 中的 RSN 取值相同, 接收实体将只会提交一个 RESET ACK PDU 给底层, 内容设置和上一次传输的 RESET ACK PDU 相同 (即 RLC 实体没有进行复位)。

如果该 RESET PDU 是 AM 实体建立/重建后接收到的第一个 RESET PDU, 或者该 PDU 的 RSN 的值和先前收到的 RESET PDU 中 RSN 值不同, 接收实体将:

a) 用一个 RESET ACK PDU 进行响应;

b) RESET ACK PDU 的内容设置参见 11.4.3.1;

c) 同时把 9.4 节中的变量 (除 VT(RST)) 复位到初始值;

d) 停止除 Timer_RST、Timer_Discard、Timer_Poll_Periodic 和 Timer_State_Periodic 外的所有定时器;

e) 复位可配置参数到原来的配置值;

f) AM RLC 接收端所有的 RLC PDU 将会被丢弃。AM RLC 发送端复位之前发送的 RLC SDU 将会被丢弃;

g) 如果发送端被请求, 则通知上层被丢弃的 SDU;

h) 将 HFN 设为接收到的 RESET PDU 中 HFNI 字段中的值;

i) UL HFN 和 DL HFN 加 1, 复位过程后的发送 PDU 和接收 PDU 将使用更新后的 HFN 值。

在由于向 MAC 发送 RLC 实体信息参数而启动了 TFC 选择交换的情况下, RLC 实体需要等到下一个 TTI 结束时才能丢弃发送侧的 RLC SDU。

11.4.3.1 RESET ACK PDU 内容设置

RESET ACK PDU的大小将等于所允许的PDU尺寸。RSN字段应当总是设置为接收到的RESET PDU中相同的值。超帧数指示字段(HFNI)将会设成等于当前应用的HFN(当由UTRAN发送RESET ACK时为DL HFN, 当从UE发送RESET ACK字段为UL HFN)。

11.4.4 发送者收到 RESET ACK PDU

当发送方在复位未决状态收到与对应 RESET PDU 有相同 RSN 的 RESET ACK PDU 时,

- a) 将 HFN 的值置为接收到的 RESET ACK PDU 的 HFNI 字段中的值;
- b) 将 9.4 节中的各个变量恢复为初始值;
- c) 停止 9.5 节中描述的除 Timer_Discard、Timer_Poll_Periodic 和 Timer_State_Periodic 外的所有定时器;
- d) 复位可配置的变量到其配置值;
- e) 丢弃 AM RLC 接收侧所有的 RLC PDU;
- f) 丢弃 AM RLC 实体发送侧复位之前发送的 RLC SDU;
- g) 如果发送端被请求, 则通知上层被丢弃的 SDU;
- h) UL HFN 和 DL HFN 加 1, 复位过程后的发送 PDU 和接收 PDU 将使用更新后的 HFN 值。

如果 AM RLC 实体没有发送 RESET PDU, 或当收到与相应 RESET PDU 不同 RSN 值的 RESET ACK PDU 时, RESET ACK PDU 被丢弃。

在数据传输就绪状态收到 RESET ACK PDU 时, RESET ACK PDU 被丢弃。

在由于向 MAC 发送 RLC 实体信息参数而启动了 TFC 选择交换的情况下, RLC 实体需要等到下一个 TTI 结束时才能丢弃发送侧的 RLC SDU。

11.4.5 VT(RST) = MaxRST

如果 VT(RST) = MaxRST, 发送实体将停止将要进行的 RLC 复位流程; 如果定时器 Timer_RESET 启动, 停止定时器; 并向高层报告不可恢复错误。

11.4.6 异常情况

11.4.6.1 Timer_RST超时

在复位过程终止之前, 如果 Timer_RST 超时, VT(RST) 加 1, 此时若 VT(RST) < MaxRST, 按原来设置重新发送 RESET PDU, 同时重启 Timer_RST。如果 VT(RST) = MaxRST, 按照 11.4.4a 处理。

11.4.6.2 无效

11.4.6.3 发送方对 RESET PDU 的接收

在复位未决状态收到 RESET PDU 时, 发送方用一个 RESET ACK PDU 响应:

步骤1) 发送方复位 9.4 节中的状态变量到它们的初值 (VT(RST) 除外);

步骤2) 停止 9.5 节中描述的所有的定时器 (Timer_Discard、Timer_Poll_Periodic 和 Timer_State_Periodic 除外);

步骤3) 复位可配置参数到其初始配置值;

步骤4) 丢弃 AM RLC 接收侧所有的 RLC PDU;

步骤5) 丢弃 AM RLC 实体发送侧复位之前发送的 RLC SDU;

步骤6) 如果发送端被请求, 则通知上层被丢弃的 SDU;

步骤7) UL HFN 和 DL HFN 加 1, 复位过程后的发送 PDU 和接收 PDU 将使用更新后的 HFN 值。

在由于向 MAC 发送 RLC 实体信息参数而启动了 TFC 选择交换的情况下, RLC 实体需要等到下一个

TTI 结束时才能丢弃发送侧的 RLC SDU。

11.5 STATUS PDU 传送过程

11.5.1 用途

状态报告传送过程用于在确认模式下运行的两个 RLC 对等实体之间状态信息的转移。状态报告传送的基本过程如图 31 所示。一个状态报告包括一个或几个 STATUS PDU。

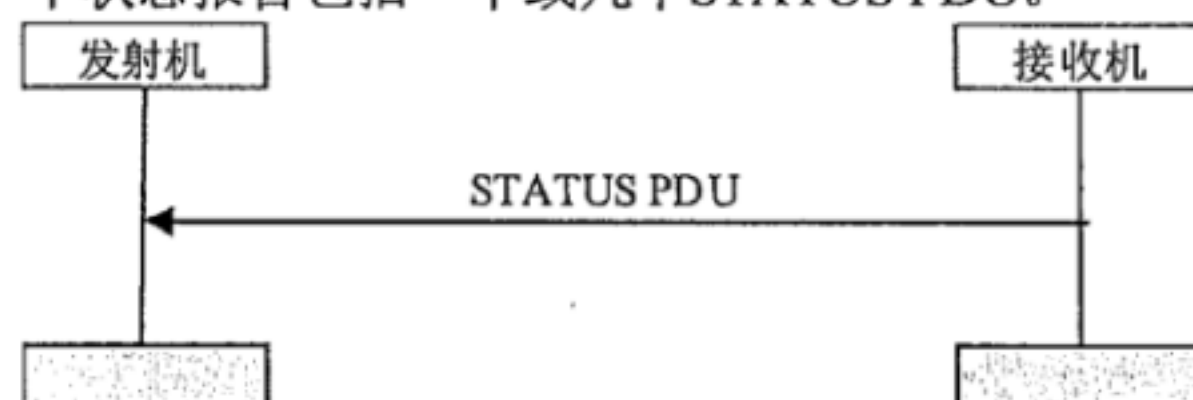


图 31 STATUS PDU 传送过程

如果上行链路配置了 2 条逻辑信道，只有确认状态报告、MRW ACK SUFI 和 WINDOW SUFI 会在第 2 条逻辑信道上发送；如果下行链路配置了 2 条逻辑信道，控制 PDU 可以在任何一条逻辑信道上发送。

STATUS PDU 优先级高于数据 AMD PDU。

11.5.2 发起

如果下列触发条件发生：

- a) 在接收到的 AMD PDU 中，轮询检测比特设置为 1。
- b) 使用了丢失 PDU 检测功能（配置了“Missing PDU 指示”），同时检测出 PDU 丢失。
- c) 使用了基于定时器的 STATUS PDU 传送，同时定时器“Timer_Status_Periodic”已经超时。
 - 接收实体将按照 9.7.2 节的说明启动状态报告。
- d) 空。
- e) 如果配置了可变 RLC PDU 大小并且收到了 POLL SUFI：
 - 认为发送端已经根据 9.4 节发送了 Poll_SN；
 - 根据 9.7.2 届响应触发。

11.5.2.1 Piggybacked STATUS PDU

如果一个 AMD PDU 包含填充部分，则接收实体可以在用 STATUS PDU 取代填充部分插入 PDU，作为 piggybacked STATUS PDU。piggybacked STATUS PDU 的发送和普通 STATUS PDU 的发送遵照相同的规则。

11.5.2.2 STATUS PDU 内容设置

STATUS PDU 的大小将与所允许的 PDU 尺寸中的一种相同。如果一个 STATUS PDU 不能容纳所有的信息，则在一个状态报告中需要传送的信息可以被划分为几个 STATUS PDU，但是 ACK SUFI 可以在一个状态报告的多于一个的 STATUS PDU 中出现。

如果状态报告流程被触发，接收方将：

- a) 如果未激活“STATUS Prohibit”功能，此时接收端将在 STATUS PDU 中填写如下信息：所有丢失的 AMD PDU 的否定应答；所有收到的 SN 至少到 VR (R) 的 AMD PDU 的肯定 ACK 应答。
- b) 在状态报告中哪个 SUFI 是随执行而定的。如果 MRW SUFI 没有发送，可以在状态报告中可选的包括 MRW SUFI。
- c) 如果 MRW_ACK SUFI 正在等待传送，可以在状态报告中可选的包括 MRW_ACK SUFI。
- d) 如果发送实体的发送窗要进行更新，可以在在状态报告中可选的包括 WINDOWS SUFI。

e) 如果此时允许使用的 STATUS PDU 的大小可以容纳此时要发送的所有的 SUFI, 则此时的状态报告将只使用一个 STATUS PDU, 即只组装一个 STATUS PDU 发送所有的 SUFI; 此时如果所有的 SUFI 还不能填满一个 STATUS PDU, 且如果此时该 PDU 不是用 ACK SUFI 结束, 则用 NO_MORE SUFI 结束该 PDU, 剩余空间使用填充 bit。

f) 如果此时允许使用的 STATUS PDU 的大小不能容纳此时要发送的所有的 SUFI, 则本次的状态报告将会使用多个 STATUS PDU, 注意此时, 不能有 SUFI 分割在两个 STATUS PDU 中, 指示同一个 AMD PDU 的 SUFI 应放在同一个 STATUS PDU 中。如果 SUFI 没有把 STATUS PSU 填满, 并且, 如果此时该 PDU 不是用 ACK SUFI 结束, 则用 NO_MORE SUFI 结束该 PDU, 剩余空间使用填充 bit。

在各个 SUFI 中 Bitmap SUFI 用于指示被接收和/或丢失的 PDU。List SUFI 和/或 Relative List SUFI 只用于指示丢失的 PDU。Acknowledgement SUFI 用于指示被接收的 PDU (SUFI 内容详见 9.2.2.11)。如果 PDU 的 $SN \geq VR(H)$, 则不给出任何信息。

11.5.2.3 向下层提交 STATUS PDU

接收端将:

- a) 通知下层被调度的要发送的 STATUS PDU;
- b) 向下层提交被请求传输的 PDU (包括 STATUS PDUs 或者 Piggybacked STATUS PDUs, 或者可选的 AMD PDUs);
- c) 如果配置了“基于定时器的 STATUS 传送”, 且此时若定时器 $Timer_Status_Periodic$ 过期, 则重启该定时器;
- d) 如果该 STATUS PDU 中包含 MRW SUFI, 则在发送时启动定时器 $Timer_MRW$ 。

11.5.3 发送实体对 STATUS PDU 的接收

当接收到 STATUS PDU/piggybacked STATUS PDU, 如果该 STATUS PDU 肯定确认了一个 RLC SDU, 需要的话发送实体将通知上层对等 AM RLC 实体已收到这一 RLC SDU。发送实体将根据收到的 STATUS PDU/piggybacked STATUS PDU 更新状态变量 $VT(A)$ 和 $VT(MS)$ 。

如果 STATUS PDU 包含对发送 PDU 的否定确认, 则将启动 AMD PDU 传送过程, 重传 AMD PDU。重传 PDU 的优先权高于新的 PDU。

如果一个 AMD PDU 在一个 STATUS PDU 中被否定确认了多次, 则只重传一次 AMD PDU。

如果 STATUS PDU 包括 MRW SUFI, 则按照 11.6.3 要求处理。

如果 STATUS PDU 包括 MRW_ACK SUFI, 则按照 11.6.4 要求处理。

如果 STATUS PDU 包括 WINDOW SUFI, 更新当前发送窗的大小和变量 $VT(WS)$ 。

11.5.4 无效

11.6 显式信令的 SDU 丢弃过程

11.6.1 概述

显式信令过程的 SDU 丢弃用来在两个对等的确认模式 RLC 实体丢弃 SDU 和传送丢弃信息。

发送实体对于一个在确认模式的 RLC 实体下的 SDU, 当达到 MaxDAT 重传次数或传输时间超过了预定的值 (定时器超时) 仍没被发送成功, 则该 SDU 将被丢弃。向接收者发送移动接收窗口指示 (MRW SUFI), 这样, 接收者将丢弃承载该 SDU 的 AMD PDU, 更新接收窗口。

利用显式信令进行 SDU 丢弃功能的基本过程如图 32 所示。

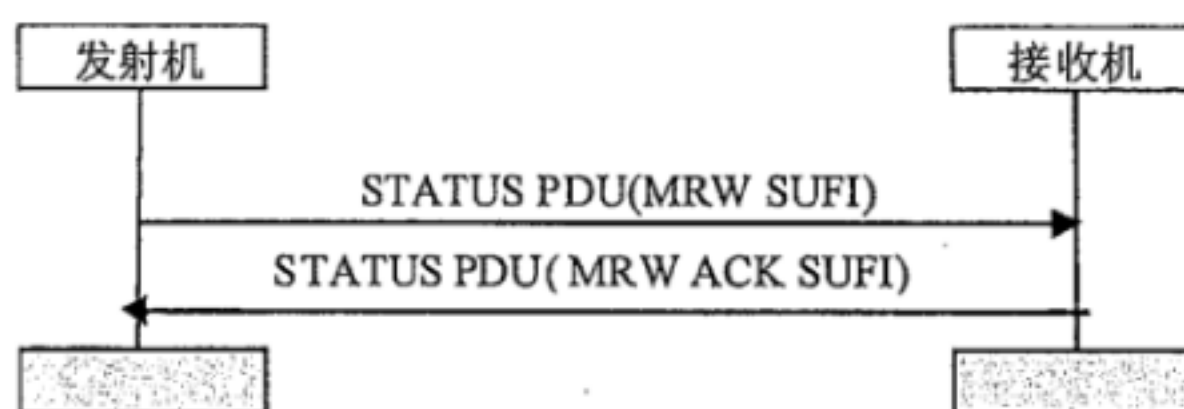


图32 利用显式信令的 SDU 丢弃

11.6.2 发起

满足下列条件之一时，发送者启动这个过程。

a) 使用显式信令进行基于定时器的 SDU 丢弃功能，定时器 *Timer_Discard* 对于一个 SDU 超时，此时该 SDU 的一个或多个段已发送到下层。

b) 使用显示信令进行基于定时器的 SDU 丢弃功能，定时器 *Timer_Discard* 对于一个 SDU 超时，并且配置了 *Send MRW*。

c) 对于一个 AMD PDU，配置了“MaxDAT 次传送后丢弃的机制”，重复传输次数达到 MaxDAT 次。

如果此时配置了“使用显式信令进行基于定时器的 SDU 丢弃功能”，发送方将丢弃到定时器 *Timer_Discard* 超时对应的 SDU 为止的全部 SDU。如果配置了“MaxDAT 次传送后丢弃的机制”，发送方将丢弃那些有分段在“Sequence Number” SN 属于 $[VT(A), X]$ 或“Length Indicators”指示 SDU 结束的 PDU 中的所有 SDU，其中 X 是 VT(DAT) 大于 MaxDAT 的 AMD PDU 的“Sequence Number”。如果需要的话，将这些被丢弃的 SDU 通知给上层。

发送方也将丢弃所有包含丢弃 SDU 分段或“Length Indicators”指示 SDU 结束的 AMD PDU，除非该 PDU 还包含定时器没有超时的 SDU 分段。

如果通知接收方被丢弃 SDU 超过 15 个，并且配置了“发送 MRW”，将最先 15 个 SDU 的丢弃信息组合在一个 MRW SUFI 中；把剩余的 SDU 丢弃信息包含在下一个 MRW SUFI 中，在下一个显示信令的 SDU 丢弃过程中发送；如果没配置“发送 MRW”，将全部 SDU 的丢弃信息组合在一个 MRW SUFI 中。

如果通知接收方被丢弃 SDU 少于等于 15 个，则不区分是否配置“发送 MRW”，将全部 SDU 的丢弃信息组合在一个 MRW SUFI 中。

当 MRW SUFI 准备就绪，则按顺序向 MAC 提交包含了该 MRW SUFI 的 STATUS PDU 或 piggybacked STATUS PDU。如果该 MRW SUFI 字段中的 $SN_MRWLENGTH > VT(S)$ ，则更新 $VT(S)$ 值为 $SN_MRWLENGTH$ 。同时启动定时器 *Timer_MRW*。

当显式信令的 SDU 丢弃过程正在进行时，如果触发一个新的 SDU 丢弃过程，那么在当前的 SDU 丢弃过程被一个终止条件终止之前将不发送新的 MRW SUFI。

11.6.2.1 空

11.6.2.2 状态 PDU 内容设置

发送方如果配置了“发送 MRW”，

a) 若 AMD PDU 中没有新的 SDU，且该 AMD PDU 包含了最后一个丢弃的 SDU 的 LI 信息，或者如果 AMD PDU 包含指示上一个丢弃的 SDU 结束的 HE 字段特殊值，则设置该 MRW SUFI 的最后一个 SN_MRWi 字段的值为该 AMD PDU 的序号+1，或者设置 HE 字段的特殊值来指示上一个丢弃的 SDU 结束， $NLENGTH$ 字段的值为“0000”。

b) 否则设置该 MRW SUFI 的最后一个 SN_MRWi 字段的值为该 AMD PDU 的序号。并在接收方被

丢弃的最后一个数据字节将是在第 NLENGTH 个长度指示字段指示的字节。

c) NLENGTH 字段的值应满足: 接收方丢弃的最后一个字节, 就是包含了最后丢弃的 SDU 的长度指示 (LI) 的 AMD PDU, 它的第 NLENGTH 个长度指示字段指示的字节。

d) 把 MRW SUFI 中其他的每个 SN_MRWi 字段的值设为包含第 i 个丢弃 SDU 的长度指示字段的 AMD PDU 的序号, 或者 HE 字段的特殊值来指示第 i 个丢弃 SDU 的结束。

发送方如果没配置“发送 MRW”,

a) 如果 AMD PDU 中没有新的 SDU, 且该 AMD PDU 包含了最后一个丢弃的 SDU 的 LI 信息, 或者如果 AMD PDU 包含 HE 字段的特殊值指示上一个丢弃 SDU 的结束, 则设置该 MRW SUFI 的最后一个 SN_MRWi 字段的值为该 AMD PDU 的序号+1, 或者设置 HE 字段的特殊值指示上一个丢弃 SDU 的结束, NLENGTH 字段的值为“0000”。

b) 否则设置该 MRW SUFI 的最后一个 SN_MRWi 字段的值为该 AMD PDU 的序号。并在接收方被丢弃的最后一个数据字节将是在第 NLENGTH 个长度指示字段指示的字节

c) NLENGTH 字段的值应满足: 接收方丢弃的最后一个字节, 就是包含了在接收方最后丢弃的 SDU 的长度指示 (LI) 的 AMD PDU, 它的第 NLENGTH 个长度指示字段指示的字节。

d) 并可选的把 MRW SUFI 中其他的每个 SN_MRWi 字段的值设为包含第 i 个丢弃 SDU 的长度指示字段的 AMD PDU 的序号, 或者设置 HE 字段的特殊值来指示接收端准备丢弃的第 i 个 SDU 结束。

如果 MRW SUFI 只包含一个 SN_MRWi, 并且 SN_MRWi 的值大于等于 $VT(A) + \text{Configured_Tx_Window_Size}$, 设置该 MRW SUFI 中的 LENGTH 字段的值为“0000”, 否则设置该 MRW SUFI 中的 LENGTH 字段的值为该 MRW SUFI 中 SN_MRWi 字段的数量。其中 SN_MRWi 字段的值应在 $VT(A)$ 和 $VT(A) + \text{Configured_Tx_Window_Size}$ 之间。

11.6.3 STATUS PDU 在接收侧的接收

当接收到一个包含 MRW SUFI 的 STATUS PDU/piggybacked STATUS PDU, 接收方将:

a) 如果接收 MRW SUFI 中的 LENGTH 字段为“0000”, 考虑 SN_MRWi 大于等于 $VR(R)$; 否则考虑 SN_MRWi 小于 $VR(MR)$;

b) 考虑其他的 SN_MRWi 按序排列, SN_MRWi 大于等于 SN_MRWi-1 ;

c) 递交直到并包括系列号为 SN_MRWLENGTH-1 的 PDU 在内的 PDU 中成功接收的 SDUs;

d) 丢弃直到并包括系列号为 SN_MRWLENGTH-1 的 PDU 在内的全部 PDU;

e) 如果接收 MRW SUFI 中的 NLENGTH 字段设为“0000”, 从“Sequence Number”为 SN_MRWLENGTH 的 AMD PDU 中的第一个数据字节开始重组 SDU; 否则丢弃该 PDU 中从第一个直到 (并包括) 第 NLENGTH 个长度指示的数据字节;

f) 如果接收 MRW SUFI 中的 LENGTH 字段不设为“0000”, 丢弃“Sequence Number”为 SN_MRWLENGTH 的 AMD PDU 中从第一个直到 (并包括) 第 NLENGTH 个长度指示的数据字节; 把后续的数据字节进行组合。

如果配置了“Send MRW”, 通知高层以前未传送过给高层的丢弃 SDU 或被其他 MRW SUFI 丢弃的全部 SDU。

根据所接收的 STATUS PDU/piggybacked STATUS PDU 更新状态变量 $VR(R)$, $VR(H)$ 和 $VR(MR)$ 。

根据 11.6.3.1 组合 MRW_ACK SUFI, 并启动一个包含一个 MRW_ACK SUFI 的状态报告通知发送方。

11.6.3.1 STATUS PDU 内容设置

接收方将把 MRW_ACK SUFI 包含在下一个 STATUS PDU/piggybacked STATUS PDU 中传送给发送方, 参见 11.5.2 节;

MRW_ACK SUFI 中的 SN_ACK 字段设置成接收到 MRW SUFI 后更新的 VR (R) 值;

如果 SN_ACK 字段等于接收到的 MRW SUFI 中 SN_MRLENGTH, 设置 MRW_ACK SUFI 中的 N 字段等于 SN_MRLENGTH 的值; 否则 N 字段设为 “0000”。

11.6.4 终止

在如下情况下发送端显式信令丢弃过程将被终止:

- a) 接收到一个包含一个 MRW_ACK SUFI 并且 SN_ACK > 发送的 MRW_SUFI 中 SN_MRW 值的 STATUS PDU/piggyback STATUS PDU, 并且 MRW_ACK SUFI 中的 N 字段等于 0;
- b) 接收到一个包含一个 MRW_ACK SUFI 的 STATUS PDU/piggyback STATUS PDU。这个 MRW_ACK 的 SN_ACK = SN_MRLENGTH 并且 N 等于在发送端发送的 MRW_SUFI 中的 NLENGTH 的值;
- c) 接收到一个包含一个 MRW_ACK SUFI 的 STATUS PDU/piggyback STATUS PDU。该 PDU 指示到 “Sequence Number” 为 (发送 MRW SUFI 的 SN_MRLENGTH-1) 为止的全部 PDU 被对端实体接收到或被丢弃掉。

如果以上条件有一条符合, Timer_MRW 终止, 丢弃过程终止, 并根据 STATUS PDU/piggyback STATUS PDU 更新 VT (A) 和 VT (MS)。

发送方不用通知高层被请求丢弃的 SDU。

11.6.5 VT (MRW) = MaxMRW

当 VT (MRW) 达到 MaxMRW, 丢弃过程终止, Timer_MRW 终止并发起 RLC 复位过程。

11.6.6 定时器 Timer_MRW 的超时

如果在丢弃过程终止之前定时器 “Timer_MRW” 超时, VT (MRW) 加 1。如果 VT (MRW) 小于 MaxMRW, 那么重传 MRW SUFI, 并且根据 9.5 节重新启动定时器 “Timer_MRW”。重传的 MRW SUFI 包含在一个新的状态报告中, 如果该报告包含除 MRW_SUFI 之外其他的 SUFI, 则由这些 SUFI 指示的状态信息将被更新。状态报告将被包含在一个 STATUS PDU 或 piggyback STATUS PDU 中传送。

11.6.7 异常情况

11.6.7.1 失效的/损坏的 MRW SUFI 的接收

如果接收到的 MRW SUFI 包含的有关接收窗的信息是过时的 (接收者的窗口已经前移, 即已不是 MRW 命令所指示的), 那么 MRW 命令将被丢弃, 并且发送包含 SUFI MRW_ACK 的状态报告, SN_ACK 字段置为 VR (R) 和 N 字段的值置 0。该状态报告在下一个 STATUS PDU 或 piggyback STATUS PDU 中传送。

11.6.7.2 废弃的/损坏的 MRW_ACK SUFI 的接收

MRW_ACK SUFI 将会在如下情况之一下被发送方丢弃:

- a) 如果没有显式信令的 SDU 丢弃过程正在进行;
- b) 如果接收到的 MRW_ACK SUFI 中 SN_ACK 小于被发送的 MRW SUFI 中 SN_MRLENGTH。
- c) 如果接收到的 MRW_ACK SUFI 中的 SN_ACK 字段等于被发送的 MRW SUFI 的 SN_MRLENGTH, 并且在接收的 MRW_ACK SUFI 中字段 N 的值不等于被发送的 MRW SUFI 的 NLENGTH;

d) 如果接收到的 MRW_ACK SUFI 中的 SN_ACK 字段大于发送的 MRW SUFI 的 SN_MRWLENGTH, 并且接收的 MRW_ACK SUFI 的字段 N 不等于 “0000”。

11.7 流量控制

11.7.1 概述

发送端接收到接收端发送的Window Size SUFI信息时, 根据该信息要求更新发送窗口大小, 从而完成流量控制的功能。由于Window Size SUFI信息存在丢失的可能, 导致更新发送端发送窗口失败, 尤其是在需要增加发送端的发送窗口时, 导致无线链路的使用效率降低, 为此需要解决该缺陷。针对该缺陷, 可采用如下几种更新发送端发送窗口方案或其他方案。

11.7.2 增加 Window Size ACK SUFI 作为 Window Size SUFI 的确认信息的方案

增加一个新的SUFI即Window Size ACK SUFI, 作为收到Window Size SUFI的确认信息。如图33所示。以下Window_ACK SUFI作为Window Size ACK SUFI的简称, Window SUFI作为 Window Size SUFI的简称。Window_ACK SUFI的具体构成如图33所示。

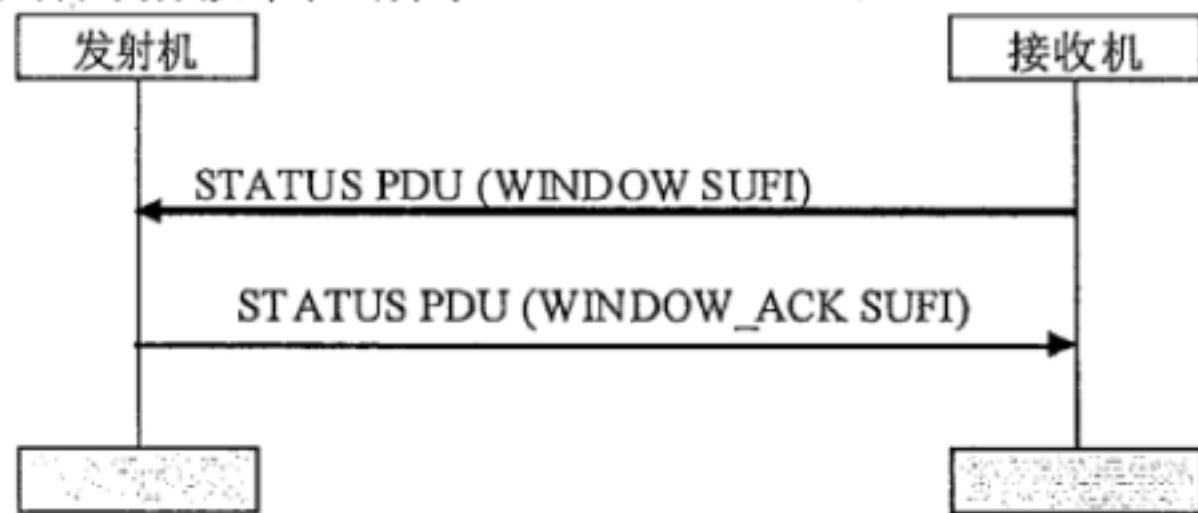


图33 Window_ACK SUFI 作为 Window SUFI 的确认信息

11.7.2.1 Window_ACK SUFI的构成

Window_ACK SUFI的描述如下在9.2.2.11的SUFI的基础上, 做如下扩充:

比特	描述
0000	不再有数据 (NO_MORE)
0001	窗口大小 (WINDOW)
0010	确认 (ACK)
0011	列表 (LIST)
0100	比特位图 (BITMAP)
0101	相关列表 (Rlist)
0110	移动接收窗口 (MRW)
0111	移动接收窗口确认 (MRW_ACK)
1000	窗口大小确认 (Window_ACK)
1001~1111	保留 (在这个协议版本中, 这种编码的 PDU 是无效的)

Window_ACK SUFI作为收到Window SUFI后的响应信息。增加Window SUFI发送的可靠性。Window_ACK SUFI的构成如图34所示。

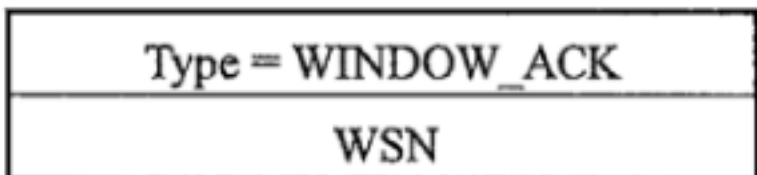


图34 状态 PDU 中的 WINDOW_ACK 字段

WSN

长度: 12bit。

Window_ACK SUFI 作为收到 Window SUFI 后的响应信息。WSN 的值置为最新收到的 Window SUFI

中的 WSN 值。WSN 的取值范围是 $[0, 2^{12}-1]$ 。无论收到的 Window SUFI 中的 WSN 的值是多少，只要在这个范围内，就把该 WSN 作为 Window SUFI 中的 WSN 值。

11.7.2.2 相关定时器与变量

为了保证 RLC 接收部分发送 Window SUFI 后能被对端的发送部分接收到，需要在本端接收部分增加定时器和相关变量。具体说明如下：

a) VR(WIN) - WINDOW 命令发送状态变量

该状态变量用来作为 WINDOW 命令发送次数的计数器。每当定时器 Timer_Window 超时，VR(WIN) 增加 1，当更新发送端发送窗口的过程终止后，VR(WIN) 将被重新初始化。

该变量的初始值为 0。

b) Timer_Window

该定时器用来触发包含 WINDOW SUFI 的状态报告的重发，由 UE 侧和 UTRAN 侧上层自行设定各自 RLC 实体的 Timer_Window 参数。

在 UE 侧，当低层指示包含 WINDOW SUFI 的状态 PDU 被成功或未成功发送后，该定时器启动；在 UTRAN 侧，当包含 WINDOW SUFI 的状态 PDU 被发送到低层后，该定时器启动。

每当该定时器超时，WINDOW SUFI 要被重新发送，并且定时器重新启动。当更新发送端窗口的过程终止条件满足的时候，该定时器才停止（参见 11.7.2.7 节）。

c) MaxWIN

WINDOW 命令的最大发送次数为 MaxWIN，该参数表示 VR(WIN) 的计数上限。由 UE 侧和 UTRAN 侧上层自行设定各自 RLC 实体的 MaxWIN 参数。当 VR(WIN) = MaxWIN，可以根据不同厂商 RLC 实体的需要决定是否发起 RLC 的 RESET 过程。

11.7.2.3 发起

当接收端检测到需要更新发送端的发送窗口的时候，将发送一个包含 WINDOW SUFI 的状态 PDU 给发送端。

— 启动定时器 Timer_Window 见 11.7.2.2 节相关描述。

当前次更新发送窗口的过程还没有结束（结束的条件参考 11.7.2.5）。又发生了一次新的更新发送窗口的过程，那么首先：

— 停止定时器 Timer_Window；

— 初始化变量 VR(WIN) 为 0。

然后再发起一次新的更新发送端的发送窗口的过程，发送包含最新的 WINDOW SUFI 的状态 PDU。

— 启动定时器 Timer_Window。

11.7.2.4 发送端收到包含 WINDOW SUFI 的状态 PDU

当收到一个包含 WINDOW SUFI 的状态 PDU，发送端将：

— 更新发送端的发送窗口如 9.2.2.11.3 节相关描述；

— 发送一个包含 WINDOW_ACK SUFI 的状态 PDU 作为回应，WINDOW_ACK SUFI 中的 WSN 值置为最近一次收到的 WINDOW SUFI 中的 WSN 值。

11.7.2.5 终止

当接收端收到的 WINDOW_ACK SUFI 中的 WSN 值与最近一次发送的 WINDOW SUFI 中的 WSN 值相等，终止更新发送端发送窗口的过程：

- 停止定时器Timer_Window;
- 初始化变量VR(WIN)为0。

11.7.2.6 达到最大发送次数

如果 $VR(WIN) = MaxWIN$, 发送 WINDOW SUFI 的 RLC 端将会:

- 终止更新发送端发送窗口的过程;
- 若定时器 Timer_Window 在运行, 终止该定时器;

根据各厂商 RLC 实现的情况决定是否发起 RLC 的 RESET 过程(若发起 RLC 的 RESET 过程, 参见 11.4 节)。

11.7.2.7 Timer_Window定时器超时

当更新发送端发送窗口过程终止以前, 定时器Timer_Window超时, 发送WINDOW SUFI 的RLC端将会:

- 将 $VR(WIN)$ 加 1;
- 若 $VR(WIN) < MaxWIN$;
 - 置 WINDOW SUFI 为上次发送的信息;
 - 在一个新的状态报告中包含 WINDOW SUFI (若状态报告中包含有其他的 SUFI, 这些内容需要被更新);
 - 通过状态 PDU 发送该状态报告或者在 AMD PDU 中通过捎带的形式发送该状态报告;
 - 在此次更新发送端窗口的过程中, 重新启动定时器 Timer_Window;
- 若 $VR(WIN) = MaxWIN$:
 - 按 11.7.2.6 节论述的过程处理。

11.7.2.8 异常情况

发送端无论收到的WINDOW SUFI是否有效, 都需要给对端回一个WINDOW_ACK SUFI, 其中的WSN值置为刚收到的WINDOW SUFI中的WSN值。

接收端如果收到的WINDOW_ACK SUFI中的WSN值和本端发送的WINDOW SUFI中的WSN值不一致, 忽略该WINDOW_ACK SUFI, 继续进行本次更新发送端发送窗口的过程。

11.7.3 接收端通过检测接收缓存更新发送窗口的方案

本方案的原理为: 接收端检测Window Size SUFI的触发条件, 若满足触发条件, 则发送Window Size SUFI到发送端。等过一段合适的时间(可设置)后, 接收端再检测一下触发条件, 若此时已不满足触发条件, 证明发送端Window Size SUFI已收到(流控机制已起效果), 则不需再发送Window Size SUFI; 若此时仍满足触发条件, 发送端有可能未收到Window Size SUFI(流控机制未达到预期效果), 则仍需继续发送Window Size SUFI。

完整的解决方案如下:

Window Size SUFI基本都是根据接收缓存的情况来触发的。若接收缓存的占有率大于或等于某一指标A(为便于阐述, 称为第一缓存门限, 可以根据具体的业务或应用场景灵活设置, 如A=70%), 接收端可发送Window Size SUFI信息给发送端, 减小发送窗口。反之, 若接收缓存的占有率小于或等于某一指标B(为便于阐述, 称为第二缓存门限, 可以根据具体的业务或应用场景灵活设置, 如B=30%), 接收端可发送Window Size SUFI信息给发送端, 增大发送窗口。

注: 本方案的参数设置均由RLC实体完成。

本方案的具体步骤如下：

步骤1) 检查接收缓存的占有率：

若大于或等于某一指标 A，需要发送端减小发送窗口，转到 2；

若小于或等于某一指标 B，需要发送端增大发送窗口，转到 4；

其他情况，转到 1。

步骤2) 接收端发送 Window Size SUFI 信息，启动定时器 T1（可以根据具体的业务或应用场景灵活设置，如 T1=60ms）。

步骤3) 等 T1 到后，再检查接收缓存的占有率，

若仍大于或等于 A，则证明流控机制未起明显作用（有可能是 Window Size SUFI 传输丢失），需要再次计算并发送 Window Size SUFI 信息并启动定时器 T。转到 3。

若小于 A，则证明流控已起作用。不再发送 Window Size SUFI 信息，转到 1。

步骤4) 接收端发送 Window Size SUFI 信息，启动定时器 T2（可以根据具体的业务或应用场景灵活设置，如 T2=80ms）。

步骤5) 等 T2 到后，再检查接收缓存的占有率，

若仍小于或等于 B，则证明流控机制未起明显作用（有可能是 Window Size SUFI 传输丢失），需要再次计算并发送 Window Size SUFI 信息并启动定时器 T2。转到 5。

若大于 B，则证明流控已起作用。不再发送 Window Size SUFI 信息，转到 1。

具体流程参如图 35 所示。

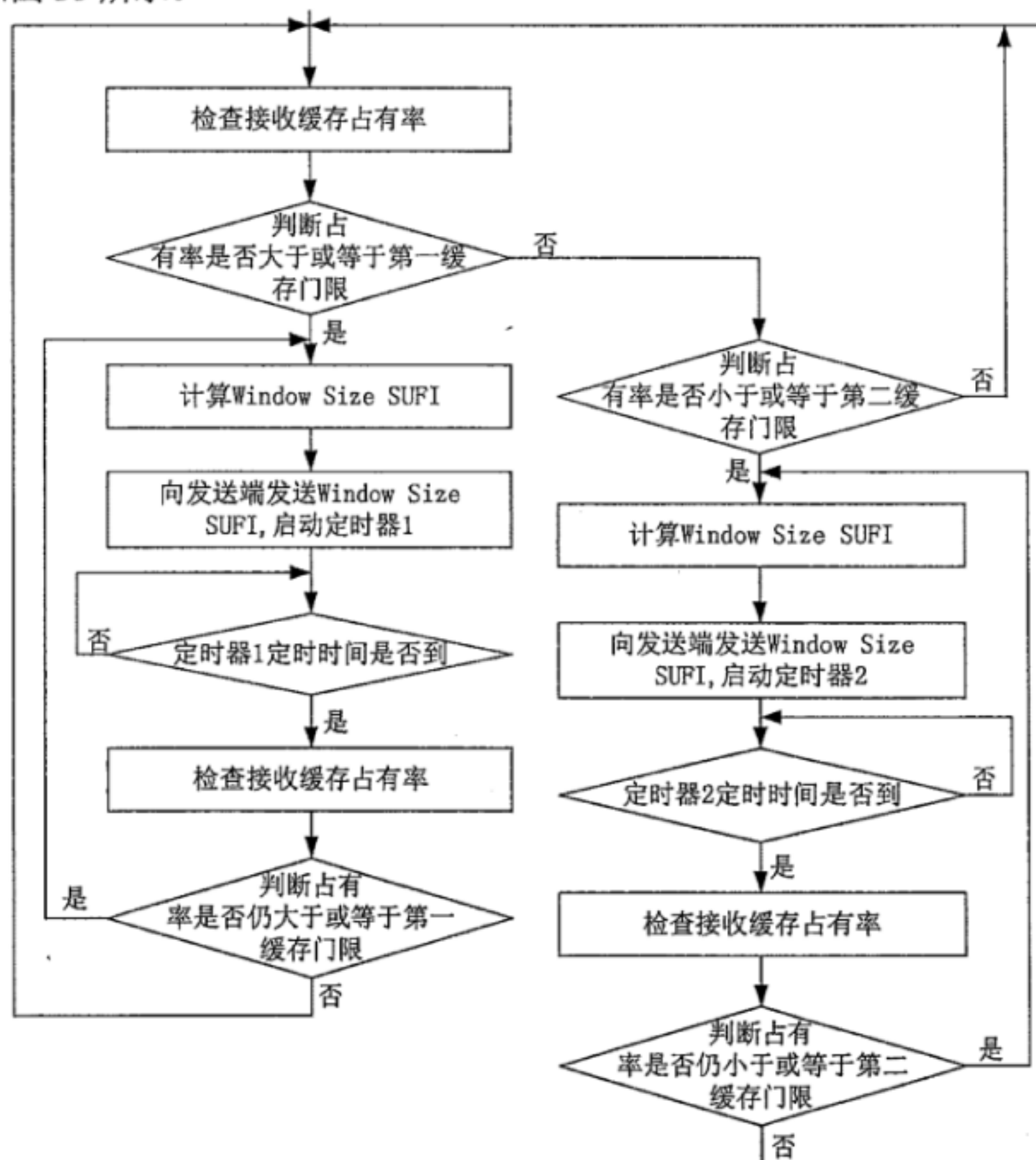


图35 接收端通过检测接收缓存更新发送窗口的流程

11.7.4 周期发送包含 Window Size SUFI 的状态报告的方案

接收端第一次检测到有 Window SUFI 需要发送后, 启动周期发送的机制, 每隔 Timer_Window_Periodic 发送一个包含 Window SUFI 的状态报告, 直到该定时器终止。该定时器的终止发生在 Am RLC 实体重建, 或收到对端发起的 RESET 过程, 或应上层要求终止。该定时器停止后, 当接收端再次检测到有 Window SUFI 需要发送时, 再次启动该定时器。发送 Window SUFI 时, 所填的 WSN 值按如下原则确定:

- 若接收端无需调整发送端的发送窗口, 将 WSN 值设为 4095;
- 若接收端需要调整发送端的发送窗口, 将 WSN 值设为相应的值。

定时器 Timer_Window_Periodic 按如下要求处理:

只有当上层要求周期发送 Window Size SUFI 时, 才使用该定时器。由 UE 侧和 UTRAN 侧上层自行设定对应 RLC 实体的 Timer_Window_Periodic 参数。当包含 WINDOW SUFI 的状态 PDU 被发送到低层以后, 启动该定时器。

当该定时器超时, 一个新的包含 WINDOW SUFI 的状态 PDU 需要被发送, 然后该定时器重启。该定时器可以被上层终止。当收到对端发起的 RLC RESET 过程, 或 RLC 实体发生重建过程, 该定时器被终止。当该定时器终止后, 如果又有 WINDOW SUFI 需要被发送, 当该 WINDOW SUFI 被发送到低层后, 该定时器又将启动。当 UE 进入 CELL_PCH 或 URA_PCH 状态, 上层需要终止该定时器。

11.7.5 在每个状态报告中都包含 Window Size SUFI 的方案

接收端在生成的每个状态报告中都包含 Window Size SUFI, 发送 Window SUFI 时, 所填的 WSN 值按如下原则确定:

- 若接收端无需调整发送端的发送窗口, 将 WSN 值设为 4095;
- 若接收端需要调整发送端的发送窗口, 将 WSN 值设为相应的值。

11.7.6 发送端通过检测发送变量自行恢复发送窗口的方案

发送端定期检测变量 Configured_Tx_Window_size, VT(S) 和 VT(A) 的关系加以推断接收端的缓存状况, 从而决定是否恢复发送端的发送窗口。变量关系如下:

$$K = \frac{(4096 + VT(S) - VT(A) \bmod 4096)}{\text{Configured_Tx_Windows_size}} \times 100, K \leq N$$

当 K 低于某个门限 N 时, 将 VT(WS) 恢复为 Configured_Tx_Window_size。否则维持 VT(WS) 不变。

11.7.7 通过数据 PDU 来保证 Window Size SUFI 可靠传输的方案

本方案的主要思想是: 在数据 PDU 中捎带 Window Size sufi 或者构造一个空数据 PDU 来捎带 Window Size sufi, 从而保证 Window Size sufi 的可靠传输。具体描述如下:

每次当 UE 接收到数据, 如果发现有属于接收窗口内的 PDU 丢失, 则比较丢失 PDU 数量和当前 UE 接收端剩余可用存储空间, 如果丢失 PDU 所需占用的空间接近或者等于接收端剩余可用空间, 则发送 Window Size sufi, WSN 的值设置为 UE 当前的 VR(H) - VR(R)。当所有丢失 PDU 都已经被成功处理 (包括被成功重传或者被发送方指示丢弃), 则发送 Window Size sufi (WSN 等于 4095), 通知发送方将发送窗口增加至初始配置值。

减少发送窗口可以分两种情况考虑:

— 记录所有丢失PDU的“Sequence Number”，如果UE同时也有数据发送到RNC，则将含有Window Size sufi的状态PDU捎带在某个数据PDU中发送（如果存在这样的PDU的话），同时记录这个PDU的“Sequence Number”和Window Size sufi中参数WSN的值。当接收到的来自RNC的状态PDU表明了这一被记录“Sequence Number”的PDU已经被正确接收，则说明发送的发送窗口已经得到调整；否则重新发送Window Size sufi。

— 记录所有丢失PDU的“Sequence Number”，如果UE没有数据发送到RNC，或者不存在某个上行PDU有足够的空余空间来捎带该状态PDU，那么构造一个空的数据PDU，里面没有实际数据，将UE欲发送的状态PDU捎带进去，同时记录这个PDU的“Sequence Number”和Window Size sufi中参数WSN的值。当接收到的来自RNC的状态PDU表明了这一被记录“Sequence Number”的PDU已经被正确接收，则说明发送的发送窗口已经得到调整；否则重新发送Window Size sufi。

当UE发送了减小发送窗口的Window Size sufi之后，若再收到来自RNC的发送数据并且如果收到的PDU数量大于所纪录的WSN值，那么再次发送Window Size sufi。

增加发送窗口可以分两种情况考虑：

— 如果UE同时也有数据发送到RNC，则将含有Window Size sufi的状态PDU捎带在某个数据PDU中发送（如果存在这样的PDU的话），同时记录这个PDU的“Sequence Number”。当接收到的来自RNC的状态PDU表明了这一被记录“Sequence Number”的PDU已经被正确接收，则说明发送的发送窗口已经得到调整；否则重新发送Window Size sufi。

— 如果UE没有数据发送到RNC，或者不存在某个上行PDU有足够的空余空间来捎带状态PDU，那么可以构造一个空的PDU，里面没有实际的数据，将UE欲发送的状态PDU捎带进去，同时记录这个PDU的“Sequence Number”。当接收到的来自RNC的状态PDU表明了这一被记录“Sequence Number”的PDU已经被正确接收，则说明发送的发送窗口已经得到调整；否则重新发送Window Size sufi。

具体流程如下所述：

步骤1) 侧面(UE)RLC接收来自对端(RNC)的数据，有转2)，否则转1)；

步骤2) RLC检查丢失PDU（不在接收窗口之内的除外）的情况，有转3)，否则转1)；

步骤3) 比较丢失PDU（如果有并且属于接收窗口之内的话）所需要占用的空间和接收端剩余可用存储空间，如果接近或者等于，则转4)，否则转1)；

步骤4) 记录所有丢失PDU的“Sequence Number”，构造含有Window Size sufi的状态PDU，WSN等于UE当前的VR(H)-VR(R)，记录WSN值。转5)；

步骤5) 如果有需要发送（到RNC）的PDU，并且满足捎带条件，则捎带Window Size sufi发送，记录此特定PDU“Sequence Number”，转7)，否则转6)；

步骤6) 构造一个PDU（其中数据部分为空），将含有Window Size sufi的状态PDU捎带发送，

步骤7) 记录此特定PDU“Sequence Number”，转7)；

步骤8) 分析接收到（来自RNC的）PDU，如果所有记录的丢失PDU已经全部成功重传，则转8)，否则转9)；

步骤9) 如果（来自RNC的）状态PDU指明记录的特定PDU已经被正常接收，转10)，否则重新发送特定PDU（注意这时Window Size sufi就不需要发送了，即如果特定PDU是一个数据加捎带的PDU，则去掉捎带部分；如果特定PDU仅仅只有捎带部分，则将捎带改成填充发送），转1)；

步骤10) 如果(来自 RNC 的)状态 PDU 指明记录的特定 PDU 已经被正常接收,转 7),否则重新发送 Window Size sufi,转 5);

步骤11) 构造构造含有 Window Size sufi 的状态 PDU, WSN 等于当前丢失 4095,转 11);

步骤12) 如果有需要发送(到 RNC)的 PDU,并且满足捎带条件,则捎带 Window Size sufi 发送,记录此特定 PDU “Sequence Number”,转 13),否则转 12);

步骤13) 构造一个 PDU (其中数据部分为空),将含有 Window Size sufi 的状态 PDU 捎带发送,

步骤14) 记录此特定 PDU “Sequence Number”,转 13);

步骤15) 如果(来自 RNC 的)状态 PDU 指明记录的特定 PDU 已经被正常接收,转 1),否则重新发送;

步骤16) Window Size sufi,转 11)。

参 考 文 献

- [1] 3GPP TS 25.221—physical channels and mapping of transport channels on physical channels (TDD)
 - [2] 3GPP TS 25.222—multiplexing and channel coding (TDD)
 - [3] 3GPP TS 25.223—spreading and modulation (TDD)
 - [4] 3GPP TS 25.224—physical layer procedure (TDD)
 - [5] 3GPP TS 25.225—physical layer measurement (TDD)
 - [6] 3GPP TS 25.331—radio resource control protocol specification
 - [7] 3GPP TS 25.301—radio interface protocol architecture
 - [8] 3GPP TS 25.302—services provided by the physical layer
-

中华人民共和国
通信行业标准

2GHz TD-SCDMA 数字蜂窝移动通信网
多载波高速分组接入 Uu 接口层 2 技术要求
第 2 部分: RLC 协议

YD/T 2856.2-2015

*

人民邮电出版社出版发行
北京市丰台区成寿寺路 11 号邮电出版大厦
邮政编码: 100164
北京康利胶印厂印刷
版权所有 不得翻印

*

开本: 880×1230 1/16 2016 年 2 月第 1 版
印张: 5.25 2016 年 2 月北京第 1 次印刷
字数: 142 千字

15115 · 755

定价: 55 元

本书如有印装质量问题, 请与本社联系 电话: (010)81055492