

YD

中华人民共和国通信行业标准

YD/T 1545.2-2007

2GHz WCDMA 蜂窝移动通信网 Un 接口层二技术要求（第一阶段） 第 2 部分：RLC

Technical Specification for Uu Interface of 2GHz WCDMA Digital
Cellular Mobile Communication Network Layer 2 (Phase I)
Part 2: RLC

(3GPP R99 TS25.322 v3.d.0,MOD)

2007-05-16 发布

2007-05-16 实施

中华人民共和国信息产业部 发布

目 次

前 言	II
1 范围	1
2 规范性引用文件	1
3 缩略语	1
4 概述	2
4.1 内容	2
4.2 RLC 子层结构概述	3
5 功能	8
6 提供给上层的业务	9
6.1 业务/功能在逻辑信道上的映射	10
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	31
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	47
10.1 错误序列号	47
10.2 不一致的状态指示	47
10.3 无效的 PDU 格式	47
11 基本过程	47
11.1 透明模式数据(TMD)传送过程	47
11.2 非证实模式数据(UMD)传送过程	48
11.3 证实模式数据(AMD)传送过程	50
11.4 RLC 复位过程	53
11.5 STATUS PDU 传送过程	55
11.6 显式信令过程的 SDU 丢弃	57
11.7 流量控制	60
参考文献	68

前 言

本部分是《2GHz WCDMA数字蜂窝移动通信网 Uu接口层二技术要求（第一阶段）》标准之一，该标准分为2个部分：

1.YD/T1545.1-2007 2GHz WCDMA数字蜂窝移动通信网 Uu接口层二技术要求（第一阶段）第1部分：MAC；

2. YD/T1545.2-2007 2GHz WCDMA数字蜂窝移动通信网 Uu接口层二技术要求（第一阶段）第2部分：RLC；

本部分等同采用3GPP TS25.321 V3.e.0《媒体接入控制（MAC）协议规范》。

本部分是2GHz WCDMA 数字蜂窝移动通信网系列标准之一，该系列标准的结构和名称预计如下：

- 1.YD/T1374-2007 2GHz TD-SCDMA/WCDMA 数字蜂窝移动通信网 Iu 接口技术要求（第二阶段）
- 2.YD/T1543-2007 2GHz WCDMA 数字蜂窝移动通信网 Iu 接口技术要求（第一阶段）
- 3.YD/T1544-2007 2GHz WCDMA 数字蜂窝移动通信网 Uu 接口物理层技术要求（第一阶段）
- 4.YD/T1545-2007 2GHz WCDMA 数字蜂窝移动通信网 Uu 接口层二技术要求（第一阶段）
- 5.YD/T1546-2007 2GHz WCDMA 数字蜂窝移动通信网 Uu 接口 RRC 层技术要求（第一阶段）
- 6.YD/T1547-2007 2GHz WCDMA 数字蜂窝移动通信网 终端设备技术要求（第二阶段）
- 7.YD/T1548-2007 2GHz WCDMA 数字蜂窝移动通信网 终端设备测试方法（第二阶段）
- 8.YD/T1549-2007 2GHz WCDMA 数字蜂窝移动通信网 Iur 接口测试方法（第一阶段）
- 9.YD/T1550-2007 2GHz WCDMA 数字蜂窝移动通信网 Iub 接口技术要求（第一阶段）
- 10.YD/T1551-2007 2GHz WCDMA 数字蜂窝移动通信网 Iub 接口测试方法（第一阶段）
- 11.YD/T1552-2007 2GHz WCDMA 数字蜂窝移动通信网 无线接入网络设备技术要求（第一阶段）
- 12.YD/T1553-2007 2GHz WCDMA 数字蜂窝移动通信网 无线接入网络设备测试方法（第一阶段）

随着技术的发展，还将制定后续的相关标准。

随着技术的发展，还将制定后续的相关标准。

本部分修改采用3GPP TS 25. 322 v3.d.0《无线链路控制(RLC)协议规范》。

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：信息产业部电信研究院、上海贝尔阿尔卡特股份有限公司、中兴通讯股份有限公司

本部分主要起草人：李文宇、林 辉、卓天真、徐 菲、张建林、张义成、王之曦

2GHz WCDMA 数字蜂窝移动通信网

Uu 接口层二技术要求（第一阶段）第 2 部分：RLC

1 范围

本部分规定了 2GHz WCDMA (IMT-DS) 系统的空中接口层 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 WCDMA 数字蜂窝移动通信网的 Uu 接口的研发、生产和检测。

2 规范性引用文件

下列文件中的条款通过本部分中的引用而成为本部分的条款。凡是注日期的引用文件, 其随后所有的修改单 (不包括勘误的内容) 或修订版均不适用于本部分, 然而, 鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件, 其最新版本适用于本部分。

3 缩略语

下列缩略语适用于本部分:

ARR	Automatic Repeat Request	自动重复请求
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
CCTCH	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
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
DSCH	Downlink Shared Channel	下行共享信道
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	Handover	切换
ITU	International Telecommunication Union	国际电信联盟
kbit/s	kilo-bit 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
MS	Mobile Station	移动台
MM	Mobility Management	移动性管理
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
SCCCH	Shared Channel Control Channel	共享信道控制信道
SN	Sequence Number	序号
TCH	Traffic Channel	业务信道
TDD	Time Division Duplex	时分双工
TFI	Transport Format Indicator	传输格式指示
TTI	Transmission Time Interval	传送时间间隔
TFCI	Transport Format Combination Indicator	传输格式组合指示
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
UMTS	Universal Mobile Telecommunications System	全球移动通信系统

URA	UTRAN Registration Area	UTRAN 登记区
UTRA	UMTS Terrestrial Radio Access	UMTS 陆地无线接入
UTRAN	UMTS Terrestrial Radio Access Network	UMTS 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

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 3 种 RLC 实体构成：透明模式（TM）、非证实模式（UM）和证实模式（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 节）。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）和非证实模式（UM）业务有一个发送和一个接收 RLC 实体。对于证实模式（AM）业务有一个发送和接收合并的实体。

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。

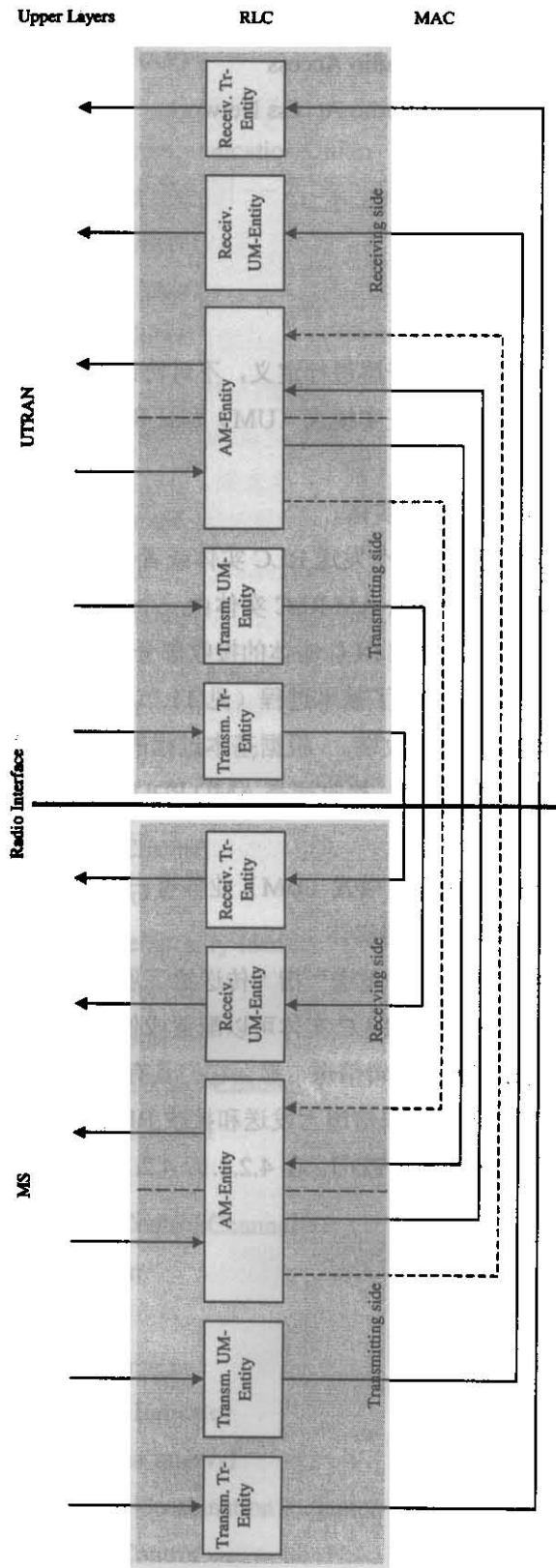


图1 RLC子层总体模型

4.2.1.1 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。

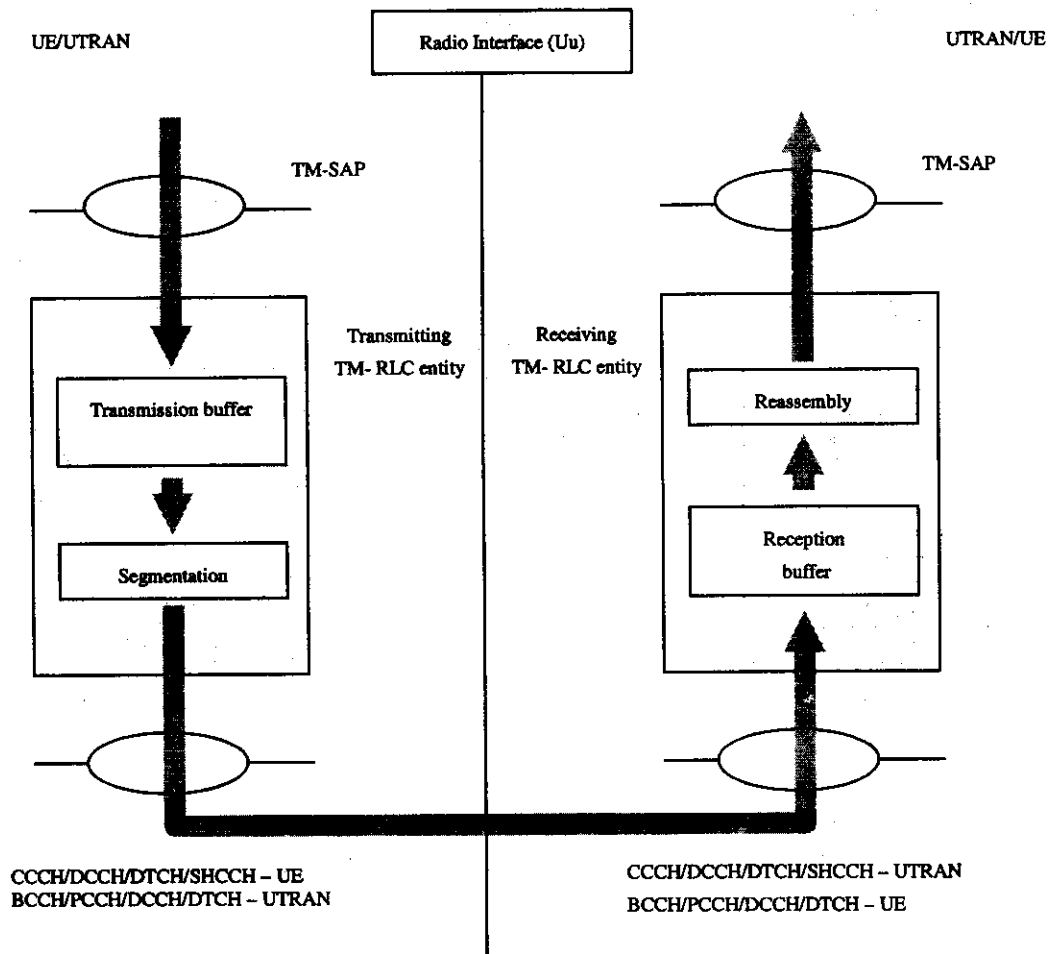


图2 两个透明模式对等实体的模型

4.2.1.1.1 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 的长度必须是一个有效 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。

4.2.1.1.2 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。

4.2.1.2 非证实模式 (UM) RLC 实体

图 3 显示了两个非证实模式对等 RLC 实体的模型。

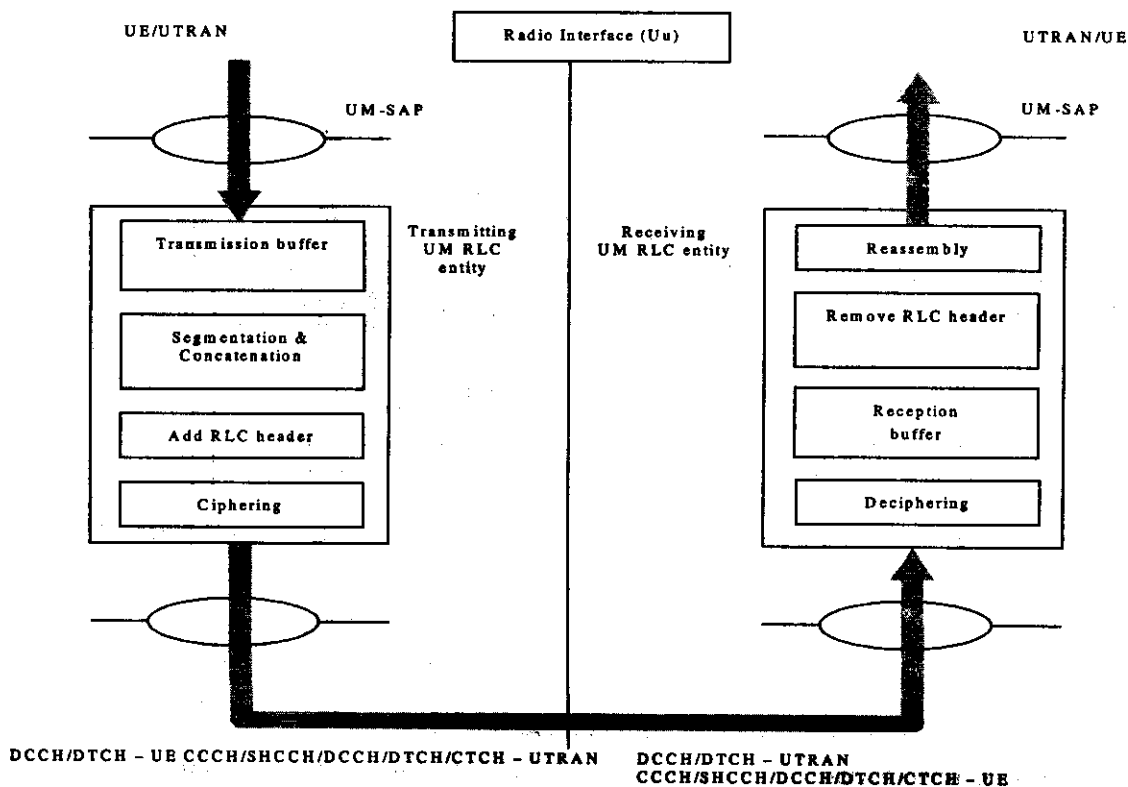


图 3 两个非证实模式对等实体的模型

4.2.1.2.1 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。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。

4.2.1.2.2 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。

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)。

接收 UM RLC 实体通过 UM-SAP 向上层传送 RLC SDU。

4.2.1.3 证实模式 (AM) RLC 实体

图 4 显示了证实模式 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 4 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。AMD PDU 和控制 PDU 应该具有相同的 RLC PDU 大小。

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，在第二个上发送控制 PDU。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。

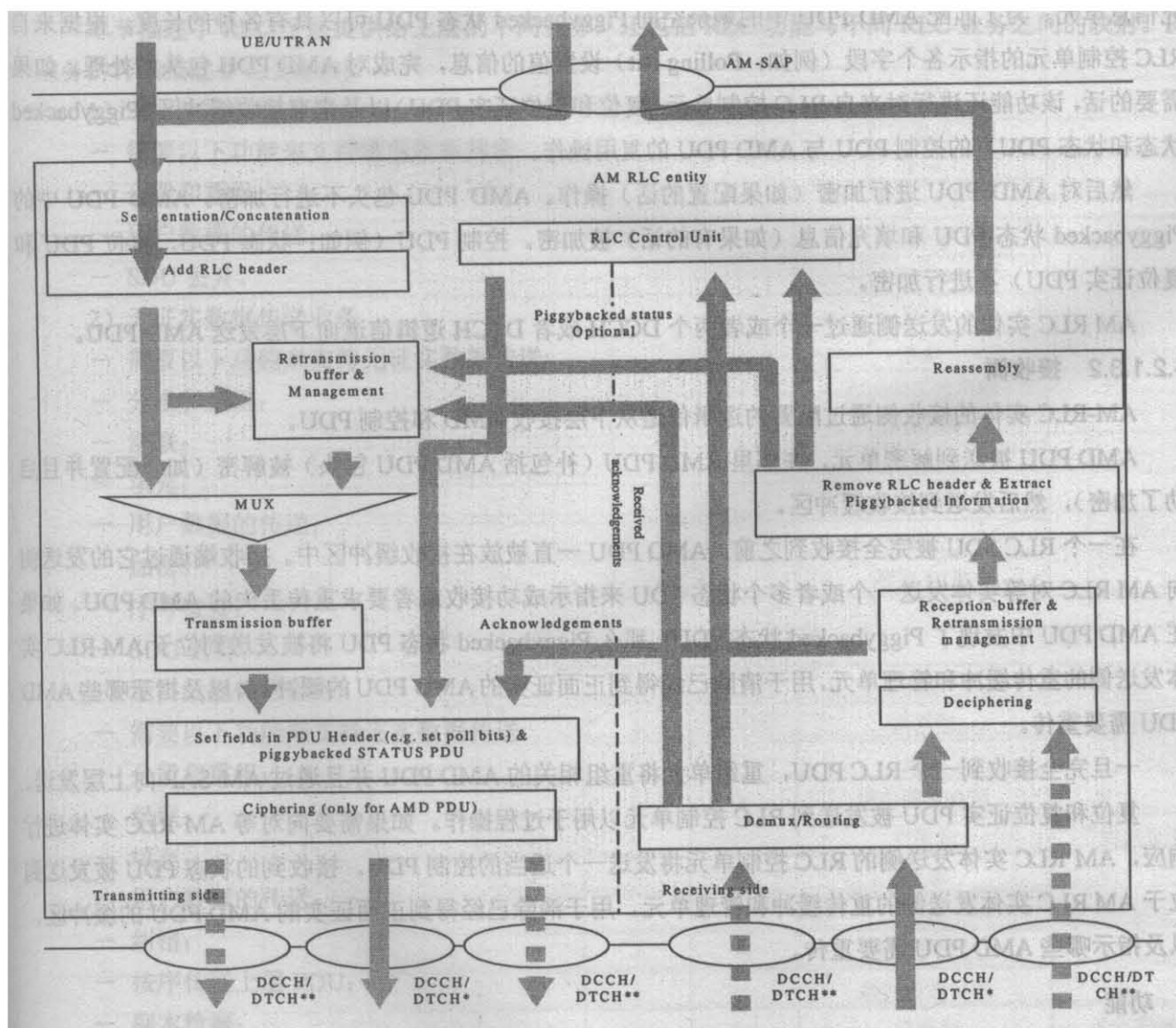


图 4 证实模式实体的模型

4.2.1.3.1 发送侧

AM-RLC 实体的发送侧通过 AM-SAP 从上层接收 RLC SDU。

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。AMD PDU 的长度是半静态的参数，它是由上层进行配置的，并且只能通过上层重建 AM RLC 实体来进行修改。AMD PDU 可能包含分段的和/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。使用长度标识来定义 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。

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 的正面或者负面的证实信息。

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（复位和复位证实 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 和复位证实 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。

4.2.1.3.2 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。

AMD PDU 被送到解密单元，在那里 AMD PDU（补包括 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证实的 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 PDU，重组单元将重组相关的 AMD PDU 并且通过 AM-SAP 向上层发送。

复位和复位证实 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证实的 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文献[3]：

- 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；
- 副本检测；

- 流量控制;
- 序号检查;
- 协议错误检测和恢复;
- 加密;
- SDU 丢弃。

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文献[3]。

1) 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。

2) 无证实数据传送业务

- 需要以下功能来支持无证实数据传送:
- 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。

3) 证实数据传送业务

- 需要以下功能来支持证实数据传送:
- 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。

4) 根据上层的定义进行 QoS 的维护

5) 不可恢复错误的通知

6.1 业务/功能在逻辑信道上的映射

下表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	-	-	-	+
非证实业务	适用性	-	-	+	+
	分段	-	-	+	+
	级联	-	-	+	+
	填充	-	-	+	+
	用户数据的传送	-	-	+	+
	加密	-	-	+	+
	SDU 丢弃	-	-	+	+
证实业务	适用性	-	-	+	+
	分段	-	-	+	+
	级联	-	-	+	+
	填充	-	-	+	+
	用户数据的传送	-	-	+	+
	流量控制	-	-	+	+
	纠错	-	-	+	+
	协议错误纠正和恢复	-	-	+	+
	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
透明业务	适用性	+	+	-	-	-	+	-
	重组	-	-	-	-	-	+	-
	用户数据传输	+	+	-	-	-	+	-
无证实业务	适用性	-	-	+	+	+	+	+
	重组	-	-	+	+	+	+	+
	解密	-	-	-	-	+	+	-
	序列号检查	-	-	+	+	+	+	+
	用户数据传输	-	-	+	+	+	+	+
证实业务	适用性	-	-	-	-	+	+	-
	重组	-	-	-	-	+	+	-
	纠错	-	-	-	-	+	+	-
	流量控制	-	-	-	-	+	+	-
	按序发送	-	-	-	-	+	+	-
	副本检测	-	-	-	-	+	+	-
	协议错误纠正和恢复	-	-	-	-	+	+	-
	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
透明业务	适用性	+	+	-	-	-	+	-
	分段	-	-	-	-	-	+	-
	用户数据的传送	+	+	-	-	-	+	-
	SDU 丢弃	-	-	-	-	-	+	-
无证实业务	适用性	-	-	+	+	+	+	+
	分段	-	-	+	+	+	+	+
	级联	-	-	+	+	+	+	+
	填充	-	-	+	+	+	+	+
	加密	-	-	-	-	+	+	-
	用户数据传输	-	-	+	+	+	+	+
	SDU 丢弃	-	-	-	-	+	+	-
证实业务	适用性	-	-	-	-	+	+	-
	分段	-	-	-	-	+	+	-
	级联	-	-	-	-	+	+	-
	填充	-	-	-	-	+	+	-
	用户数据传送	-	-	-	-	+	+	-
	流量控制	-	-	-	-	+	+	-
	纠错	-	-	-	-	+	+	-
	协议错误纠正和恢复	-	-	-	-	+	+	-
	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	+	+	-	+
无证实业务	适用性	-	-	+	+
	重组	-	-	+	+
	解密	-	-	+	+
	序列号检查	-	-	+	+
	用户数据传输	-	-	+	+
证实业务	适用性	-	-	+	+
	重组	-	-	+	+
	纠错	-	-	+	+
	流量控制	-	-	+	+
	按序发送	-	-	+	+
	副本检测	-	-	+	+
	协议错误纠正和恢复	-	-	+	+
	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文献[3]。

- 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。

表 5 RLC 和上层之间的原语

名称	参数			
	Req.	Ind.	Resp.	Conf.
RLC-AM-DATA	Data, CNF, MUI, UE-ID type indicator	Data, DiscardInfo	没有定义	MUI
RLC-UM-DATA	Data, Use special LI, UE-ID type indicator	Data	没有定义	没有定义
RLC-TM-DATA	Data, UE-ID type indicator	Data, Error_Indicator	没有定义	没有定义
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)	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，在证实模式下，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证实模式下发送的 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。

RLC-UM-DATA-Req/Ind

- RLC-UM-DATA-Req，在非证实模式下，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证实模式下发送的 RLC SDU。

RLC-TR-DATA-Req/Ind

- RLC-TR-DATA-Req，在透明模式下，上层使用该原语请求传输 RLC SDU。
- RLC-TR-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。

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“数据”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, “数据”参数的长度是 8 比特的整数倍; 否则 (TM RLC 实体) 数据参数的长度是一个比特流, 它的长度不一定是 8 比特的整数倍。

2) 参数“证实请求(CNF)”指示了 AM RLC 实体的发送侧是否需要证实对等 RLC AM 实体 RLC SDU 的接收。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 的正面证实信息, 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 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: 不可恢复的错误, 如数据链路层丢失或者可恢复状态事件, 例如复位。

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) “AM_参数”仅适用于 AM 操作。它包括 PDU 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)、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”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）和最大 UMD PDU 尺寸（见 9.2.2.8 节）。

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 节和 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“错误标识”指示错误的 RLC SDU（见 11.1.3 节）。

17) 参数“UE-ID 类型标识”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“Use special LI”指示使用在 RLC PDU 起始处指示 RLC SDU 开始的 LI。

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

1) TMD PDU (透明模式数据 PDU)

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

2) UMD PDU (非证实模式数据 PDU)

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

3) AMD PDU (证实模式数据 PDU)

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

9.1.2 控制 PDU

控制 PDU 仅用于证实模式。

1) 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；
- 接收端向发送端证实收到移动接收窗口的请求。

2) 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，依次来同步两个对

等实体。它是由发送端向接收端发送的。

3) 复位证实 (RESETACK) PDU

RESETACK PDU 是对复位 PDU 的一个证实。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
无证实	UMD	顺序非证实模式数据
证实	AMD	顺序证实模式数据
	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 丢弃证实
	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 丢弃证实
	RESET	复位命令
	RESETACK	复位证实

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 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。

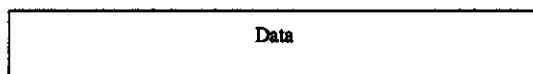


图 5 TMD PDU

9.2.1.3 UMD PDU

当 RLC 工作在非证实模式下时，使用 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的字节组成。

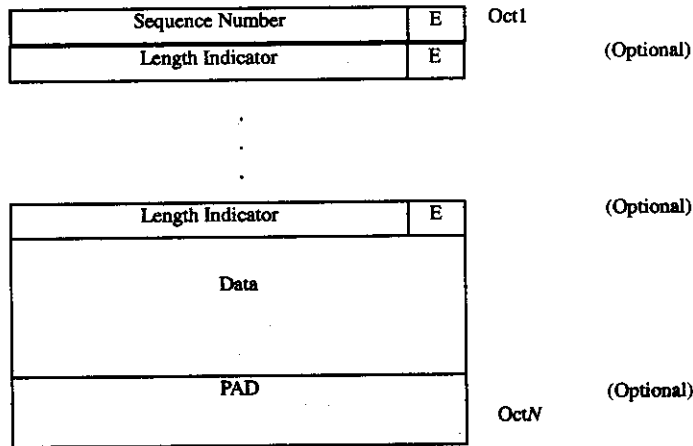
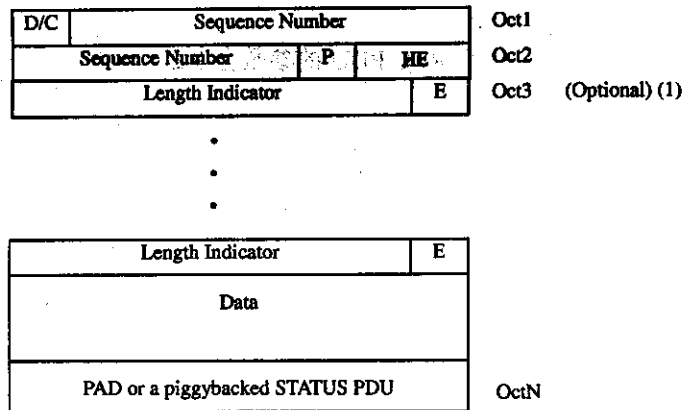


图 6 UMD PDU

注：“长度标识”可能是 15bit

9.2.1.4 AMD PDU

当 RLC 操作在证实模式时，使用 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，这两个字节中包含序号。RLC 包头由前两个字节和所有包含长度标识的字节组成。



注：“长度标识”可能是 15bit.

图 7 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 的格式见图 8。每个 super field (SUFI) 的长度取决于它的类型和内容。

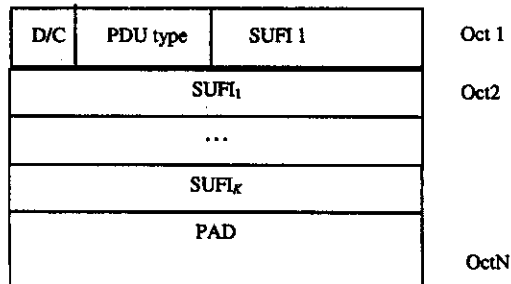


图 8 状态 PDU (STATUS PDU)

状态 PDU 可以包含不同类型的 super field。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，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。

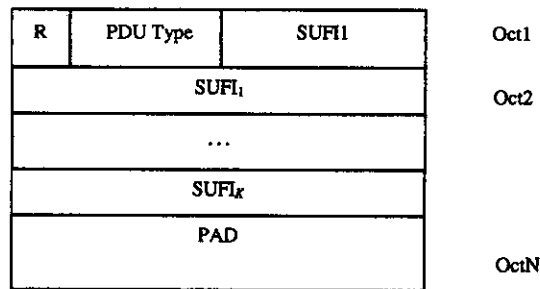


图 9 Piggybacked 状态 PDU

9.2.1.7 复位、复位证实 PDU

RESET PDU 中包含一个 1bit 的序号字段 (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。

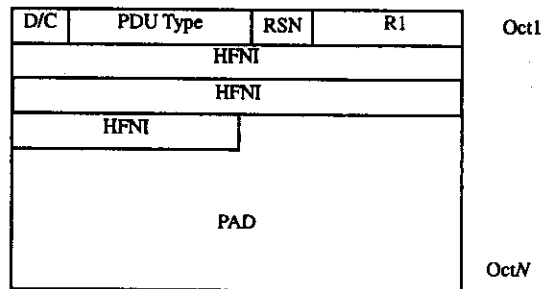


图 10 RESET, RESET ACK PDU

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。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 的长度应该是整数字节。

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	复位证实(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 (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 bit	用于重传和重组
UMD PDU	7 bit	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

该比特指示了下一个八位组是否是一个“长度标识”和 E 比特。

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

9.2.2.6 保留 1(R1)

长度: 3 bit

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 bit

该两比特字段指示了下一个字节是数据还是“长度标识”和 E 比特。

Value	Description
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
10-11	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)

“长度标识”用于指示在 PDU 内终结的每一个 RLC SDU 的最后一个字节。

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。

“长度标识”的大小可能是 7bit 或者 15bit。“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“长度标识”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：

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

对于 UM：

- 如果“最大 UMD PDU 大小” ≤ 125 Byte：
- 应该使用 7bit 的“长度标识”。
- 否则
- 应该使用 15bit 的“长度标识”。
- 在“最大 UMD PDU 大小”的各个修改之间，“长度标识”的尺寸大小对于所有 UMD PDU 相同。
- 如果在下行链路上配置了参数“Use special LI”；并且
- 如果 RLC SDU 在 RLC PDU 开始处开始；并且
- 如果没有“长度标识”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 15bit “长度标识”的情况下）填满上一个 RLC PDU。
- 如果使用 7bit “长度标识”
- 应该使用“111 1100”作为“长度标识”的值；
- 如果使用 15bit “长度标识”
- 应该使用“111 1111 1111 1100”作为“长度标识”的值
- 在上行链路：
- 不使用值为“111 1100”或者“111 1111 1111 1100”的“长度标识”。

在 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“长度标识”

的情况下：

- 如果使用 7bit 的“长度标识”
- 在下一个 PDU 中应该使用值为“000 0000”的“长度标识”作为第一个“长度标识”；
- 如果使用 15bit 的“长度标识”
- 在下一个 PDU 中应该使用值为“000 0000 0000 0000”的“长度标识”作为第一个“长度标识”。

在 PDU 包含一个指示 RLC SDU 差一个字节填满 PDU 的 15bit “长度标识”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“长度标识”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。

在 PDU 中使用 15bit “长度标识”并且 RLC SDU 的最后一个分段正好差一个字节填满 PDU。

- 如果在下列 PDU 中使用 15bit “长度标识”：
- 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“长度标识”。
- 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“长度标识”指示存在填充。

— 如果在下列 PDU 中使用 7bit “长度标识”：

- 如果 RLC 被配置为 UM 模式：
- 在下一个 PDU 中应该使用值为“000 0000”的“长度标识”作为第一个“长度标识”，并且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 “长度标识”，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“长度标识”作为 PDU 的最后一个“长度标识”来指示使用了填充。
- 如果在 RLC PDU 中使用 15bit “长度标识”，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“长度标识”作为 PDU 的最后一个“长度标识”来指示使用了填充。

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

如果一个“长度标识”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“长度标识”、适当的填充“长度标识”以及填充组成的 RLC PDU。

使用预先定义的“长度标识”值来指示填充。在下面的图标中，按照“长度标识”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“长度标识”才能够指示填充空间。这些值应该放在 PDU 的其他所有“长度标识”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“长度标识”来指示 piggybacked 状态 PDU 的存在。这个“长度标识”替换了填充“长度标识”。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“长度标识”来指示在 AMD PDU 中仍然包含填充。

如果配置了“通过显式的信令丢弃 SDU”：

- 一个 AMD PDU 最多可以包含 15 个“长度标识”，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	Description
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“长度标识”
1111100	UMD PDU: 这个 RLC PDU 中的第一个字节是一个 RLC SDU 的第一个字节。 AMD PDU: 保留（在这个协议版本中，这种编码的 PDU 将被丢弃）
1111101	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	Description
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“长度标识”
11111111111011	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“长度标识”。上一个 RLC PDU 中剩余的一个字节被忽略
11111111111100	UMD PDU: 这个 RLC PDU 的第一个字节是一个 RLC SDU 的第一个字节。AMD PDU: 保留（在这个协议版本中，这种编码的 PDU 将被丢弃）
11111111111101	保留（在这个协议版本中，这种编码的 PDU 将被丢弃）
11111111111110	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）
11111111111111	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、非证实和证实模式下，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。

非证实模式数据和证实模式数据：

- RLC SDU 的长度为 8bit 的整数倍。

- 为了完全填满数据字段以及避免不必要的填充，一个 RLC SDU 的最后一个分段应该与下一个 RLCSDU 的第一个分段相连接。使用“长度标识”来指示 RLC SDU 间的边界。（见 9.2.2.8 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)的 AMD PDU 的信息，也就是说还没有到达接收端的 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、证实等）、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 11 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。

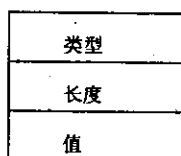


图 11 SUFI 的结构

类型字段的长度为 4bit，它可以是下列值中的任意一个。

bit	描述
0000	不再有数据(NO_MORE)
0001	窗口大小(WINDOW)
0010	证实(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	移动接收窗口证实(MRW_ACK)
1000-1111	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 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，见下图 12。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。

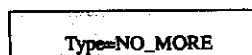


图 12 状态 PDU 中的 NO_MORE 字段

9.2.2.11.2 证实 SUFI

“证实” SUFI 由一个类型标识字段 (ACK) 和一个序列号 (LSN) 组成，如下图 13 所示。证实 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。

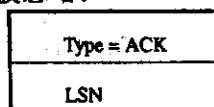


图 13 状态 PDU 中的 ACK 字段

LSN

长度: 12 bit

证实“序列号” < 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 应该被包含在相同的 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 SUFI

窗口尺寸 SUFI 由一个类型标识 (窗口) 和一个窗口尺寸 (WSN) 组成，如下图 14 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。

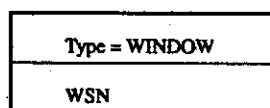


图 14 状态 PDU 中的窗口字段

WSN

长度: 12 bit

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，SUFI 将被丢弃。一旦接收到这个 SUFI，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 SUFI

列表 SUFI 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, 如下图 15 所示。

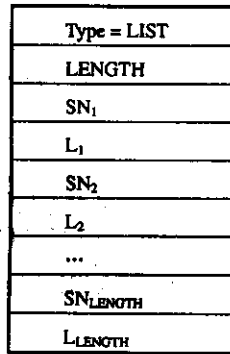


图 15 状态 PDU 中的列表字段

长度

长度: 4 bit

LIST 类型的 SUFI 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 bit

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 SUFI 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 16 所示。

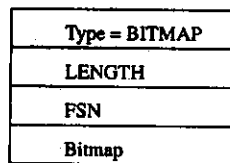


图 16 状态 PDU 中的 Bitmap 字段

LENGTH

长度: 4 bit

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 bit

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 “序列号”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 “序列号”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 ($bit_position \in [0, LENGTH*8 - 1]$):

1: $SN = (FSN + bit_position)$ 已经正确接收

0: $SN = (FSN + bit_position)$ 没有正确接收

9.2.2.11.6 相关列表 SUFI

相关列表 SUFI 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 17 所示。

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

图 17 状态 PDU 中的相关列表字段

LENGTH

长度: 4 bit

RLIST 类型的 SUFI 中的码字 (CW) 数。

FSN

长度: 12 bit

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

CW

长度: 4 bits

CW 由 4 比特组成, 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
$X_1 X_2 X_3 0$	数的下三个比特是 $X_1 X_2 X_3$ 并且数在下一个 CW 中继续。在这个 CW 中, 最高有效位是 X_1
$X_1 X_2 X_3 1$	数的下三个比特是 $X_1 X_2 X_3$ 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 到并且包括下一个错误 AMD PDU 的间隔。

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 移动接收窗口证实 SUFI

“移动接收窗口证实” SUFI 用来证实 MRW_SUFI 的接收。它的格式如图 18 所示。

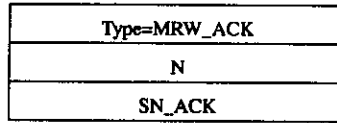


图 18 状态 PDU 中的 MRW_ACK 字段

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 SUFI 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_SUFI 有响应。

SN_ACK

长度：12bit

SN_ACK 字段用来指示在接收到 MRW SUFI 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_SUFI 有响应。

9.2.2.11.8 移动接收窗口（MRW）SUFI

“移动接收窗口” SUFI 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。格式见图 19。

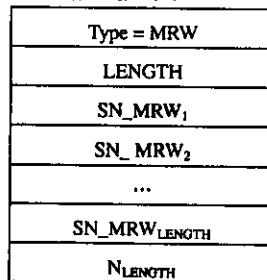


图 19 状态 PDU 中的 MRW 字段

LENGTH

长度：4 bit

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：12bit

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“序列号”，该 AMD PDU 包含在接收端被丢弃的第 i 个 RLC SDU 的“长度标识”（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“序列号” $< SN_MRW_{LENGTH}$ 的 AMD PDU，并相应的移动接收窗口。另外，当 $N_LENGTH>0$ 时，接收端必须丢弃第一个 N_LENGTH “长度标识”以及“序列号” SN_MRW_{LENGTH} 的 AMD PDU 中相应的数据字节。

N_LENGTH

长度：4 bit

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“长度标识”与在接收端被丢弃的最后一个 RLC SDU 相对应。 $N_LENGTH=0$ 表示最后一个 RLC SDU 在“序列号” $SN_MRW_{LENGTH}-1$ 的 AMD PDU 中结束，并且“序列号” SN_MRW_{LENGTH} 的 AMD PDU 中的第一个数据字节是将要被重传的第一个数据字节。

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 8 比特的整数倍，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。

9.2 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

图 20 说明了透明模式 RLC 实体（传送和接收）的状态模型。透明模式实体可以处于下列状态之一。

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;
- 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;
- 认为是已被终结。

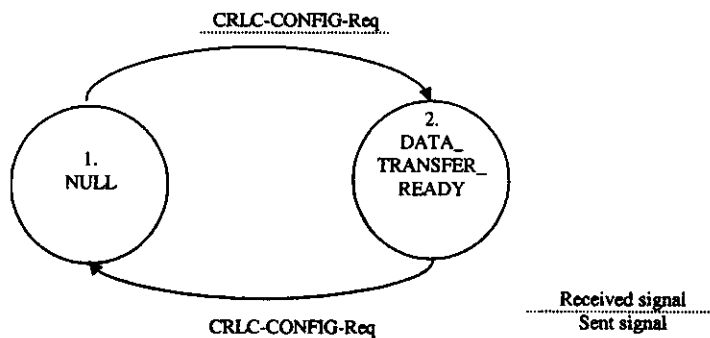


图 20 透明模式实体的状态模型

9.3.2 非证实模式实体的状态模型

图 21 描述了非证实模式 RLC 实体 (发送和接收) 的状态模型。一个非证实模式实体可以处于以下状态之一。

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;
- 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证实模式数据的交换。

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

- 进入 NULL 状态;
- 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“序列号”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-RESUME-Req, RLC 实体:

- 进入 DATA_TRANSFER_READY 状态;
- 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。

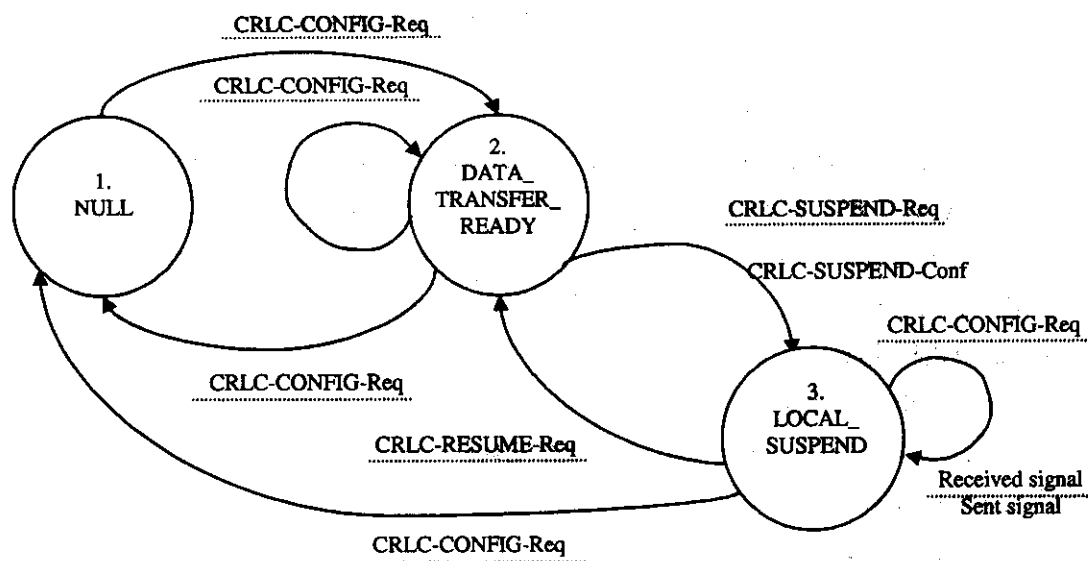


图 21 非证实模式实体的状态模型

9.3.3 证实模式实体的状态模型

图 22 描述了证实模式 RLC 实体（发送和接收）的状态模型。一个证实模式实体可以处于以下状态之一。

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;
- 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 节在实体之间进行证实模式数据的交换。

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

- 进入 NULL 状态;
- 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);
- 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；
- 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；
- 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)；
- 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；
- 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。

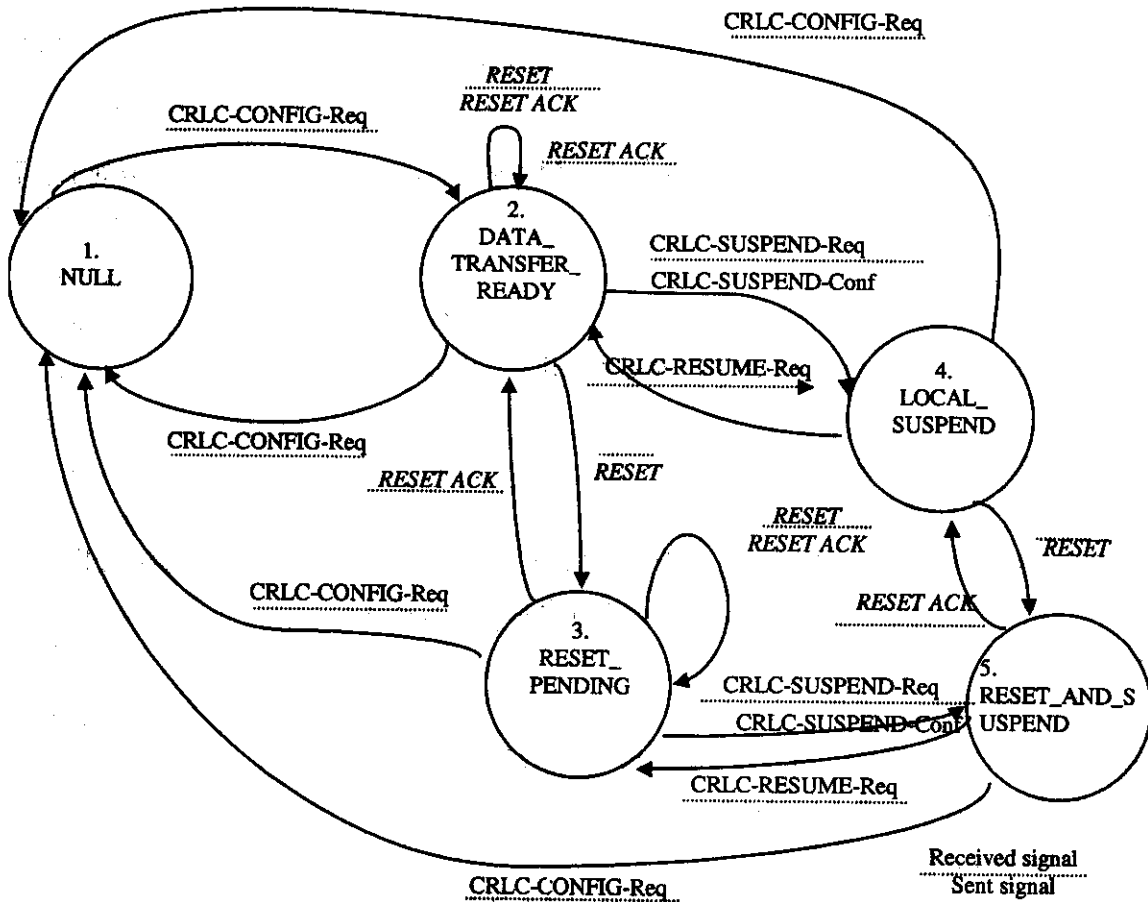


图 22 证实模式实体的状态模型

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“序列号”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;
- 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;
- 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);
- 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;
- 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;
- 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;
- 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) 和 VR(US) 相关的算术运算都受到 UM 模数的影响。在执行一个模数基应该使用的状态变量或者序列号值的算术比较时, 将从所有相关的值中减去这个模数基 (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) 和 VR(US) 分别作为 AM 和 UM 的模数基。

在接收端, RLC 维护下列状态变量:

1) 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“序列号”, 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。

2) VT(A) – 证实状态变量

这个状态变量包含了最后一个被按顺序证实的 AMD PDU 的“序列号”后的“序列号”，其形成了可接收证实的发送窗口的下届。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证实的 AMD PDU 后的第一个“序列号”。

3) 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，相应的变量值就加增加。

该变量的初始值为 0。

4) 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“序列号”， $VT(MS) = VT(A) + VT(WS)$ 。该值表示发送窗口的上界。除非 $VT(S) \geq VT(MS)$ ，发送端不发送“序列号” $\geq VT(MS)$ 的 AMD PDU。在那种情况下，“序列号”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。

5) 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。

6) 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。

7) 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 的所有 AMD PDU 被至少发送过一次后，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 SDU 最后一个分段的 AMD PDU 第一次发送时，“Polling bit”被设为 1。

该变量的初始值为 0。

8) 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。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。当对等 RLC 实体初始化 RLC 复位时，VT(RST) 不进行复位。

该变量的初始值为 0。

9) VT(MRW) – MRW 命令发送状态变量

这个状态变量用来计数 MRW 命令被安排发送的次数。每当 MRW SUFI 被安排发送，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。

10) 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。

这个变量得初始值是 Configured_Tx_Window_size。

在接收端，RLC 维护以下状态变量：

11) 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“序列号”。当接收到具有与 VR(R) 相等的“序列号”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“序列号”。

12) VR(H) – 最高期望状态变量

这个状态变量包含接收到的最高 AMD PDU “序列号”后的序列号。当接收到“序列号” x 满足 $VR(H) \leq x < VR(MR)$ 的 AMD PDU 时，这个状态变量应该被设为 $x+1$ 。

13) 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“序列号”， $VR(MR) = VR(R) + \text{Configured_Rx_Window_Size}$ 。

14) VR(US) – 接收者发送顺序状态变量

这个状态变量包含接收到的最后一个 UMD PDU 后的“序列号”。当接收到“序列号”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。

15) VR(EP) – 估计的 PDU 计数器状态变量

这个状态变量包含了作为最近传输的状态报告的结果，仍然希望被重传的 AMD PDU 的数目。在每个 TTI 的结尾，这个变量根据在这段时间内接收到的 AMD PDU 的总数进行减少。

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。

1) Timer_Poll

— 只有上层进行了相关配置时，才使用该定时器。定时器的值由上乘信令指示。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，当一个包含轮询信息的 AMD PDU 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“序列号”小于或者等于 $x-1$ 的 AMD PDU 的正面证实；
- 对于“序列号” = $x-1$ 的 AMD PDU 的反面证实。

如果定时器超时，并且没有收到满足上述条件的 STATUS PDU，则：

- 应该再一次向接收端发起轮询；
- 定时器应该重新启动；
- 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。

2) 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，当一个包含轮询信息的 AMD PDU 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。

3) Timer_EPC

只有在上层配置了 EPC 功能时，才使用该定时器，该定时器用来计算回环延迟，即从一个状态 PDU 的发送到接收到第一个重传 AMD PDU 之间的时间。该定时器的初始值由上层信令指示。

在 UE 端，当下层指示状态报告的第一个 STATUS PDU 被成功或者不成功发送时，该定时器启动。在 UTRAN 端，当状态报告的第一个 STATUS PDU 被发送给下层时，该定时器启动。仅当 Timer_EPC 超时后，才能像 9.7.4 节中描述的那样减少 VR(EP) 的值。

4) 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 被证实之前定时器超时，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。

5) 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（还没有被证实），那么：
- 触发一个轮询。

6) 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证实状态报告。如果状态报告包括 SUFIs、LIST、BITMAP、RLIST 或 ACK 中的任何一个的话，那么它就是一个证实状态报告。定时器的值由上层信令指示。

在 UE 端，当下层指示一个证实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，当一个证实状态报告的最后一个 STATUS PDU 被发送给下层时，该定时器启动。

从证实状态报告被触发开始，直到 Timer_Status_Prohibit timer 超时，证实被禁止。如果在证实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，证实不被禁止。

7) 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。

8) 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。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。

9) 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。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。

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。

1) 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。

2) 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。

3) 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。

4) 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。

5) 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。

6) 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 $VT(W)$ 状态变量的值。

7) Configured_Rx_Window_Size.

这个协议参数指示了接收窗口尺寸。

8) 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。

9.7 指定功能

本节中所定义的功能是标准化的。

9.7.1 证实模式的轮询功能

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”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：

1) 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）；

- 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。

2) 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；

- 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。

3) 轮询定时器

定时器 $Timer_Poll$ 根据 9.5 节 1) 启动和停止。当定时器 $Timer_Poll$ 超时，发送端触发轮询功能。

4) 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。

5) Every Poll_SDU SDU

发送端对于每个 Poll_SDU SDU 触发轮询功能。对于包括 RLC SDU 分段的最后一个 AMD PDU 的第一次发送，触发轮询。

6) 基于窗口

当 9.6 节 4) (“Poll_Window”) 中描述的条件满足时，发送端触发轮询功能。

7) 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 2)), 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 2));
- 如果有一个或者多个 AMD PDU 将要发送, 或者存在未被接收端证实的 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 已经被证实):

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;
- 如果有一个或者多个 AMD PDU 将要发送, 或者存在未被接收端证实的 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 已经被证实):

不初始化轮询功能。

9.7.2 证实模式的状态传送

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:

1) 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。

2) 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 7)。当上层配置了“周期性状态阻塞”时, 触发器不应该被激活。

3) EPC 机制

根据 9.5 节 3) 启动定时器 `Timer_EPC`, 根据 9.7.4 节设置并且减少状态变量 `VR(EP)`。如果在变量 `VR(EP)` 等于 0 以前没有收到所有请求重传的 AMD PDU, 那么接收端触发一个新的状态报告。9.7.4 节中对 EPC 机制进行了更加详细的描述。

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 SUFI 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:

1) STATUS 禁止

根据 9.5 节 6) 启动定时器 `Timer_Status_Prohibit`。在证实被禁止的情况下, 不允许接收端发送状态报告 (见 9.5 节 6))。如果在此期间触发了一个状态报告, 那么当定时器 `Timer_Status_Prohibit` 超时的时候发送该状态报告, 具体描述如下。

2) EPC 机制

如果激活 EPC 机制并且触发了一个状态报告的发送, 那么发送将被推迟到“EPC 机制”结束时进行,

具体描述如下。

当触发一个状态报告时，接收端应该：

- 如果功能“STATUS 禁止”或者“EPC 机制”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”或者“EPC 机制”之一所禁止）：
- 如果在状态报告中要求 MRW, MRW_ACK 或者 WINDOW 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 超时或者“EPC 机制”终结的时候，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”或者“EPC 机制”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 证实模式、非证实模式和透明模式的 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。

下面是 RLC SDU 丢弃功能的一系列操作模式（见表 7），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 删除的条件的列表

操作模式	Presence
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

该可供选择的方法仅适用于工作在证实模式的 RLC 实体。它使用基于定时器(Timer_Discard)的 SDU 丢弃触发机制。这使得 SDU 丢弃功能对信道速率的变化不敏感，并且提供了准确定义最大延迟的方法。然而，连接的 SDU 丢失率由于 SDU 被丢弃而提高。

对于每个从高层接收到的 SDU，发送端应该：

- 启动定时器 Timer_Discard。

当 SDU 的定时器 Timer_Discard 超时，发送端应该：

- 丢弃该 SDU；
- 如果配置了“Send MRW”，或者被丢弃的 SDU 的一个或者多个分段已经被发送给下层，那么：
- 根据 11.6 节，使用明确的信号通知接收端。

注：对于配置“Send MRW”和与此配置相连的功能的支持在实现上是相关联的。

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证实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（对于工作在非证实模式的 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证实模式 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 SDU 分段；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，因此仅适用于证实模式 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

如果非证实 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 丢失，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。

对于证实模式 RLC 实体，总是配置 SDU 丢弃模式。

9.7.4 证实模式的估计的 PDU 计数器

估计的 PDU 计数器（EPC）仅适用于工作在证实模式的 RLC 实体。EPC 是一种由上层配置的，用于在接收端安排状态报告重传的机制。利用该机制，接收端发送一个新的状态报告，在该状态报告中请求还没有接收到的 AMD PDU。两个连续的状态报告重传之间的时间是不固定的，但是它是由定时器 Timer_EPC 和状态变量 VR(EP)控制的，它们调整这个时间以适应回环时延和在 TFI 中指示的当前比特速率，其目的是使状态报告重传的延迟最小。

当一个状态报告被一些机制触发并且提交给下层（在 UTRAN），或者下层指示了一个或者多个 AMD PDU 被成功或不成功发送（在 UE）来请求重传丢失的 AMD PDU 时，变量 VR(EP)被设为等于要求的 AMD PDU 的数目。至少需要一个请求的 AMD PDU 来激活 EPC 机制。变量 VR(EP)是一个计数器，在每一个 TTI 间隔，它被减去在这个 TTI 期间在相应的逻辑信道上应该接收到的 AMD PDU 的估计数目。

定时器 Timer_EPC 控制在它开始倒数之前，变量 VR(EP)需要等待的最大时间。当接收端发送重传

请求后（当状态报告的第一个 STATUS PDU 被传递给下层（在 UTRAN）或者下层指示了它成功或不成功的发送（在 UE）），该定时器立刻启动。定时器 Timer_EPC 的初始值由上层配置。典型的，它将基于回环时延，这包括传输时延和在发送端、接收端以及帧结构的处理时间。这个定时器也可以被实现为计数器，它对 10ms 无线帧的数目进行计数，用来统计在收到第一个请求的 AMD PDU 之前，期望的已经过的 10ms 无线帧的数目。

如果当变量 VR(EP)等于 0，并且不是所有这些请求的 AMD PDU 都已被正确接收，那么将发送一个新的状态报告，并且相应的重新启动 EPC 机制。当状态报告的第一个 STATUS PDU 被提交给下层（在 UTRAN）或者下层指示它被成功或未成功传送（在 UE）时，定时器 Timer_EPC 会再一次启动，如果所有请求的 PDU 都被正确的接收，那么 EPC 机制停止。

9.7.5 证实模式和非证实模式传输的本地挂起功能

高层可能将 RLC 实体挂起。

当工作在非证实模式的 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)当前值的证实信息来对挂起请求进行证实。
- 不发送“序列号” $SN \geq VT(US) + N$ 的 UMD PDU。

当工作在证实模式的 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)当前值的证实信息来对挂起请求进行证实。
- 不发送“序列号” $SN \geq VT(US) + N$ 的 AMD PDU。

当工作在非证实模式的 RLC 实体被上层回复时，RLC 实体应该：

- 恢复数据传输过程。

当工作在证实模式的 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 证实和非证实模式的 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 证实和非证实模式的 RLC 重建功能

上层可能重建 RLC 实体。

RLC 重建功能适用于 AM 和 UM，在上层要求重建 RLC 实体时使用。

当上层重建 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 UL 和 DL 的超帧号 (HFN)。
- 如果 RLC 实体工作在非证实模式，那么：
- 如果它是一个接收 UM RLC 实体，那么：
- 丢弃所有 UMD PDU。
- 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 RLC SDU 没有丢弃，那么不停止 Timer_Discard。
- 否则，如果 RLC 实体工作在证实模式：
- 丢弃 RLC 实体发送侧和接收侧的所有 AMD PDU 和控制 PDU。
- 停止 9.5 节中描述的所有定时器，除了 Timer_Poll_Periodic 和 Timer_Status_Periodic。

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 证实和非证实模式的加密

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 23 所示。

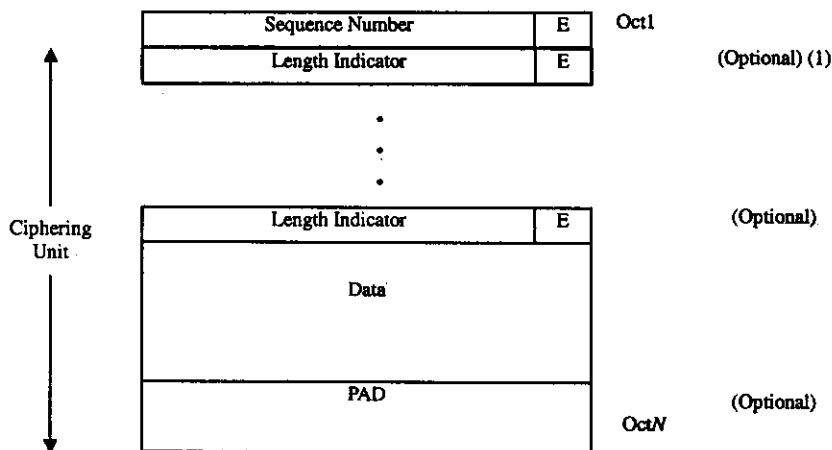


图 23 UMD PDU 的加密单元

- 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。如下图 24 所示。

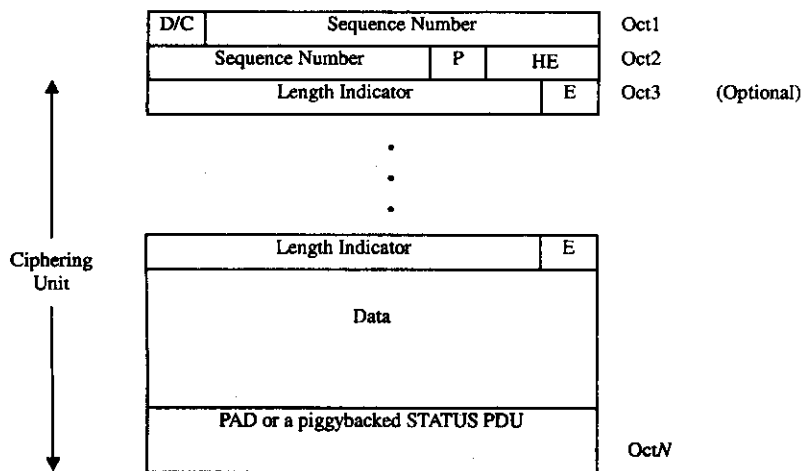


图 24 AMD PDU 的加密单元

由上层配置加密算法和加密键文献[8]并且如文献[9]中定义的使用加密方法。

文献[9]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 (定义为文献[9]中所描述的无线承载标识。它将使用文献[8]中所描述的 RB 标识-1)；
- CK (加密键)。

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，至少有一个被否定证实的 AMD PDU 的序列号不在“证实状态变量”VT(A)和“发送状态变量”VT(S)的值之间；或者
- 一个 ACK 的 LSN 不在“证实状态变量”VT(A)和“发送状态变量”VT(S)之间。

如果一个 AM RLC 实体接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。

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。图 25 说明了透明模式数据传送的基本过程。发送实体可以是 UE 或网络，接收实体可以是网络或 UE。

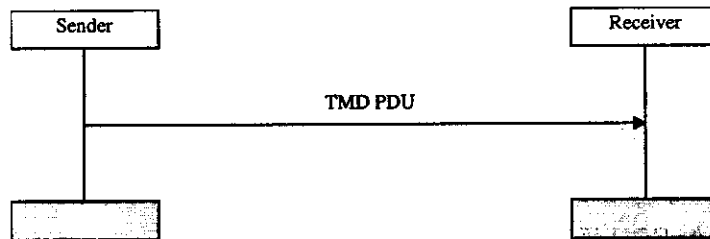


图 25 透明模式数据传送过程

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：

- 1) 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；
- 2) 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；
- 3) 根据 MAC 层请求的 TMD PDU 数量把 TMD PDU 发送给 MAC；
- 4) 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。

1) 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；

2) 如果设为“**No Detect**”，RLC 将把全部接收的 RLC_SDU 向高层传送；

3) 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

11.1.4.1 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。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 非证实模式数据 (UMD) 传送过程

11.2.1 概述

非证实模式数据传送过程是用于在非证实模式下两个 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。图 26 说明了用于非证实模式数据传送的基本过程。发送实体可以是 UE 或网络，接收实体可以是网络或 UE。

可用的信道为 DTCH、DCCH、CCCH（仅用于下行链路）、SHCCH（仅用于下行链路）。逻辑信道的类型取决于 RLC 实体是位于用户平面 (DTCH) 还是位于控制平面。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。

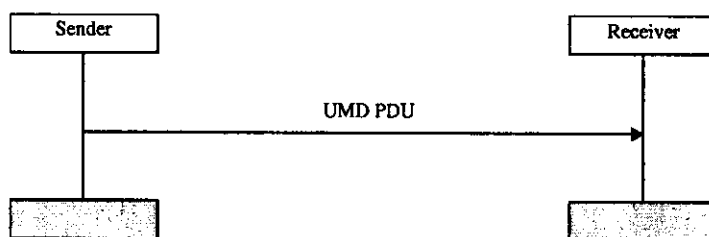


图 26 非证实模式数据传送过程

11.2.2 UMD PDU 传送

当有来自高层的非证实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：

1) 序列号 (SN) 字段将设置为 VT (US)。

2) 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。

3) 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。

11.2.2.1 UMD PDU 的传送

当要传送的 SDU 准备就绪，发送实体将

1) 通知 MAC 将要发送的 SDU 的数量和大小；

2) 根据下层指示的 PDU 的大小将 SDU 分段或连接，放入 TMD PDU 中；

3) 根据请求的数量，把相应的 UMD PDU 发送给 MAC；

4) 对于每个被传输的 UMD PDU，更新 VT (US) 状态变量的值；

5) 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)。

11.2.2.2 UMD PDU 的接收

当接收到一个 UMD PDU，接收实体将

1) 根据接收的 PDU 更新 VR (US) 状态变量 (参见 9.4)；

2) 把 PDU 重新组装成 RLC SDU；

3) 如果变量 VT(US) 的更新步长大于 1，即有一个或更多的 PDU 丢失了。在这些丢失的 PDU 中有分割片的 SDU 都将被丢弃；

4) 如果下行链路收到的 UMD PDU 的第一个长度指示 (LI) 值为 "1111 100" 或 "1111 1111 1111 100"，该 UMD PDU 中的第一个数据字节为一个 RLC SDU 的首字节；

5) RLC 通过 UM-SAP 向高层传送 RLC SDU。

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 没有显示信令丢弃 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。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 UMDPDU，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。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。

11.3 证实模式数据 (AMD) 传送过程

11.3.1 概述

证实模式数据传送过程用于证实模式下两个 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。图 27 说明了证实模式数据传送的基本过程。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。

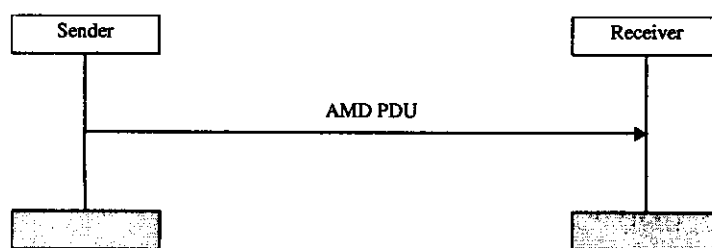


图 27 证实模式数据传送过程

11.3.2 AMD PDU 的传送

当有来自高层的证实模式数据传送或 PDU 的重传的请求，发送实体启动此过程。

发送实体根据高层配置的 PDU 的固定大小将 RLC SDU 分段放入多个 AMD PDU 中。同时为在一个 PDU 中结束的每个 SDU 设置一个长度指示字段 (LI)。

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 $Timer_Discard$ ，在接收 SDU 时启动。

只允许发送实体重传那些由接收实体指示了已丢失的 PDU。

当轮询被“Poll timer”或“Timer based”触发并被允许的条件下，如果没有 AMD PDU 被传送或重传，为了完成轮询，当 $Configured_Tx_Window_Size$ 大于等于 2048，选择序列号为 $VT(S)-1$ 的 PDU 重传；当 $Configured_Tx_Window_Size$ 小于 2048，选择序列号为 $VT(S)-1$ 的 PDU 或还没有被证实的 PDU 来重传。

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 11.3.3。

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：

- 1) 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；
- 2) 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：

- 1) 序列号字段将被设置为 VT(S) 且 VT(S) 将被更新;
- 2) 为在一个 PDU 中结束的每个 SDU 设置一个长度指示字段 (LI)。LI 字段的设置参见 9.2.2.8;
- 3) 轮询检测比特的设置在 11.3.2.1.1 节中规定。

如果是重传:

- 1) 重传 PDU 的序列号和第一次传送时相同;
- 2) 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。
- 3) 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 的序列号小于 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。

- 1) 发送实体通知 MAC 允许传送或重传的 PDU 的数量;
- 2) AMD PDU 的内容根据 11.3.2.1 设置;
- 3) 发送给 MAC 层请求数量的 AMD PDU;
- 4) 重传 PDU 的优先权高于首次传送的 PDU;
- 5) 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;
- 6) 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;

7) 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, 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 发起显式信令的 SDU 丢弃流程（SDU_discard with explicit signalling）。

11.3.5 异常情况

11.3.4.1 Timer_Poll timeout (void)

当 Timer_Poll 超时，发送实体将重新发送轮询检测。轮询检测可以在一个新的 PDU 或一个重传 PDU 中重传。

11.3.4.2 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。

11.3.4.3 定时器 Timer_Discard 超时

11.3.5.1.1 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。

11.3.4.4 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。

11.3.4.5 AMD PDU 的长度指示保留值

如果接收到的 PDU 的长度指示值在该协议版本中为保留值，接收实体将丢弃该 PDU 并且按丢失 PDU 处理。

11.4 RLC 复位过程

11.4.1 概述

RLC 复位过程用来复位在证实模式下运行的两个对等 RLC 实体。图 28 说明了 RLC 复位的基本过程。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 和序列号为 VT(S)-1 的 AMD PDU 相关，如果没有 AMD PDU 传送，序列号为 0。

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

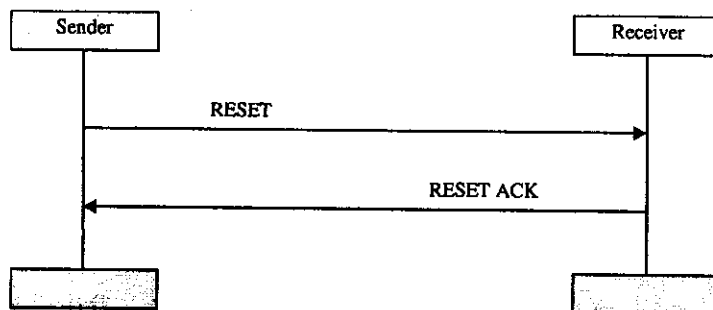


图 28 RLC 复位过程

11.4.2 复位发起

当 $VT(DAT)$ 等于 $MaxDAT$ ，并且配置了 “No_discard after MaxDAT number of transmission”，发送实体将发起 RLC 复位流程。

当发送实体收到一个含有错误序列号 (SN) 的 STATUS PDU，将停止发送任何 AMD PDU 和 STATUS PDU， $VT(RST)$ 加 1，此时

- 1) 如果 $VT(RST)=MaxMRW$ ，发送实体将传送一个 RESET PDU，并完成 11.4.4.1) 给出的流程；
- 2) 如果 $VT(RST)<MaxMRW$ ，发送实体将传送一个 RESET PDU，并启动定时器 $Timer_RST$ 。

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 需要等到下一个 TTI 结束时才启动复位过程。

发起复位过程后，只有在收到与对应 RESET PDU 相同的 RSN 值的 RESET ACK PDU 或者收到高层的重建/释放请求后才能结束此过程。复位过程不被收到来自对等实体的 RESET PDU 中断。

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 的序列号。每次 AM 实体建立或重建后的第一个 RESET PDU 的序列号为 0。每发送一个新的 RESET PDU，序列号加 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：

- 1) 用一个 RESET ACK PDU 进行响应；
- 2) RESET ACK PDU 的内容设置参见 11.4.3.1；
- 3) 同时把 9.4 节中的变量 (除 $VT(RST)$) 复位到初始值；
- 4) 停止除 $Timer_RST$ 、 $Timer_Discard$ 、 $Timer_Poll_Periodic$ 和 $Timer_State_Periodic$ 外的所有定时器；
- 5) 复位可配置参数到原来的配置值；
- 6) 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；
- 7) 将 HFN 设为接收到的 RESET PDU 中 HFNI 字段中的值；
- 8) 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。

如果 RSN 的值和先前收到的 RESET PDU 中 RSN 值相同 (即该 RESET PDU 是一个重传的 RESET PDU)，要么和收到不同 RSN 值一样处理，要么只向对等 RLC 实体发送一个 RESET ACK PDU。

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：

- 1) 将 HFN 的值置为接收到的 RESET ACK PDU 的 HFNI 字段中的值;
- 2) 将 9.4 节中的各个变量恢复为初始值;
- 3) 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;
- 4) 复位可配置的变量到其配置值;
- 5) 丢弃 AM RLC 接收侧所有的 RLC PDU;
- 6) 丢弃 AM RLC 实体发送侧复位之前发送的 RLC SDU;
- 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。

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 11.4.4.1) 处理。

11.4.6.2 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。

- 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);
- 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);
- 3) 复位可配置参数到其初始配置值;
- 4) 丢弃 AM RLC 接收侧所有的 RLC PDU;
- 5) 丢弃 AM RLC 实体发送侧复位之前发送的 RLC SDU;
- 6) 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证实模式下运行的两个 RLC 对等实体之间状态信息的转移。图 29 说明了状态报告传送的基本过程。一个状态报告包括一个或几个 STATUS PDU。

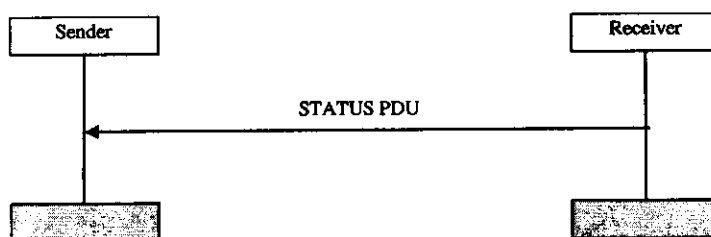


图 29 STATUS PDU 传送过程

如果上行链路配置了 2 条逻辑信道，控制 PDU 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 优先权高于数据 AMD PDU。

11.5.2 发起

如果下列触发条件发生：

- 1) 在接收到的 AMD PDU 中，轮询检测比特设置为 1。
- 2) 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。
- 3) 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。

如果接收者位于控制平面，接收者将在 DCCH 上传送状态报告，如果接收者位于用户平面，则接收者将在 DTCH 上传送状态报告。

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：

- 1) 如果未激活“STATUS Prohibit”和“EPC mechanism”功能，接收实体将在状态报告中证实丢失 AMD PDU 的信息和正确接收到的至少 $VR(R)$ 个 AMD PDU 的信息。
- 2) 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。
- 3) 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。
- 4) 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。
- 5) 如果所有的 SUFI 要包含在一个 STATUS PDU 中，使用一种允许的 PDU 的大小构成一个状态报告。
- 6) 如果 SUFI 字段不能填满整个 STATUS PDU，则将插入填充部分。如果 PDU 包含填充部分，最后一个 SUFI 字段将是 ACK SUFI 或 No More SUFI。如果在 STATUS PDU 中没有填充，就不必在 STATUS PDU 中包含 NO_MORE SUFI 或 ACK SUFI。

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 节)。如果 PDU 的 $SN \geq VR(H)$ ，则不给出任何信息。

11.5.3 发送实体对 STATUS PDU 的接收

当接收到 STATUS PDU/piggybacked STATUS PDU，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 的否定证实，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 中被否定证实了多次，则只重传一次 AMD PDU。

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

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

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 异常情况

11.5.4.1 $VR(EP)$ 为 0 并且请求的 PDU 没有收到

如果使用了 EPC 机制并且 $VR(EP)$ 等于 0，但是没有收到所有要求重传的 PDU，接收者将：

重传 STATUS 报告。重传的 STATUS 报告可以包含新的或不同的 SUFI 字段，目的是指示了一些以前丢失的 PDU 已经收到，和一些新的 PDU 已经丢失。

11.6 显式信令过程的 SDU 丢弃

11.6.1 概述

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

发送实体对于一个在证实模式的 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。

图 30 说明了利用显式信令进行 SDU 丢弃功能的基本过程。

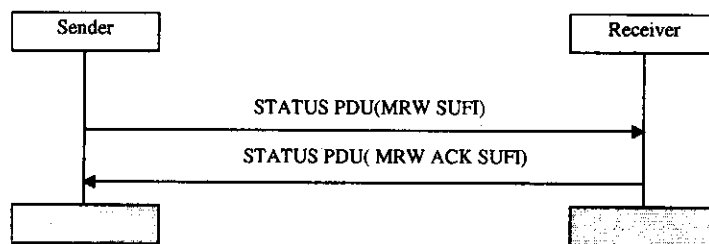


图 30 利用显式信令的 SDU 丢弃

11.6.2 发起

满足下列条件之一时，发送者启动这个过程。

1) 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。

2) 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。

3) 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序列号 SN 属于 $[VT(A), X]$ 的 AMD PDU 中的全部 SDU，其中 X 是 $VT(DAT)$ 大于 MaxDAT 的 AMD PDU 的序列号。

发送方也将丢弃所有包含丢弃 SDU 分段的 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 $SN_MRW_{LENGTH} > VT(S)$ ，则更新 $VT(S)$ 值为 SN_MRW_{LENGTH} 。同时启动定时器 $Timer_MRW$ 。

当定时器“ $Timer_MRW$ ”正在运行时，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 状态 PDU 内容设置

发送方如果配置了“发送 MRW”：

1) 若一个 AMD PDU 中包含最后一个被丢弃的 SDU，并且其长度指示 LI 也在该 PDU 中，同时该 PDU 中没有新的 SDU，设置该 MRW SUFI 的最后一个 SN_MRW_i 字段的值为该 AMD PDU 的序号+1， N_{LENGTH} 字段的值为“0000”。

2) 否则设置该 MRW SUFI 的最后一个 SN_MRW_i 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 N_{LENGTH} 个长度指示字段指示的字节。

3) N_{LENGTH} 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 N_{LENGTH} 个长度指示字段指示的字节。

4) 把 MRW SUFI 中其他的每个 SN_MRW_i 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。

发送方如果没配置“发送 MRW”：

1) 如果一个 AMD PDU 中包含最后一个将被接收方丢弃的 SDU，并且其长度指示 LI 也在该 PDU 中，同时该 PDU 中没有新的 SDU，设置该 MRW SUFI 的最后一个 SN_MRW_i 字段的值为该 AMD PDU 的序号+1， N_{LENGTH} 字段的值为“0000”。

2) 否则设置该 MRW SUFI 的最后一个 SN_MRW_i 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 N_{LENGTH} 个长度指示字段指示的字节

3) N_{LENGTH} 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 N_{LENGTH} 个长度指示字段指示的字节。

4) 并可选项的把 MRW SUFI 中其他的每个 SN_MRW_i 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。

如果 MRW SUFI 只包含一个 SN_MRW_i ，并且 SN_MRW_i 的值大于等于

VT(A)+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_MRW_i 字段的数量。其中 SN_MRW_i 字段的值应在 VT(A)和 VT(A)+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:

- 1) 如果接收 MRW SUFI 中的 LENGTH 字段为“0000”, 考虑 SN_MRW_i 大于等于 VR(R); 否则考虑 SN_MRW_i 小于 VR(MR)。
- 2) 考虑其他的 SN_MRW_i 按序排列, SN_MRW_i 大于等于 SN_MRW_{i-1}。
- 3) 丢弃直到并包括序列号为 SN_MRW_{LENGTH-1} 的 PDU 在内的全部 PDU。
- 4) 如果接收 MRW SUFI 中的 N_{LENGTH} 字段设为“0000”, 从序列号为 SN_MRW_{LENGTH} 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 N_{LENGTH} 个长度指示的数据字节。

如果接收 MRW SUFI 中的 LENGTH 字段不设为“0000”, 丢弃序列号为 SN_MRW_{LENGTH} 的 AMD PDU 中从第一个直到 (并包括) 第 N_{LENGTH} 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 节组合 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_MRW_{LENGTH}, 设置 MRW_ACK SUFI 中的 N 字段等于 SN_MRW_{LENGTH} 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:

- 1) 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。
- 2) 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_MRW_{LENGTH} 并且 N 等于在发送端发送的 MRW_SUFI 中的 N_{LENGTH} 的值。
- 3) 接收到一个包含一个 MRW_ACK SUFI 的 STATUS PDU/piggyback STATUS PDU。该 PDU 指示到序列号为发送 MRW SUFI 的 SN_MRW_{LENGTH} 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 废弃的/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：

- 1) 如果定时器 Timer_MRW 没激活。
- 2) 如果接收到的 MRW_ACK SUFI 中 SN_ACK 小于被发送的 MRW SUFI 中 SN_MRW_{LENGTH}。
- 3) 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_MRW_{LENGTH}，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 N_{LENGTH}。
- 4) 如果接收到的 MRW_ACK SUFI 中的 SN_ACK 字段大于发送的 MRW SUFI 的 SN_MRW_{LENGTH}，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，导致数据传输速率效率的降低，为此需要解决该缺陷。针对该缺陷，可采用以下四种更新发送端发送窗口方案。

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 31 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 的具体构成见图 32，

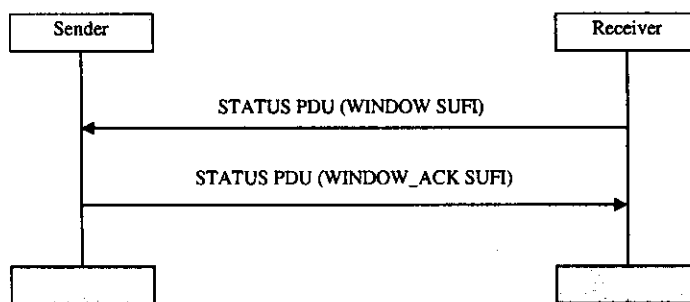


图 31 Window_ACK SUFI 作为 Window SUFI 的确认信息

11.7.2.1 Window_ACK SUFI 的构成

Window_ACK SUFI 的描述如下在 9.2.2.11 节的 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	证实(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	移动接收窗口证实(MRW_ACK)
1000	窗口大小证实(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 32 所示。

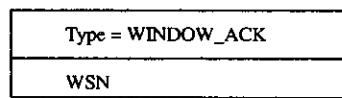


图 32 状态 PDU 中的 WINDOW_ACK 字段

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:

1) 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。

2) 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）。

3) 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 $VR(WIN) = \text{MaxWIN}$ ，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 节)。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 $VR(WIN) = \text{MaxWIN}$ ，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 $VR(WIN) < \text{MaxWIN}$;
- 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 发送端通过检测发送变量自行恢复发送窗口的方案

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:

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。

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 $T1=60ms$)。

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)。

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 $T2=80ms$)。

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)。

具体流程参见图 33。

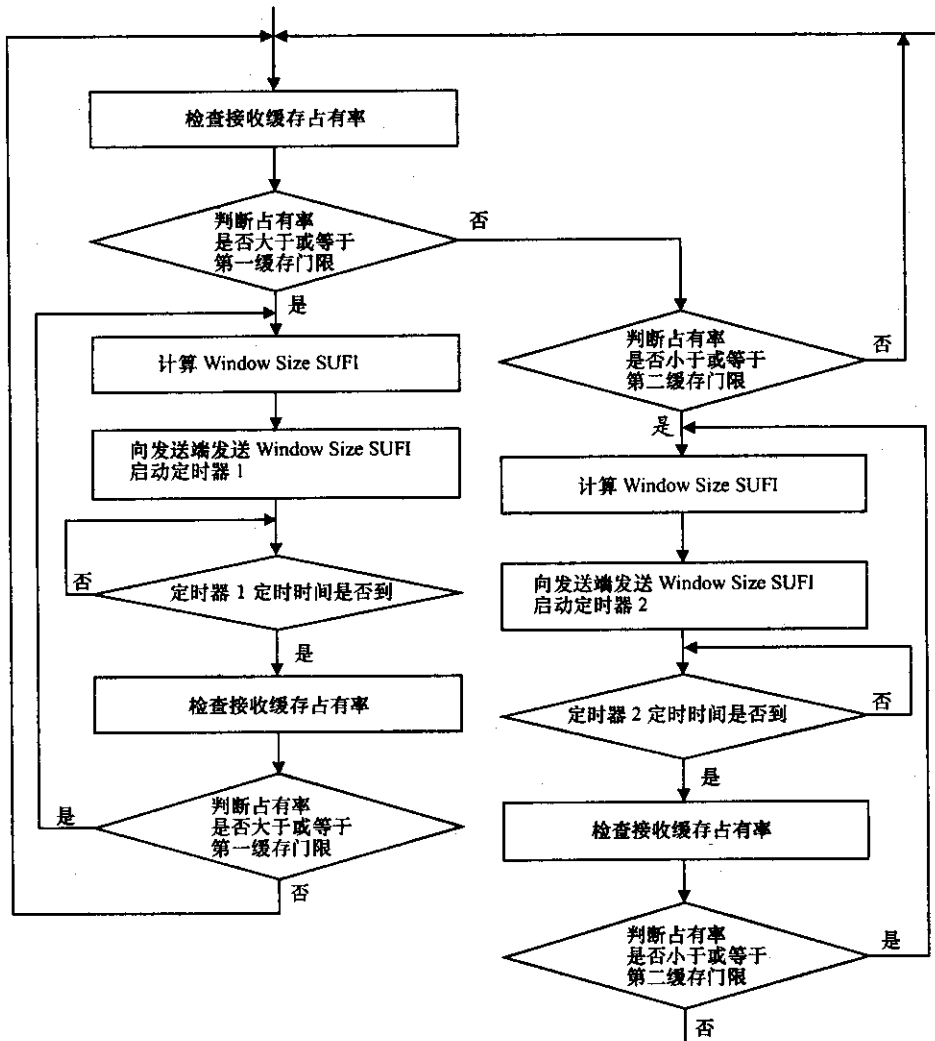


图 33 接收端通过检测接收缓存更新发送窗口的流程

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 Size 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 值充为 4095；
- 若接收端无需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。

该定时器超时，一个新的包含 WINDOW SUFI 的状态 PDU 需要被发送时，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 的过程，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 值充为 4095；
- 若接收端无需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：

$$K = \frac{(4096 + VT(S) - VT(A)) \bmod 4096}{\text{Configured_Tx_Window_Size}} \times 100$$

$$K \leq N$$

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 的序列号，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 的序列号和 Window Size sufi 中参数 WSN 的值。当接收到的来自 RNC 的状态 PDU 表明了这一被记录序列号的 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 的序列号，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 的序列号和 Window Size sufi 中参数 WSN 的值。当接收到的来自 RNC 的状态 PDU 表明了这一被记录序列号的 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 的序列号。当接收到的来自 RNC 的状态 PDU 表明了这一被记录序列号的 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 的序列号。当接收到的来自 RNC 的状态 PDU 表明了这一被记录序列号的 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:

- 1) (UE 侧)RLC 接收来自对端(RNC)的数据, 有转 2); 否则转 1)。
- 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)。
- 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)。
- 4) 记录所有丢失 PDU 的序列号, 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)。
- 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 序列号, 转 7); 否则转 6)。
- 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, 记录此特定 PDU 序列号, 转 7)。
- 7) 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)。
- 8) 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)。
- 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 7); 否则重新发送 Window Size sufi, 转 5);
- 10) 构造构造含有 Window Size sufi 的状态 PDU, WSN 等于当前丢失 4095, 转 11)。
- 11) 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 序列号, 转 13); 否则转 12)。
- 12) 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, 记录此特定 PDU 序列号, 转 13)。
- 13) 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 Window Size sufi, 转 11)。

参考文献

- [1] 3GPP TS 25.401: "RAN Overall Description"
- [2] 3GPP TR 25.945: "Vocabulary for the UTRAN"
- [3] 3GPP TS 25.301: "Radio Interface Protocol Architecture"
- [4] 3GPP TS 25.302: "Services Provided by the Physical Layer"
- [5] 3GPP TS 25.303: "UE Functions and Inter-Layer Procedures in Connected Mode"
- [6] 3GPP TS 25.304: "UE Procedures in Idle Mode"
- [7] 3GPP TS 25.321: "MAC Protocol Specification"
- [8] 3GPP TS.25.331: "RRC Protocol Specification"

注：参考的 3GPP 规范是 3GPP R99 2002 年 12 月版。
