**文件编号：LTE\_HeNB\_MAC\_ULSCHED\_XXSJ\_V1.0**

**TD-LTE HeNB协议栈软件系统**

**上行调度资源分配模块**

**详细设计说明书**

拟制：王丽萍

时间：2013-6-20

**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 | 王丽萍 | 2013-6-20 |  |  |
|  |  |  |  |  |  |

修改列表：

（1）。



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](#_Toc359599708)

[1 引言 5](#_Toc359599709)

[1.1 编写目的 5](#_Toc359599710)

[1.2 背景 5](#_Toc359599711)

[1.3 定义 5](#_Toc359599712)

[1.4 参考资料 5](#_Toc359599713)

[2 模块描述 6](#_Toc359599714)

[2.1 模块综述 6](#_Toc359599715)

[3 模块设计 7](#_Toc359599716)

[3.1 算法整体思路 7](#_Toc359599717)

[3.2 关键算法 7](#_Toc359599718)

[3.3 函数描述 8](#_Toc359599719)

[3.4 函数调用关系 10](#_Toc359599720)

[4 数据描述 11](#_Toc359599721)

[4.1数据结构说明 11](#_Toc359599722)

[4.1.1 调度器提供的上行调度资源分配过程中要使用的数据结构 11](#_Toc359599723)

[4.1.2 上行调度资源分配内部结构体 12](#_Toc359599724)

[4.1.3上行资源分配结果 13](#_Toc359599725)

[4.2全局变量（宏、常量）说明 13](#_Toc359599726)

[4.3数据库说明 14](#_Toc359599727)

[5 函数定义 15](#_Toc359599728)

[5.1引用函数描述 15](#_Toc359599729)

[5.2内部函数定义 15](#_Toc359599730)

[INT32 get\_mcs\_itbs\_by\_snr(INT8 snr, UINT8 \*mcs\_p, UINT8 \*itbs\_p) 15](#_Toc359599731)

[INT32 get\_ue\_ave\_snr(UINT16 rnti,UL\_schedule\_input\_param \*ul\_input, INT8 \*ave\_snr) 16](#_Toc359599732)

[UINT16 \*get\_ul\_reception\_data(UINT16 rnti, UL\_schedule\_input\_param \*ul\_param\_p) 16](#_Toc359599733)

[UINT16 accumulate\_service\_data(UINT16 rnti, UL\_schedule\_input\_param \*ul\_param\_p, UINT8 lc\_id) 17](#_Toc359599734)

[UINT32 pf\_function(PF\_node \*pf\_node\_p, UL\_schedule\_input\_param \*ul\_param\_p, UINT8 lc\_id) 19](#_Toc359599735)

[UINT32 transform\_data\_into\_nprb(UINT8 itbs,UINT32 data) 19](#_Toc359599736)

[INT32 rank\_orderings\_for\_ues(ListType \*pf\_mid\_list, PF\_node \*pf\_node\_p) 20](#_Toc359599737)

[INT32 handle\_service\_in\_lch(UL\_schedule\_input\_param \*ul\_input, ListType \*pf\_func\_result\_p) 21](#_Toc359599738)

[INT32 rbs\_pre\_allocate\_to\_service(ListType \*pf\_func\_result\_p) 23](#_Toc359599739)

[INT32 caculate\_services\_rbs\_for\_ue(ListType \*pf\_mid\_list\_p, UE\_PRB\_lcid \*ue\_prb\_lcid\_p) 24](#_Toc359599740)

[INT32 caculate\_bsr\_rbs\_for\_ue(UL\_schedule\_input\_param \*ul\_input, UE\_PRB\_BSR \*ue\_prb\_bsr\_p) 25](#_Toc359599741)

[INT32 decide\_ue\_rb\_length(UE\_PRB\_lcid \*ue\_prb\_lcid\_p, UE\_PRB\_BSR \*ue\_prb\_bsr\_p, UINT32 \*ue\_rb\_length\_result\_p) 26](#_Toc359599742)

[INT32 build\_ue\_rb\_matrix(UL\_schedule\_input\_param \*ul\_input, INT8 \*\*ue\_rb\_matrx\_p) 26](#_Toc359599743)

[INT32 find\_best\_ue\_on\_rbs\_segment(UINT8 \*ue\_alloc\_flag\_p, INT8 \*\*array\_p, UINT16 i\_s, UINT16 i\_e, UINT8 \*ue\_index) 28](#_Toc359599744)

[INT32 rbs\_final\_allocate\_to\_ue(UL\_schedule\_input\_param \*ul\_input, UINT32 \*ue\_rb\_length\_result\_p, ListType \*ul\_sched\_dci\_result\_p) 29](#_Toc359599745)

[6 接口设计 31](#_Toc359599746)

[6.1用户接口 31](#_Toc359599747)

[6.2硬件接口 31](#_Toc359599748)

[6.3软件接口 31](#_Toc359599749)

[INT32 ul\_schedule\_process(UL\_schedule\_input\_param \*ul\_input, ListType \*ul\_sched\_dci\_result\_p) 31](#_Toc359599750)

[7 功能测试设计 32](#_Toc359599751)

[8 其他 32](#_Toc359599752)

**详细设计说明书**

# 1 引言

## 1.1 编写目的

本说明书对LTE HeNB端MAC上行调度模块中资源分配子模块详细说明，阐述HeNB端MAC上行调度模块资源分配的设计思想以及其工作流程。本说明书的读者为LTE HeNB端MAC上行调度模块设计、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》
4. 物理层协议211、212、213
5. 调度器设计文档

# 

# 2 模块描述

## 2.1 模块综述

上行调度模块功能为为各个UE进行上行调度，调度结果通过PDCCH DCI0和PDSCH RAR PDU两种形式发放上行授权，指示其相对上行帧的上行发送资源。其中以PDSCH RAR PDU形式发送的上行调度结果由下行调度模块提供输出，以PDCCH DCI0形式发送的上行调度结果由发送模块通过调用上行调度模块的接口函数进行上行调度结果的输出。



## 3.2 关键算法

1.PF算法；

2.参考考虑信道质量及资源块连续性的RME算法思想

具体参数含义及详细思路参见：《TD-LTE\_HeNB协议栈软件系统上行调度资源分配算法调研及设计-调度器》

## 3.3 函数描述

|  |  |  |
| --- | --- | --- |
| **函数名** | **功能** | **函数类型** |
| ul\_schedule\_process () | 上行调度主执行函数资源分配整个过程 | 模块外部接口 |
| get\_mcs\_itbs\_by\_snr () | 根据SNR值得到MCS值和ITBS值 | 模块内部接口 |
| get\_ue\_ave\_snr () | 对UE在带宽内的所有SNR值取平均 |
| get\_ul\_reception\_data () | 接收每次上行调度时每个逻辑信道内的数据 |
| accumulate\_service\_data () | 统计在一个系统帧内各逻辑信道数据量的累积 |
| pf\_function () | 在一个逻辑信道内计算用户的PF优先级 |
| transform\_data\_into\_nprb () | 把数据量（bit）转换成相应所需要的RB数（协议213表格7.1） |
| rank\_orderings\_for\_ues () | 在逻辑信道内对用户进行排序 |
| handle\_service\_in\_lch () | 按业务优先级对逻辑信道进行处理（逻辑信道与业务是一一对应的） |
| rbs\_pre\_allocate\_to\_service() | 在带宽内对业务优先级逻辑信道内的UE进行资源预分配 |
| caculate\_services\_rbs\_for\_ue（） | 统计各个UE在各个逻辑信道内预分配的资源块总数 |
| caculate\_bsr\_rbs\_for\_ue（） | 统计各个UE的BSR数据量总结并转换成所需要的资源总数 |
| decide\_ue\_rb\_length（） | 根据统计的逻辑信道内总分配的资源块总数与BSR资源块总数，选择两者中小者作为为UE最终分配的资源块长度 |
| build\_ue\_rb\_matrix（） | 按每个UE的带宽内的所有snr值建立UE-RB矩阵 |
| find\_best\_ue\_on\_rbs\_segment（） | 在划分质量的的RB段内选择具有最好的信道 |
| rbs\_final\_allocate\_to\_ue（） | 根据UE的信道质量和UE的资源块长度决定UE的资源块起始值和长度、MCS并记录到DCI0中 |

## 3.4 函数调用关系

图2 函数调用关系图

# 4 数据描述

## 4.1数据结构说明

### 4.1.1 调度器提供数据结构





图、小区配置和UE配置信息图、上行输入参数中主要结构体

### 4.1.2 上行调度资源分配内部结构体

/\*处理逻辑信道中涉及的中间节点\*/

typedef struct{

NodeType ln;

UINT16 rnti;

INT8 snr; /\*present ave snr,used to get the ave mcs in PF alogrithm\*/

UINT16 mcs; /\*present ave mcs,used in PF alogrithm\*/

UINT8 itbs;

UINT32 past\_T\_time\_data;/\*the accumulated data received in WINDOWS\*/

UINT32 pf\_priority;

UINT32 lcid\_target\_data;/\*the target data in current logical channel\*/

UINT32 lcid\_n\_prb;/\*the RBs needed to send the target data in current logical channel\*/

}PF\_node;

对应的链表：ListType \*pf\_mid\_list；

/\*记录UE的逻辑信道资源预分配结果\*/

typedef struct{

UINT32 all\_n\_prb;/\* the total RBs pre-allocated to UE in all the logicals for ue\*/

UINT32 lcid\_n\_prb[MAX\_LC\_LIST];/\*the pre-allocated RBs in one logical for ue \*/

}UE\_PRB\_lcid;

/\*记录UE的BSR信息结构体\*/

typedef struct{

UINT8 bsr[MAX\_NR\_LCG];

UINT32 bsr\_bit\_data;

UINT32 bsr\_Nprb;

}UE\_PRB\_BSR;

### 4.1.3上行资源分配结果

/\*上行调度结果节点\*/

typedef struct {

NodeType ln;

UINT32 size; /\*调度数据大小\*/

UINT16 rb\_start; /\*RV起始位置\*/

UINT16 rb\_len; /\*RB个数\*/

UINT16 rnti; /\*RNTI值\*/

UINT8 mcs; /\*MCS值\*/

UINT8 ndi; /\*NDI值\*/

UINT8 rv; /\*RV值\*/

} UlSchedResultNode;

/\*上行调度结果链表\*/

typedef ListType UlSchedResultLst;

## 4.2全局变量（宏、常量）说明

* 宏定义

#define WINDOWS 10 /\*设置一个系统帧（10个子帧）为一个时间窗\*/

#define RNTIMIN 61 /\*RNTI的最小值\*/

#define MCSEXPEND 100 /\*MCS的扩展倍数防止计算优先级时分子MCS值（0～28）过小导致用户的优先级为0\*/

* 全局变量

UINT16 g\_window\_data[MAX\_UE][MAX\_LC\_LIST][WINDOWS];

/\*记录某用户在某子帧某逻辑信道接收的数据量\*/

* 常量

extern INT32 BSR\_BUFF\_SIZE[63]; //由BSR索引值得到相应的BSR的字节数

（321协议）

const INT32 ITBS\_NPRB[27][111]；//由itbs和data 可推出相应的RB数目

（213协议）

INT32 SNR\_MCS\_ITBS[28][3]； //由SNR值得出MCS和itbs的值

（213协议）及调研总结得下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNR（dB） | MCS Index | Modultaion Order | TBS Index | Efficiency |
| <(-5) | 0 | 2 (QPSK) | 0 | 0.20 |
| -5 | 1 | 2 (QPSK) | 1 | 0.25 |
| -4 | 2 | 2 (QPSK) | 2 | 0.31 |
| -3 | 3 | 2 (QPSK) | 3 | 0.41 |
| -2 | 4 | 2 (QPSK) | 4 | 0.50 |
| -1 | 5 | 2 (QPSK) | 5 | 0.62 |
| 0 | 6 | 2 (QPSK) | 6 | 0.73 |
| 1 | 7 | 2 (QPSK) | 7 | 0.86 |
| 2 | 8 | 2 (QPSK) | 8 | 0.98 |
| 3 | 9 | 2 (QPSK) | 9 | 1.11 |
| 4 | 10 | 2 (QPSK) | 10 | 1.23 |
| 5 | 11 | 2 (QPSK) | 10 | 1.23 |
| 6 | 12 | 4 (16QAM) | 11 | 1.41 |
| 7 | 13 | 4 (16QAM) | 12 | 1.60 |
| 8 | 14 | 4 (16QAM) | 13 | 1.80 |
| 9 | 15 | 4 (16QAM) | 14 | 2.01 |
| 10 | 16 | 4 (16QAM) | 15 | 2.14 |
| 11 | 17 | 4 (16QAM) | 16 | 2.28 |
| 12 | 18 | 4 (16QAM) | 17 | 2.52 |
| 13 | 19 | 4 (16QAM) | 18 | 2.77 |
| 14 | 20 | 4 (16QAM) | 19 | 3.01 |
| 15 | 21 | 6 (64QAM) | 19 | 3.01 |
| 16 | 22 | 6 (64QAM) | 20 | 3.25 |
| 17 | 23 | 6 (64QAM) | 21 | 3.51 |
| 18 | 24 | 6 (64QAM) | 22 | 3.77 |
| 19 | 25 | 6 (64QAM) | 23 | 4.02 |
| 20 | 26 | 6 (64QAM) | 24 | 4.26 |
| 21 | 27 | 6 (64QAM) | 25 | 4.45 |
| 22 | 28 | 6 (64QAM) | 26 | 4.63 |
| >22 | 28 | 6 (64QAM) | 26 | 4.63 |

## 4.3数据库说明

无

# 5 函数定义

## 5.1引用函数描述

|  |  |  |
| --- | --- | --- |
| 模块名 | 引用函数名 | 函数描述 |
| 调度器模块 | get\_cell\_configure\_info() | 获取上行带宽 |
| 调度器模块 | get\_scheduler\_ue\_entity\_info () | 获取UE配置 |

## 5.2内部函数定义

### INT32 get\_mcs\_itbs\_by\_snr(INT8 snr, UINT8 \*mcs\_p, UINT8 \*itbs\_p)

**函数描述**

根据SNR值得到MCS值和ITBS值

**输入**

snr //SNR具体值

**算法与处理流程**



**输出**

Mcs 和itbs的值

**返回**

0，操作成功；

-1, 系统错误

### INT32 get\_ue\_ave\_snr(UINT16 rnti,UL\_schedule\_input\_param \*ul\_input, INT8 \*ave\_snr)

**函数描述**

对UE在带宽内的所有SNR值取平均

**输入**

ul\_input //上行输入参数

**算法与处理流程**



**输出**

ave\_snr //全频带宽内用户的平均SNR值

**返回**

0，操作成功；

-1, 系统错误

### UINT16 \*get\_ul\_reception\_data(UINT16 rnti, UL\_schedule\_input\_param \*ul\_param\_p)

**函数描述**

接收每次上行调度时每个逻辑信道内的数据

**输入**

用户rnti

上行输入参数ul\_param\_p

**算法与处理流程**



**输出**

接收的数据

**返回**

0，操作成功；

-1, 系统错误

### UINT16 accumulate\_service\_data(UINT16 rnti, UL\_schedule\_input\_param \*ul\_param\_p, UINT8 lc\_id)

**函数描述**

统计在一个系统帧内各逻辑信道数据量的累积



**输入**

用户rnti

上行输入参数ul\_param\_p

逻辑信道号 lc\_id

**算法与处理流程**



**输出**

逻辑信道WINDOWS内累积数据

**返回**

0，操作成功；

-1, 系统错误

### UINT32 pf\_function(PF\_node \*pf\_node\_p, UL\_schedule\_input\_param \*ul\_param\_p, UINT8 lc\_id)

**函数描述**

在一个逻辑信道内计算用户的PF优先级

**输入**

用户中间节点pf\_node\_p

上行输入参数ul\_param\_p

逻辑信道号 lc\_id

**算法与处理流程**



**输出**

用户的优先级

**返回**

0，操作成功；

-1, 系统错误

### UINT32 transform\_data\_into\_nprb(UINT8 itbs,UINT32 data)

**函数描述**

把数据量（bit）转换成相应所需要的RB数（协议213表格7.1）

**输入**

TBS索引值itbs

数据量 data

**算法与处理流程**

输入TBS索引值itbs，数据量 data通过索引表格const INT32 ITBS\_NPRB[27][111]，

得到最接近的data对应的RB数

**输出**

N\_PRB

**返回**

0，操作成功；

-1, 系统错误

### INT32 rank\_orderings\_for\_ues(ListType \*pf\_mid\_list, PF\_node \*pf\_node\_p)

**函数描述**

在逻辑信道内对用户进行排序

**输入**

包含用户的逻辑信道链表pf\_mid\_list

链表中的用户节点pf\_node\_p

**算法与处理流程**



**输出**

把用户插入到合适的位置

**返回**

0，操作成功；

-1, 系统错误

### INT32 handle\_service\_in\_lch(UL\_schedule\_input\_param \*ul\_input, ListType \*pf\_func\_result\_p)

**函数描述**

按业务优先级对逻辑信道进行处理（逻辑信道与业务是一一对应的）

**输入**

上行输入参数ul\_input

**算法与处理流程**



**输出**

逻辑信道链表pf\_func\_result\_p (结构如下所示)



**返回**

0，操作成功；

-1, 系统错误

### INT32 rbs\_pre\_allocate\_to\_service(ListType \*pf\_func\_result\_p)

**函数描述**

在带宽内对业务优先级逻辑信道内的UE进行资源预分配

**输入**

逻辑信道处理的业务结果链表pf\_func\_result\_p

**算法与处理流程**



**输出**

**返回**

0，操作成功；

-1, 系统错误

### INT32 caculate\_services\_rbs\_for\_ue(ListType \*pf\_mid\_list\_p, UE\_PRB\_lcid \*ue\_prb\_lcid\_p)

**函数描述**

统计各个UE在各个逻辑信道内预分配的资源块总数

**输入**

**算法与处理流程**

**输出**

**返回**

0，操作成功；

-1, 系统错误

### INT32 caculate\_bsr\_rbs\_for\_ue(UL\_schedule\_input\_param \*ul\_input, UE\_PRB\_BSR \*ue\_prb\_bsr\_p)

**函数描述**

统计各个UE的BSR数据量总结并转换成所需要的资源总数

**输入**

上行输入参数ul\_input

**算法与处理流程**



**输出**

用户逻辑信道分配结果ue\_prb\_bsr\_p

**返回**

0，操作成功；

-1, 系统错误

### INT32 decide\_ue\_rb\_length(UE\_PRB\_lcid \*ue\_prb\_lcid\_p, UE\_PRB\_BSR \*ue\_prb\_bsr\_p, UINT32 \*ue\_rb\_length\_result\_p)

**函数描述**

根据统计的逻辑信道内总分配的资源块总数与BSR资源块总数，选择两者中小者作为为UE最终分配的资源块长度

**输入**

逻辑信道内总分配的资源块总数ue\_prb\_lcid\_p

BSR资源块总数ue\_prb\_bsr\_p,

**算法与处理流程**

在逻辑信道内总分配的资源块总数ue\_prb\_lcid\_p，BSR资源块总数ue\_prb\_bsr\_p间选择较小的作为UE的RB\_length.

**输出**

UE最终分配的资源块长度ue\_rb\_length\_result\_p

**返回**

0，操作成功；

-1, 系统错误

### INT32 build\_ue\_rb\_matrix(UL\_schedule\_input\_param \*ul\_input, INT8 \*\*ue\_rb\_matrx\_p)

**函数描述**

按每个UE的带宽内的所有snr值建立UE-RB矩阵



**输入**

上行输入参数ul\_input

**算法与处理流程**



**输出**

UE-RB矩阵ue\_rb\_matrx\_p

**返回**

0，操作成功；

-1, 系统错误

### INT32 find\_best\_ue\_on\_rbs\_segment(UINT8 \*ue\_alloc\_flag\_p, INT8 \*\*array\_p, UINT16 i\_s, UINT16 i\_e, UINT8 \*ue\_index)

**函数描述**

在划分质量的的RB段内选择具有最好的信道

**输入**

UE是否被分配资源标志ue\_alloc\_flag\_p

UE-RB矩阵array\_p

RB段起始值 i\_s

RB段组终值 i\_e

**算法与处理流程**



**输出**

具有最佳信道质量的用户ue\_index

**返回**

0，操作成功；

-1, 系统错误

### INT32 rbs\_final\_allocate\_to\_ue(UL\_schedule\_input\_param \*ul\_input, UINT32 \*ue\_rb\_length\_result\_p, ListType \*ul\_sched\_dci\_result\_p)

**函数描述**

根据UE的信道质量和UE的资源块长度决定UE的资源块起始值和长度、MCS并记录到DCI0中



**输入**

上行输入参数ul\_input

UE的最终分配资源块长度ue\_rb\_length\_result\_p

**算法与处理流程**



**输出**

生成的用户上行授权结果DCI 0 :ul\_sched\_dci\_result\_p

**返回**

0，操作成功；

-1, 系统错误

# 6 接口设计

## 6.1用户接口

无

## 6.2硬件接口

无

## 6.3软件接口

### INT32 ul\_schedule\_process(UL\_schedule\_input\_param \*ul\_input, ListType \*ul\_sched\_dci\_result\_p)

**功能描述：**

上行调度主执行函数资源分配整个过程

**输入**：

上行输入参数ul\_input

**算法与处理流程**



**输出**：

生成的用户上行授权结果DCI 0 :ul\_sched\_dci\_result\_p

**返回**：0/-1，操作是否成功。

# 7 功能测试设计

# 8 其他

说明：

目前只考虑了上行调度资源动态分配的过程，从资源分配的算法进行详细设计，没有考虑与其它模块间的交互关系。下一步工作应是与其它模块进行整合。