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

**TD-LTE HeNB协议栈软件系统**

**MAC-随机接入模块**

**概要设计说明书**

拟制：李亚楠

时间：2011-03-25

**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 | 李亚楠 | 2011-03-25 | 建立 | 无 |

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**

[1 引言 1](#_Toc290901326)

[1.1 编写目的 1](#_Toc290901327)

[1.2 背景 1](#_Toc290901328)

[1.3 定义 1](#_Toc290901329)

[1.4 参考资料 1](#_Toc290901330)

[2 模块概述 2](#_Toc290901331)

[2.1 特性列表 2](#_Toc290901332)

[2.2 模块功能描述 2](#_Toc290901333)

[2.3 与其他模块的关系 3](#_Toc290901334)

[3 总体设计 4](#_Toc290901335)

[3.1 基本设计 4](#_Toc290901336)

[3.2.1前导接收流程 5](#_Toc290901337)

[3.2.2 RAR准备流程 5](#_Toc290901338)

[3.2.3 Msg3接收及处理流程 6](#_Toc290901339)

[3.3 模块结构与子模块划分 7](#_Toc290901340)

[3.4 子模块结构设计 7](#_Toc290901341)

[3.5 全局变量说明 8](#_Toc290901342)

[4 接口设计 9](#_Toc290901343)

[4.1 外部接口 9](#_Toc290901344)

[4.1.1 与LowMAC模块的接口 9](#_Toc290901345)

[4.1.2 与MAC 发送模块的接口 9](#_Toc290901346)

[4.1.3 与MAC 接收模块的接口 10](#_Toc290901347)

[4.1.4 与核心数据管理模块的接口 11](#_Toc290901348)

[4.1.5 与调度模块的接口 13](#_Toc290901349)

[4.2 内部接口 13](#_Toc290901350)

[5 数据结构设计 15](#_Toc290901351)

[6 错误处理 17](#_Toc290901352)

[7. 其他说明 18](#_Toc290901353)

**模块概要设计说明书**

# 1 引言

## 1.1 编写目的

本文档将作为MAC随机接入模块详细设计编写的依据，概要说明了本模块功能、结构、函数定义，以及与其它模块的接口。本说明书的读者为LTE HeNB端MAC随机接入模块设计、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：暂无

软件平台：Linux

## 1.3 定义

1. RA：random access
2. CE：control element
3. TC-RNTI：temporary C-RNTI
4. RAR：random access response

## 1.4 参考资料

1. 《3GPP TS 36.321 v8.9.0: "E-UTRA Medium Access Control protocol specification "》；
2. 《3GPP TS 36.300 v8.12.0: " E-UTRA and E-UTRAN Overall description "》；
3. 《3GPP TS 36.331 v8.11.0:" E-UTRA Radio Resource Control protocol specification "》；
4. 《LTE\_HeNB协议栈软件系统MAC随机接入模块理解报告\_v1.0》；

# 2 模块概述

## 2.1 特性列表



图1 随机接入模块位置关系图

随机接入模块位于MAC子系统数据收发模块中，主要负责随机接入信息的接收、处理。该模块功能可概括描述接收并处理来自Lowmac层的前导信息和MAC接收模块的Msg3信息，能够根据MAC发送模块的rnti获取相应的RAR信息。

## 2.2 模块功能描述

MAC随机接入模块处于MAC层。

该模块能够接收并处理来自Lowmac层的前导信息和MAC接收模块的Msg3信息，能够根据MAC发送模块的rnti获取相应的RAR信息。当接收到Lowmac层的前导信息时，该模块能够根据rnti完成实体建立；当收到接收模块的Msg3时，该模块能够根据Msg3的内容进行相应实体的处理工作；当收到MAC发送模块的获取RAR消息时，该模块能够根据rnti类型来执行RAR相关信息的获取，并将RAR信息递交给MAC发送模块。

## 2.3 与其他模块的关系



图2随机接入模块与其它模块接口关系图

随机接入模块与其它模块的输入输出关系如图2所示。

* 随机接入模块与lowmac模块的接口

随机接入模块向lowmac模块提供接口rcv\_ra\_preamble（）；

* 随机接入模块与MAC接收模块的接口

随机接入模块向MAC接收模块提供接口rcv\_msg3（）、rcv\_crnti\_ce（）；

* 随机接入模块与MAC发送模块的接口

随机接入模块向MAC发送模块提供接口get\_rar（）；

* 随机接入模块与核心数据管理模块的接口

随机接入模块向核心数据管理模块不提供接口；核心数据管理模块向随机接入模块提供接口create\_ue\_entity（）、get\_available\_c\_rnti（）、delete\_ue\_entity（）、get\_rnti\_type（）、get\_rnti\_entity（）；

* 随机接入模块与调度模块的接口

随机接入模块向调度模块不提供接口；调度模块向随机接入模块提供接口inform\_rcv\_crnti\_ce（）。

# 3 总体设计

## 3.1 基本设计

随机接入过程主要分为基于竞争和基于非竞争的两类随机接入过程。对于这两类随机接入过程，在HeNB端MAC层随机接入模块主要负责随机接入前导的接收、随机接入响应信息的获取及Msg3消息的接收功能。

**3.2 流程设计**

在HeNB端的随机接入过程主要分为以下流程：

1. 前导接收流程
2. RAR准备流程
3. Msg3接收及处理流程

### 3.2.1前导接收流程



图5 前导接收及处理流程图

### 3.2.2 RAR准备流程



图6 RAR准备流程图

### 3.2.3 Msg3接收及处理流程



图7 Msg3接收处理流程图

## 3.3 模块结构与子模块划分



图8随机接入模块子模块划分图

随机接入模块子模块划分如图8所示，子模块划分如下。

* 前导处理子模块

供给lowmac模块接口，得到UE发送的前导及RA-RNTI。负责对接收前导的识别，相应实体的建立、删除，数据结构的内存分配与释放。

* RAR准备子模块

供给MAC发送模块接口，负责RAR信息的传递。

* Msg3处理子模块

供给MAC接收模块接口，得到UE发送的Msg3。负责对接收msg3的判断及相应tc-rnti实体的处理，对收到C-RNTI CE的随机接入情况进行c-rnti实体的查询及对调度模块进行C-RNTI情况的通知。

## 3.4 子模块结构设计

|  |  |  |
| --- | --- | --- |
| **函数名** | **功能** | **子模块：部分** |
| INT32 init\_ra() | 负责随机接入模块的初始化 | **RA管理** |
| INT32 rcv\_ra\_preamble(UINT16 ra\_rnti, UINT8 preamble\_index) | 从lowmac层接收RA-RNTI、preamble\_index，负责RA-RNTI链表的管理 | **前导处理** |
| RarMsgLst \*get\_rar(UINT16 rnti) | 为调度到的RA\_RNTI构造RAR消息内容 | **RAR准备** |
| INT32 rcv\_crnti\_ce(UINT16 tc\_rnti, UINT8\* msg\_p) | 从MAC 接收模块接收C-RNTI控制单元,进行C-RNTI实体的查找及TC-RNTI实体的处理 | **Msg3处理** |
| INT32 rcv\_ msg3 (UINT16 tc\_rnti, UINT16 msg\_len, UINT8 \*msg\_p) | 接收CCCH SDU，将CCCH SDU递交CE list。 | **Msg3处理** |
| INT32 cleanup\_ra() | 负责RA-RNTI实体及未处理TC-RNTI实体的删除。 | **RA管理** |

## 3.5 全局变量说明

/\* RA-RNTI message list table \*/

RaRntiMsgNode \*g\_ra\_rnti\_table [RA\_RNTI\_TBL\_SIZE];

/\* 供给调度模块的ra-rnti链表 \*/

ListType g\_ra\_rnti\_msg\_lst;

# 4 接口设计

## 4.1 外部接口

### 4.1.1 与LowMAC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

1. **函数接口**
2. **INT32 rcv\_ra\_preamble (UINT16 ra\_rnti, UINT8 preamble\_index)**

**函数描述**

接收到PRACH上发送的前导，记录该用户随机接入信息，并添加到RA用户队列中。

**输入**

ra\_rnti： RA\_RNTI值；

preamble\_index：接收到的前导序列码；

**输出**

无

**返回**

成功返回0，无内存错误返回-1，输入参数错误返回-2，系统参数错误返回-3。

### 4.1.2 与MAC 发送模块的接口

#### 4.1.2.1 需要MAC 发送模块提供的接口

无

#### 4.1.2.2 提供给MAC 发送模块的接口

1. **RarMsgLst \*get\_rar (UINT16 rnti)**

**函数描述**

该函数根据发送模块递交的rnti通过g\_ra\_rnti\_table[rnti]找到对应的ra-rnti链表，然后对该链表中各rar节点进行调度结果查询。若获得调度，则对该rar节点进行RAR信息完善，然后将其从原ra-rnti链表中删除并添加到scheduled\_rar链表中。进行TC-RNTI值的获取及TC-RNTI实体建立后，将scheduled\_rar链表递交给发送模块。

**输入**

rnti：RA\_RNTI值

**输出**

无

**返回**

成功返回0，无内存错误返回-1，输入参数错误返回-2，系统参数错误返回-3。

### 4.1.3 与MAC 接收模块的接口

#### 4.1.3.1 需要MAC 接收模块提供的接口

无

#### 4.1.3.2 提供给MAC 接收模块的接口

1. **INT32 rcv\_msg3 (UINT16 tc\_rnti, UINT16 msg\_len, UINT8 \*msg\_p)**

**函数描述**

找到该tc-rnti对应的TC-RNTI实体，将接收的msg3即CCCH SDU递交给TC-RNTI实体中的CE list。

**输入**

tc\_rnti：TC\_RNTI值；

msg\_len：CCCH消息长度

msg\_p：ccch sdu所在消息指针；

**输出**

无

**返回**

成功返回0，无内存错误返回-1，输入参数错误返回-2，系统参数错误返回-3。

1. **INT32 rcv\_crnti\_ce (UINT16 tc\_rnti, UINT8\* msg\_p)**

**函数描述**

对具有C-RNTI终端发来的c-rnti控制单元，首先查找tc-rnti对应的TC-RNTI实体，根据接收的CRNTI控制单元，进行对应C-RNTI实体的查询。若查找到对应的C-RNTI实体则删除TC\_RNTI实体。若未找到C-RNTI实体则向上层进行报错处理。

**输入**

**tc\_rnti**：TC\_RNTI值；

**msg\_p**：C\_RNTI控制单元消息指针；

**输出**

无

**返回**

成功返回0，无内存错误返回-1，输入参数错误返回-2，系统参数错误返回-3。

### 4.1.4 与核心数据管理模块的接口

#### 4.1.4.1 需要核心数据管理模块提供的接口

1. **函数接口**
2. **INT32 create\_ue\_entity(UINT16 rnti)**

**功能描述**

根据给出的RNTI值创建一个普通的RNTI实体，并且初始化这个RNTI实体，然后将这个RNTI放置在RNTI指针数组和RNTI链表之中。

**输入**

rnti： RNTI的标识号

**输出**

无

**返回**

成功： 0

失败：

1. **INT32 delete\_ue\_entity(UINT16 rnti)**

**功能描述**

UE与HeNB断开连接后删除该UE实体，内存释放C-RNTI回收。

**输入**

rnti： RNTI标识号

**输出**

无

**返回**

1. **INT32 get\_available\_c\_rnti()**

**功能描述**

从保存未使用RNTI值的队列数组中获取当前可用而且未用的C-RNTI或者Temp-C-RNTI的值。

**输入**

无

**输出**

无

**返回**

成功： 取出的RNTI值

失败：

1. **UeRntiMsg \*get\_rnti\_entity(UINT16 rnti)**

**功能描述**

从Henb系统中查找到RNTI值对应的RNTI实体的指针。

**输入**

Rnti： RNTI的标识号

**输出**

无

**返回**

成功： RNTI指针

失败： NULL

1. **RntiType get\_rnti\_type(UINT16 rnti)**

**功能描述**

获得RNTI类型。

**输入**

Rnti： RNTI值

**输出**

无

**返回**

成功： RNTI类型

失败： -1

#### 4.1.4.2 提供给核心数据管理模块的接口

无

### 4.1.5 与调度模块的接口

#### 4.1.5.1 需要调度模块提供的接口

1. **INT32 inform\_rcv\_crnti\_ce（UINT16 rnti）**

**功能描述**

通知调度模块收到C-RNTI CE。

**输入**

Rnti： rnti 值

**输出**

无

**返回**

无内存错误返回-1，输入参数错误返回-2，系统参数错误返回-3。

#### 4.1.5.2 提供给调度模块的接口

无

## 4.2 内部接口

1. **INT32 init\_ra ()**

**函数描述**

初始化g\_ra\_rnti\_table []及g\_ra\_rnti\_msg\_lst。

**输入**

**无**

**输出**

无

**返回**

成功返回0，无内存返回-1。

1. **INT32 cleanup\_ra ()**

**函数描述**

清除ra-rnti实体。

**输入**

无

**输出**

无

**返回**

成功返回0，输入参数错误返回-2，系统参数错误返回-3。

# 数据结构设计



图9 g\_ra\_rnti\_msg\_lst结构图



图10 g\_ra\_rnti\_table[]及g\_scheduled\_rarnti\_table[]结构图

/\* this should be defined in UL Grant management \*/

typedef struct {

UINT16 fixed\_riv;

UINT8 hopping\_flag;

UINT8 truncated\_mcs;

UINT8 tpc\_pusch;

UINT8 ul\_delay;

UINT8 cqi\_request;

} RARUlGrant;

typedef struct {

INT32 ta;

RARUlGrant \*ul\_grant\_p;

UINT16 tc\_rnti; /\* tc-rnti sent to construct rar pdu \*/

} RarInfo;

typedef struct {

NodeType ln;

RarInfo \*rar\_msg\_p; /\* Pointer to RAR message of the RA-RNTI \*/

UINT8 rapid; /\* The RAPID of the RAR message \*/

} RarNode;

typedef struct { NodeType ln;

UINT16 bytes\_size; /\* total size of all rar node \*/ UINT16 ra\_rnti; /\* ra-rnti value \*/

ListType rar\_node\_lst; UINT8 backoff\_flag; /\* 0：no backoff\_ind；1：exist backoff\_ind \*/ UINT8 backoff\_ind; /\* backoff indication value \*/

}RaRntiMsgNode;

# 6 错误处理

# 7. 其他说明

暂未考虑内容：

1. 退避值的获取；