**文件编号：LTE\_HeNB\_MAC\_HARQ\_GYSJ\_V1.0**

**TD-LTE HeNB协议栈软件系统**

**MAC\_HARQ模块**

**概要设计说明书**

拟制：张倩

时间：2010-10-27

**中国科学院计算技术研究所**

**无线通信技术研究中心**

**软件组**

**LTE协议栈研发项目组**

**修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文件编号 | 版本号 | 拟制人/  修改人 | 拟制/修改日期 | 更改理由 | 主要更改内容  （写要点即可） |
|  | 1.0 | 张倩 | 2010-10-27 |  |  |

修改列表：



本文档的程序或内容受版权法的保护，未经中科院计算所的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

**目录**

[TD-LTE HeNB协议栈软件系统 1](#_Toc291492697)

[1 引言 1](#_Toc291492698)

[1.1 编写目的 1](#_Toc291492699)

[1.2 背景 1](#_Toc291492700)

[1.3 定义 1](#_Toc291492701)

[1.4 参考资料 1](#_Toc291492702)

[2 模块概述 2](#_Toc291492703)

[2.1 特性列表 2](#_Toc291492704)

[2.2 模块功能描述 2](#_Toc291492705)

[2.3 与其他模块的关系 3](#_Toc291492706)

[3 总体设计 4](#_Toc291492707)

[3.1 基本设计 4](#_Toc291492708)

[3.2 流程设计 5](#_Toc291492709)

[3.2.1 上行HARQ处理流程 5](#_Toc291492710)

[3.2.2 接收ACK/NACK消息流程 5](#_Toc291492711)

[3.2.3 下行HARQ处理流程 6](#_Toc291492712)

[3.3 模块结构与子模块划分 6](#_Toc291492713)

[3.4 子模块结构设计 7](#_Toc291492714)

[3.5 全局变量说明 7](#_Toc291492715)

[4 接口设计 8](#_Toc291492716)

[4.1 外部接口 8](#_Toc291492717)

[4.1.1 与MAC数据接收子模块的接口 8](#_Toc291492718)

[4.1.1 与lowmac子模块的接口 8](#_Toc291492719)

[4.1.2 与MAC数据发送子模块的接口 9](#_Toc291492720)

[4.1.3 与调度子模块的接口 10](#_Toc291492721)

[4.1.4 与MAC管理模块的接口 11](#_Toc291492722)

[4.2 内部接口 12](#_Toc291492723)

[5 数据结构设计 12](#_Toc291492724)

[5.1上行HARQ数据结构 12](#_Toc291492725)

[5.2下行HARQ数据结构 13](#_Toc291492726)

[6 错误处理 14](#_Toc291492727)

[7. 其他说明 15](#_Toc291492728)

**模块概要设计说明书**

# 1 引言

## 1.1 编写目的

本说明书对LTE HeNB端MAC子系统HARQ模块的主要功能以及子模块划分进行简单的说明，阐述HeNB端MAC子系统HARQ模块的设计思想以及其工作流程。本说明书的读者为LTE HeNB端MAC子系统HARQ模块设计、编码人员、测试人员、项目组负责人员、实验室主任及相关项目管理人员。

编写本说明书的目的在于

* 为开发人员提供依据。
* 为修改和维护本系统提供条件。
* 项目负责人将根据本文档计划和控制系统设计、开发的全过程。

## 1.2 背景

软件系统名称：TD-LTE HeNB协议栈软件系统

软件开发者：无线通信技术研究中心LTE协议栈研发组

开发语言：C语言

硬件平台：

软件平台：linux 2.6

## 1.3 定义

无

## 1.4 参考资料

1. 《3GPP TS 36.321, "Medium Access Control (MAC) protocol specification", Rel. 8, V 8.9.0》
2. 《3GPP TS 36.300, "Overall description", Rel. 8, V 8.b.0》
3. 《3GPP TS 36.331, "Radio Resource Control (RRC)", Rel. 8, V 8.9.0》

# 2 模块概述

## 2.1 特性列表

LTE系统中，在上行链路系统中采用的是同步HARQ,可以是自适应的也可以是非自适应的；在下行链路系统中采用的是异步自适应的HARQ技术。

* 下行HARQ特点：

1. 异步自适应HARQ，异步HARQ依据空口条件可以提供灵活的重传调度；
2. PDCCH将HARQ进程号和新重传指示等信息发送给UE；
3. 可以支持一个子帧的多个HARQ进程。
4. 下行的传输反馈ACK/NACK是在PUCCH或PUSCH上发送。

* 上行HARQ特点：

1. 重传的HARQ数据是在预先确定的周期间隔上发送，不需额外信令指示eNB接收；
2. 每个UE都配置了最大重传调度，eNB需要考虑这个参数来进行重传调度；
3. 某一时刻只能支持一个HARQ进程。
4. 上行的传输反馈ACK/NACK是在PHICH上发送；
5. 自适应重传和非自适应重传的通过基站的调度结果决定。

## 2.2 模块功能描述

|  |  |
| --- | --- |
| 功能 | 功能描述 |
| HARQ初始化，重置，模块销毁。 | 得到基站上层为每个UE所配置的HARQ配置参数。负责初始化模块，模块最终的销毁。 |
| 上行HARQ功能 | 实现上行HARQ功能，并负责对上行数据的接收反馈。并维护每个UE的上行HARQ实体，提供给上行调度相关参数。完成上行同步HARQ的功能。 |
| 下行HARQ功能 | 实现下行HARQ功能，负责接收下行数据的反馈消息。并维护每个UE的下行HARQ实体，提供给下行调度相关参数。完成下行异步HARQ的功能。 |

## 2.3 与其他模块的关系



图1 MAC层子系统图



图2 HARQ模块与其它模块接口关系图

* 与MAC接收模块的关系
* MAC接收模块接收上行数据通知上行HARQ实体。做上行HARQ处理，根据解码是否成功来决定回复ACK/NACK。
* 与MAC发送模块的关系
* 发送下行数据做下行HARQ处理。根据新传指示来判断是新传数据还是重传数据，并记录HARQ相关信息。
* 提供给MAC发送模块接口，提供PHICH信道发送的ACK/NACK消息
* 与MAC管理模块的关系
* 提供给MAC管理模块接口，配置基站和每个UE的HARQ配置参数。
* 与MAC调度模块的关系
* 提供给MAC调度模块接口，得到调度相关参数（需要细化）。

# 3 总体设计

## 3.1 基本设计

HARQ模块完成基站端的上行HARQ和下行HARQ处理。对于基站来说，可以认为每一个UE都有的一个上行HARQ实体和一个下行HARQ实体，而各个UE间的上行HARQ实体或下行HARQ实体之间相对独立。每一个HARQ实体包括多个HARQ进程，每一个HARQ进程有专属的HARQ进程ID号。对于下行HARQ进程都对应一个HARQ buffer来缓存下行数据以备重传。

根据MAC接收与发送过程，可将HARQ设计为以下几个流程：

1. 上行HARQ处理流程。
2. 接收ACK/NACK消息流程。
3. 下行HARQ处理流程。

## 3.2 流程设计

### 3.2.1 上行HARQ处理流程

MAC接收模块收到PUSCH上行数据，调用接口函数do\_ulharq()进行上行HARQ处理。处理流程如下。

图3 上行HARQ处理流程图

### 3.2.2 接收ACK/NACK消息流程

lowmac模块收到PUSCH或PUCCH上的ACK/NACK消息行数据，调用接口函数process\_rcv\_ack（），进行下行HARQ处理。处理流程如下。

图4 接收ACK/NACK处理流程图

### 3.2.3 下行HARQ处理流程

MAC发送模块在发送PDSCH数据时做下行HARQ处理。



图5 下行HARQ处理流程图

## 3.3 模块结构与子模块划分



图6 HARQ接口调用关系图

## 3.4 子模块结构设计

模块结构如图，子模块划分如下。

* HARQ管理子模块

供给MAC管理模块接口，得到基站和每个UE的HARQ配置参数。负责初始化模块，模块最终的销毁，数据结构的内存分配与释放提。

* 上行HARQ处理子模块

实现上行HARQ功能，并负责对上行数据的接收反馈，即发送ACK/NACK消息。并维护每个UE的上行HARQ实体，提供给上行调度相关参数。

* 下行HARQ处理子模块

实现下行HARQ功能，负责接收ACK/NACK消息。并维护每个UE的下行HARQ实体，提供给下行调度相关参数。

## 3.5 全局变量说明

# 4 接口设计

## 4.1 外部接口

### 4.1.1 与MAC数据接收子模块的接口

#### 4.1.1.1 需要MAC数据接收模块提供的接口

无

#### 4.1.1.2 提供给MAC数据接收模块的接口

**1. 函数接口**

* **INT32 do\_ulharq(UINT16 rnti, INT8 dec\_result, UINT32 data\_size,**

**ResourcePhyInfo \*phy\_p)**

**函数描述**

接收到上行数据通知上行HARQ实体。根据RNTI值找到该UE的上行HARQ进程后，若解码成功回复ACK，不成功回复NACK，并在该上行HARQ进程中标明等待重传。

**输入**

rnit：RNTI值；

dec\_result：解码成功标志；

data\_size：接收数据大小；

phy\_p:这次传输的物理资源信息。

**输出**

无

**返回**

0/-1，操作是否成功。

### 4.1.1 与lowmac子模块的接口

#### 4.1.1.1 需要lowmac模块提供的接口

无

#### 4.1.1.2 提供给lowmac模块的接口

* **INT32 process\_rcv\_ack(AckLst \*ack\_lst\_p)**

**函数描述**

接收到ACK/NACK list通知相应下行HARQ实体。根据RNTI值找到该UE, 再根据收到ACK/NACK的帧号和子帧号找到下行HARQ进程后收到NACK和ACK做不同处理，即NACK情况时标明需重传，ACK情况说明下行发送成功。

**输入**

AckLst \*ack\_lst\_p：是指向ACK/NACK list的指针。

**输出**

无

**返回**

0/-1，操作是否成功。

### 4.1.2 与MAC数据发送子模块的接口

#### 4.1.2.1 需要MAC数据发送子模块提供的接口

无

#### 4.1.2.2 需要MAC数据发送子模块模块提供的接口

**1. 函数接口**

* **INT32 do\_dlharq(UINT16 rnti, UINT8 ndi, UINT32 tb\_size,**

**UINT8 pdu\_p, TbHarqParam \*tbharqparam\_p)**

**函数描述**

发送下行数据做下行HARQ处理。根据RNTI值找到该UE的下行HARQ进程后，根据新传指示来判断是新传数据还是重传数据。在记录HARQ相关信息。

**输入**

rnit：RNTI值；

ndi：新传标志；

tb\_size：数据大小；

**输出**

pdu\_p：返回重传数据

tbharqparam\_p：HARQ信息

**返回**

0/-1，操作是否成功。

* **INT32 get\_tx\_acklst( AckLst \*ack\_lst)**

**函数描述**

得到一个在PHICH发送ACK/NACK的链表。

**输入**

无

**输出**

得到一个在PHICH发送ACK/NACK的链表。第一层是RNTI，第二层是HARQ进程。

**返回**

0/-1，操作是否成功。

### 4.1.3 与调度子模块的接口

#### 4.1.3.1 需要调度模块提供的接口

无

#### 4.1.3.2 提供给调度模块的接口

**1. 函数接口**

* **INT32 get\_ulharq\_sched\_info(UINT16 rnti, UINT8 \*retran\_ind, UINT32 \*tb\_size)**

**功能描述**：

提供给上行调度，某一RNTI，当前上行HARQ情况。

**输入**：

rnti,：RNTI值。

**输出**：

retran\_ind：重传请求标志

tb\_size：重传tb大小

**返回**：

0/-1，操作是否成功。

* **INT32 get\_dlharq\_sched\_info(UINT16 rnti, UINT8 \*state, UINT32 \*tb\_size)**

**功能描述**：

提供给下行调度，某一RNTI，当前下行HARQ情况。

**输入**：

rnti,：RNTI值。

**输出**：

state：下行HARQ进程状态

tb\_size：重传tb大小

**返回**：

0/-1，操作是否成功。

### 4.1.4 与MAC管理模块的接口

#### 4.1.4.1 需要MAC管理模块提供的接口

无

#### 4.1.4.2 提供给MAC管理模块的接口

1. **函数接口**

* **INT32 config\_ue\_harq(UINT16 rnti, HarqParam harq\_conf\_param)**

**函数描述**

基站记录为每一个UE的HARQ配置。

**输入**

rnit： RNTI值；

harq\_conf\_param：HARQ配置信息（上下行最大重传次数，计时器等）。

**输出**

无

**返回**

0/-1，操作是否成功。

## 4.2 内部接口

# 5 数据结构设计

## 5.1上行HARQ数据结构



图7 上行HARQ数据结构

/\*一个UE的上行HARQ实体\*/

typedef struct {

UlProcInfo \*ulproc\_p; /\*上行HARQ实体中的HARQ进程指针\*/

UINT8 process\_num; /\*上行HARQ实体的总进程数\*/

UINT8 ul\_max\_harq\_tx; /\*该UE上行HARQ进程所配置的重传最大数\*/

UINT8 max\_harq\_msg3\_tx; /\*该UE上行MSG3配置的重传最大数\*/

} UlHarqInfo;

/\*一个UE的上行HARQ进程\*/

typedef struct {

ResourcePhyInfo phy\_info;

UINT32 tb\_size; /\*TB大小\*/

UINT8 retran\_ind; /\*重传调度指示\*/

UINT8 current\_tx\_num; /\*传输失败的次数\*/

} UlProcInfo;

/\*物理资源PRB信息\*/

typedef struct {

UINT16 rb\_start; /\*起始RB\*/

UINT16 rb\_num; /\*RB个数\*/

UINT8 mcs; /\*MCS\*/

UINT8 rv; /\*RV\*/

} ResourcePhyInfo;

## 5.2下行HARQ数据结构



图8 下行HARQ数据结构

/\*一个UE的下行HARQ实体\*/

typedef struct {

DlProcInfo \*dlproc\_p; /\*下行HARQ实体中的HARQ进程指针\*/

UINT8 total\_proc\_num; /\*下行HARQ实体的总进程数\*/

UINT8 retran\_tb\_size; /\*指示所有重传的下行HARQ进程的总TB大小\*/

UNIT8 retran\_proc\_num; /\*指示现在重传进程的个数\*/

UINT8 idle\_proc\_num; /\*指示现在空闲进程的个数\*/

UINT8 dl\_max\_harq\_tx; /\*该UE下行HARQ进程所配置的重传最大数\*/

UINT8 max\_sps\_proc\_num; /\*该UE下行SPS HARQ进程最大数\*/

} UlHarqInfo;

/\*一个下行HARQ进程\*/

typedef struct {

UINT32 dl\_tx\_time; /\*下行数据发送的时间\*/

Timer\_t wait\_ack\_timer; /\*等待ACK的计时器\*/

PduInfoLst \*pdulst\_p; /\*缓存的下行数据指针\*/

UINT32 tb\_size; /\*下行数据大小\*/

UINT8 current\_tx\_num; /\*现在下行发送的次数\*/

UINT8 rv; /\*RV\*/

UINT8 state; /\*下行进程的状态指示，idle，wait，retran三种状态\*/

} DlProcInfo;

# 6 错误处理

# 7. 其他说明

没有在上述章节中描述的问题。