**文件编号：LTE-HeNB\_RRC\_REESTAB \_GYSJ\_V1.0**

**TD-LTE HeNB协议栈软件系统**

**RRC连接重建立模块**

**概要设计说明书**

拟制：张瑞明

时间：2011-9-06

**中国科学院计算技术研究所**

**下一代互联网研究中心**

**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） |
|  | 0.5 | 张瑞明 | 2011-9-6 | 无 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 注1：每次更改归档文件（指归档到组内及研究室的文件）时，需填写此表。  注2：文件第一次归档时，“更改理由”、“主要更改内容”栏写“无”。 | | | | | |

修改列表：

无。



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](#_Toc309111665)

[1.1编写目的 1](#_Toc309111666)

[1.2背景 1](#_Toc309111667)

[1.3定义 1](#_Toc309111668)

[1.4参考资料 1](#_Toc309111669)

[2 模块描述 2](#_Toc309111670)

[2.1 模块综述 2](#_Toc309111671)

[2.2 模块接口 2](#_Toc309111672)

[3 模块设计 3](#_Toc309111673)

[3.1 模块结构 3](#_Toc309111674)

[3.2 连接重建立响应流程 4](#_Toc309111675)

[3.3 子模块结构设计 5](#_Toc309111676)

[3.4 全局变量说明 5](#_Toc309111677)

[4 接口设计 5](#_Toc309111678)

[4.1 外部接口 5](#_Toc309111679)

[4.1.1 与决策控制模块的接口 5](#_Toc309111680)

[4.1.2 与定时器管理模块的接口 7](#_Toc309111681)

[4.1.3 与RRC发送模块的接口 8](#_Toc309111682)

[4.1.4与RRC核心数据管理模块的接口 8](#_Toc309111683)

[4.2 内部接口 9](#_Toc309111684)

[4.2.1 连接建立响应子模块 9](#_Toc309111685)

[5 数据结构设计 11](#_Toc309111686)

[6 错误处理 12](#_Toc309111687)

[7其他说明 13](#_Toc309111688)

**概要设计说明书**

# 1 引言

## 1.1编写目的

本文档将作为RRC连接重建立模块的代码编写的依据。详细说明了本模块功能、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RRC连接建立模块设计、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：中科院计算所无线通信技术研究中心-LTE协议栈研发项目组

开发语言：C语言

硬件平台：暂无

软件平台：Linux

## 1.3定义

1. RRC： Radio Resource Control。
2. RB: Radio Bearer

## 1.4参考资料

1. 《3GPP TS 36.300: "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2"》；
2. 《3GPP TS 36.331: "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification"》；

# 2 模块描述

## 2.1 模块综述

本模块必须完成：

- 重配SRB1；

- 重新激活AS安全不改变算法；

该模块能够接收并处理来自决策控制的连重接建立和停止连接重建立的命令，能够根据收到的对端的RRC消息类型做相应处理。当接收到发起连接重建立的命令时，该模块能够完成连接重建立过程的校验和准备工作；当收到决策控制消息时，该模块能够根据决策消息配置的类型来执行重建RRC消息并发送；可接受反馈消息。

该模块能够在执行RRC连接重建立过程中，根据当前执行状况，启动或停止特定的定时器，并修改RRC状态和RRC连接重建立过程的状态。

## 2.2 模块接口

**图1 RRC连接重建立模块与其它模块接口关系图**

RRC连接重建立模块与其它模块的关系如图1所示，其中箭头的方向表示调用方向。

* **与决策控制模块的关系:**

决策控制在收到对端RRC消息的事件后进行分析并做出决策，触发RRC连接重建立模块的相应过程。

* **与RRC核心数据管理模块的关系:**

（1）查询数据

RRC连接重建立模块查询核心数据结构中的信息，如CRNTI。

（2）修改数据

RRC连接重建立模块能够根据过程的执行状态修改核心数据结构中的相关信息（如UE状态，保存UE ID和系统信息等）。

（3）删除数据

RRC核心数据管理子系统保存系统信息、RRC处理过程中的配置信息以及在整个RRC系统运行过程中的全局信息，并向其它模块提供删除这些信息的功能，保证系统对信息处理的一致性。

（4）建立RB

通过对基站资源的划分为UE建立起SRB1实体。

* **与定时器管理模块的关系：**

RRC连接重建立模块使用定时器管理模块提供的功能来查询定时器状态、启动和停止定时器的功能。

* **与消息构造模块的关系：**

RRC连接重建立模块使用其RRC消息构造的功能根据ASN．1标准规定的格式，将输入的RRCf肖息内容构造为RRC PDU。

* **与消息发送模块的关系：**

消息发送模块向RRC连接重建立模块将要发送的RRC PDU递交至下层。

* **与状态管理模块的关系：**

状态管理模块维护RRC状态，提供查询和修改当前状态的功能，并在RRC状态改变时进行相应处理。

* **与无线资源配置模块的关系：**

RRM模块确保有用的无线资源有效利用和为了确保E-UTRAN接纳关于请求认证的无线资源而提供测量，而E-UTRAN内的RRM模块提供管理单或者多小区无线资源的方式。具体的功能是RBC，RAC，CMC，DRA，ICIC和LB功能。

# 3 模块设计

## 3.1 模块结构

本模块必须完成：UE与HeNB RRC连接重建立成功以实现重配SRB1，且仅为其恢复数据传输和重新激活AS安全不改变算法的功能。

**RRC连接重建立成功**：UE在基站端保存的状态恢复，并为UE在基站端重建立起SRB1；

**RRC连接重建立失败**：基站在无法重新接纳UE时，会使得RRC连接重建立失败。

## 3.2 连接重建立响应流程



**图2 连接建立响应子模块流程图**

1. 决策控制模块接收到*RRCConnectionReestablishmentRequest*消息触发该流程；
2. 保存请求消息中UE ID和建立原因等各IEs在RRC entity中；
3. 调用RRM模块接口判断该UE是否可以接入：
   1. 若没有相应的资源给该UE，构造*RRCConnectionReestablishmentReject*消息，发送*RRCConnectionReestablishment*消息给UE；
   2. 若有相应的资源给该UE，为该UE恢复SRB1，构造*RRC Connection Reestablishment*消息，并发送*RRCConnectionReestablishment*消息给UE，开启连接重建立响应定时器（自己定义）；
      1. 若定时器超时，则删除该UE实体；
      2. 若定时器未超时，则继续等待；

## 3.3 子模块结构设计

**1）连接重建立响应子模块**

* 保存UE信息

将**RRC Connection Reestablishment Reques**中UE ID和establishment cause保存在RRC实体当中；

* 响应消息发送

eNB在收到UE的请求后，给UE发消息

* 配置实体

为该UE恢复SRB1，并配置下层；

* 消息构造

通过对于资源分配的控制构造RRC Connection Reestablishment和RRC Connection ReestablishmentReject消息。

* 消息发送

发送构造好的RRC消息给UE

## 3.4 全局变量说明

# 4 接口设计

## 4.1 外部接口

列出本模块提供给其他模块的接口，以及本模块需要其他模块提供的接口。包括函数接口和全局变量接口等模块间操作所用到的任何内容。其中函数接口需要将输入、输出参数及返回值确定。其中箭头的方向表示调用方向。

### 4.1.1 与决策控制模块的接口

#### 4.1.1.1 需要决策控制模块提供的接口

无。

#### 4.1.1.2 提供给决策控制模块的接口

1. **函数接口**

说明函数功能、输入输出参数、内存分配和释放方式。

* **INT32 action\_fn\_handle\_conn\_reestab\_req\_msg(Rnti rnti，void\* event\_param)**

**功能描述**

该函数是接收到重建立连接请求时候执行处理线程的函数。

**输入**

rnti：标记UE的符号

RRCConnectionReestablishmentRequest消息

**输出**

无

**返回**

成功：0

失败：-1

* **INT32 send\_conn\_reestab \_msg(UINT16 rnti, HenbConfig \*henb\_cfg\_p,**

**RadioResourceConfigDedicated\_t \*radio\_res\_cfg\_dedi\_p)**

**功能描述**

该函数是接收到重建立请求时候响应线程的执行函数。

**输入**

rnti：标记UE的符号

**输出**

无

**返回**

成功：0

失败：-1

* **INT32 action\_fn\_handle\_conn\_reestab\_compl\_msg(Rnti rnti, void\*event\_param)**

**功能描述**

该函数是接收到重建立完成请求时候响应线程的执行函数。

**输入**

rnti：标记UE的符号

RRCConnectionReestablishmentComplete消息

**输出**

无

**返回**

成功：0

失败：-1

1. **全局变量接口**

无；

1. **数据结构接口**

无；

### 4.1.2 与定时器管理模块的接口

#### 4.1.2.1 需要定时器管理模块提供的接口

1、**函数接口**

* **INT32 start\_rrc\_timer(RrcEntity \*rrc\_entity\_p, INT32 time\_name)；**

**功能描述**

该函数启动定时器timer。

**输入**

rrc\_entity\_p：rrc实体；

Timers timer：timer名；

**输出**

无。

**返回**

0：成功；

-1：失败。

* **INT32 stop\_rrc\_timer(RrcEntity \*rrc\_entity\_p);**

**功能描述**

该函数停止定时器timer。

**输入**

Timers timer：timer名。

**输出**

无。

**返回**

0：成功；

-1：失败。

#### 4.1.2.2 提供给定时器管理模块的接口

无。

### 4.1.3 与RRC发送模块的接口

#### 4.1.3.1 需要RRC发送模块提供的接口

* **INT32 tx\_rrc\_pdu(Rnti rnti, RbType rb\_type,**

**INT32** **rb\_id, UINT8 \*rrc\_pdu\_p, INT32 msg\_len)**

**功能描述**

该函数用于发送RRC Connection Reestablishment消息。

**输入**

SrbType srb\_type：SRB的类型；

rb\_id：rb的认证；

UINT8 \*rrc\_pdu,：指向RRC消息的指针；

UINT16 msg\_len\_p：RRC消息的长度；

rnti：标记UE的符号。

**输出**

无。

**返回**

0：成功；

-1：失败。

**2、全局变量接口**

无。

**3、数据结构接口**

无。

#### 4.1.3.2 提供给RRC发送模块的接口

无。

### 4.1.4与RRC核心数据管理模块的接口

#### 4.1.3.1 需要RRC核心数据模块提供的接口

* **UINT8 get \_ncc(Rnti rnti)**

**功能描述**

该函数获得重建时安全参数。

**输入**

rnti：标记UE的符号。

**输出**

无。

**返回**

0：成功；

-1：失败。

**2、全局变量接口**

无。

**3、数据结构接口**

无。

#### 4.1.3.2 提供给RRC核心数据模块的接口

无。

## 4.2 内部接口

以子模块为单位，列出为实现模块功能，所需要的模块内部接口。包括函数接口、全局变量接口等任何函数间操作所用到的内容。其中函数接口需要将输入、输出参数及返回值确定。

### 4.2.1 连接建立响应子模块

* **INT32 construct\_conn\_reestab\_pdu（RadioResConfig\_t \*radio\_res\_config\_p,**

**NextHopChainingCount \*next\_hop\_chaining\_count\_p,**

**UINT8 \*rrc\_pdu\_p, UINT16\* msg\_len\_p）**

**功能描述**

该函数是为了构造RRC Connection Reestablishment消息。

**输入**

radio\_res\_config\_p：RRC Connection Reestablishment消息中的无线资源配置IE；

next\_hop\_chaining\_count\_p: RRC Connection Reestablishment消息中的KeNB key和NCC参数更新

**输出**

rrc\_pdu\_p：输出指向RRC消息的指针；

msg\_len：输出RRC消息的长度。

**返回**

0：成功；

-1：失败。

* **INT32 construct\_conn\_reestab\_reject\_pdu（Rnti rnti）**

**功能描述**

该函数是为了构造RRC Connection Reestablishment Reject消息。

**输入**

**输出**

rrc\_pdu\_p：输出指向RRC消息的指针；

msg\_len：输出RRC消息的长度。

**返回**

0：成功；

-1：失败。

# 5 数据结构设计

1. typedef struct {

EstabCause estab\_cause; /\*from RRC Connection Request \*/

ReestabCause reestab\_cause; /\* from RRC Connection Reestab Request \*/

NasMsg last\_nas\_msg; /\*store last rx nas msg from s1 or ue msg\*/

Gmmei reg\_mme; /\*from RRC Connection Setup Complete\*/

UINT32 rrc\_trans\_id;

UINT32 sel\_plmn\_id; /\*from RRC Connection Setup Complete\*/

}RrcEntity;

1. typedef struct {

NodeType ln;

RntiType rnti\_type;

RntiType pre\_rnti\_type;

UINT16 rnti\_value;

UINT16 pre\_rnti\_value;

/\* Security Params \*/

UESecurityCapabilities\_t ue\_security\_capabilities;

UeSecurityParams ue\_security\_params;

RadioBearInfo \*radio\_bear\_info[RB\_NUM];

void \*rrc\_entity\_p;

} UeRntiMsg;

1. typedef struct {

UINT32 k\_enb[8]; /\* 256 bit \*/

UINT32 k\_up\_enc[4]; /\* 128 bit \*/

UINT32 k\_cp\_enc[4]; /\* 128 bit \*/

UINT32 k\_cp\_int[4]; /\* 128 bit \*/

UINT32 nh[8]; /\* 256 bit \*/

UINT8 ncc; /\* maximum: 3bit \*/

UINT8 enc\_algorithm; /\* Encryption Algorithm \*/

UINT8 int\_algorithm; /\* Integrity Algorithm \*/

UINT8 enc\_active\_flag; /\* define whether the encryption function has been activated \*/

UINT8 int\_active\_flag; /\* define whether the integrity protection function has been activated \*/

} UeSecurityParams;

# 6 错误处理

无

# 7其他说明