**文件编号：LTE\_HeNB\_MAC\_RX\_DYCSBG\_V1.0**

**TD-LTE HeNB协议栈软件系统**

**MAC接收模块**

**单元测试报告**

拟制：李亚楠

时间：2011-01-13

**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 | 2011.1.13 | 无 | 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。

[1 引言 1](#_Toc291493823)

[1.1 背景 1](#_Toc291493824)

[1.2 定义 1](#_Toc291493825)

[1.3 参考资料 1](#_Toc291493826)

[2 测试模块 1](#_Toc291493827)

[2.1 模块名 1](#_Toc291493828)

[2.2 文件名 1](#_Toc291493829)

[2.3 测试人 2](#_Toc291493830)

[2.4 测试用例列表 2](#_Toc291493831)

[3 函数测试 3](#_Toc291493832)

[3.1 被测函数：unpack\_subhdr（） 3](#_Toc291493833)

[3.1.1 函数描述： 3](#_Toc291493834)

[3.1.2 测试用例： 4](#_Toc291493835)

[3.2 被测函数：unpack\_payload（） 18](#_Toc291493836)

[3.2.1 函数描述： 18](#_Toc291493837)

[3.2.2 测试用例： 19](#_Toc291493838)

[3.3 被测函数：unpack\_mac\_pdu（） 27](#_Toc291493839)

[3.3.1 函数描述： 27](#_Toc291493840)

[3.3.2 测试用例： 28](#_Toc291493841)

[4 功能单元测试 38](#_Toc291493842)

[5 功能组合测试 39](#_Toc291493843)

[5.1 MAC接收功能测试 39](#_Toc291493844)

[5.1.1 测试说明： 39](#_Toc291493845)

[5.1.2 所涉及功能 39](#_Toc291493846)

[5.1.3 测试用例： 39](#_Toc291493847)

[6 其他测试 48](#_Toc291493848)

[附录1 测试类型 49](#_Toc291493849)

[1. 等价类划分 49](#_Toc291493850)

[2. 边值分析法 50](#_Toc291493851)

[3. 语句覆盖 51](#_Toc291493852)

[4. 判定覆盖 51](#_Toc291493853)

[5. 条件覆盖 52](#_Toc291493854)

[6. 判定／条件覆盖 52](#_Toc291493855)

[7. 条件组合覆盖 53](#_Toc291493856)

# 1 引言

## 1.1 背景

1. 软件系统：TD-LTE HeNB协议栈软件系统
2. 开发环境：
3. 开发工具： Vim-7.0
4. 编程语言： C语言
5. 支持库： GLIBC
6. 操作系统：Linux
7. 测试环境：
8. 运行平台： PC-Linux
9. CPU类型：AMD Athlon（tm）Processor，2.71G
10. 内存大小：960M

## 1.2 定义

PDU: protocol data unit。

SDU: service data unit。

CE: control element。

## 1.3 参考资料

1. 《LTE-HeNB协议栈软件系统MAC接收模块概要设计说明书\_v1.0》；
2. 《LTE-HeNB协议栈软件系统MAC接收模块详细设计说明书\_v1.0》。

# 2 测试模块

## 2.1 模块名

MAC\_RX模块。

## 2.2 文件名

mac\_pdu\_rx.c

mac\_pdu\_rx.h

## 2.3 测试人

李亚楠

[liyanan01@ict.ac.cn](mailto:liyanan01@ict.ac.cn)。

## 2.4 测试用例列表

|  |  |
| --- | --- |
| **测试用例** | **测试说明** |
| HeNB\_MAC\_RX\_1 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_2 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_3 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_4 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_5 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_6 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_7 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_8 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_9 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_10 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_11 | unpack\_subhdr（） |
| HeNB\_MAC\_RX\_12 | unpack\_payload（） |
| HeNB\_MAC\_RX\_13 | unpack\_payload（） |
| HeNB\_MAC\_RX\_14 | unpack\_payload（） |
| HeNB\_MAC\_RX\_15 | unpack\_payload（） |
| HeNB\_MAC\_RX\_16 | unpack\_payload（） |
| HeNB\_MAC\_RX\_17 | unpack\_payload（） |
| HeNB\_MAC\_RX\_18 | unpack\_payload（） |
| HeNB\_MAC\_RX\_19 | unpack\_payload（） |
| HeNB\_MAC\_RX\_20 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_21 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_22 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_23 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_24 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_25 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_26 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_27 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_28 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_29 | unpack\_mac\_pdu（） |
| HeNB\_MAC\_RX\_30 | 功能测试 |
| HeNB\_MAC\_RX\_31 | 功能测试 |
| HeNB\_MAC\_RX\_32 | 功能测试 |
| HeNB\_MAC\_RX\_33 | 功能测试 |
| HeNB\_MAC\_RX\_34 | 功能测试 |
| HeNB\_MAC\_RX\_35 | 功能测试 |
| HeNB\_MAC\_RX\_36 | 功能测试 |
| HeNB\_MAC\_RX\_37 | 功能测试 |

# 3 函数测试

|  |  |
| --- | --- |
| **被测函数** | **函数说明** |
| unpack\_subhdr（） | 解析MAC子头，生成子头信息链表； |
| unpack\_payload（） | 解析MAC净荷，把CE和SDU递交到相关模块； |
| unpack\_mac\_pdu（） | 解析MAC PDU； |

## 3.1 被测函数：unpack\_subhdr（）

### 3.1.1 函数描述：

**功能**

解析MAC子头，生成子头信息链表。

**输入**

UINT8 \*\*pdu \_p：指向MAC PDU起始位置的双指针；

MacSubhdrLst \*mac\_subhdr\_lst\_p；子头信息链表；

INT16 \*total\_size\_p；MAC PDU的总长度地址的指针；

**输出**

UINT8 \*\*pdu \_p：指向MAC PDU净荷起始位置的双指针；

MacSubhdrLst \*mac\_subhdr\_lst\_p；生成节点的子头信息链表；

INT16 \*total\_size\_p；MAC PDU净荷的总长度的指针；

**算法与处理流程**

检查输入参数；

while（还有子头） {

获取是否还有子头的标志位e；

获取lcid；

if （lcid不合法） {

返回；

}

为节点分配空间；

在节点中存入lcid信息；

switch （lcid） {

case ce：

记录长度信息；

计算MAC PDU剩余长度

case sdu：

if （最后一个SDU子头） {

将长度记录为0字节；

计算剩余MAC PDU长度；

} else {

解析2字节或3字节子头；

计算剩余MAC PDU长度

}

}

}

将节点加入链表；

}

### 3.1.2 测试用例：

|  |  |  |
| --- | --- | --- |
| **测试用例** | **测试类型** | **测试说明** |
| HeNB\_MAC\_RX\_1 | 等价类划分 | 输入参数错误情况 |
| HENB\_MAC\_RX\_2 | 等价类划分 | MAC PDU长度错误 |
| HENB\_MAC\_RX\_3 | 语句覆盖 | 只有CE，没有padding，其中BSR CE为截短BSR CE。 |
| HENB\_MAC\_RX\_4 | 语句覆盖 | 只有CE，没有padding，其中BSR CE为短BSR CE。 |
| HENB\_MAC\_RX\_5 | 语句覆盖 | 只有CE，没有padding，其中BSR CE为长BSR CE。 |
| HENB\_MAC\_RX\_6 | 语句覆盖 | 只有CE，插入padding子头； |
| HENB\_MAC\_RX\_7 | 语句覆盖 | 只有CE，普通padding； |
| HENB\_MAC\_RX\_8 | 语句覆盖 | 只有SDU，子头长度覆盖2字节和3字节，没有padding |
| HENB\_MAC\_RX\_9 | 语句覆盖 | 只有SDU，子头长度覆盖2字节和3字节，插入子头为1字节 |
| HENB\_MAC\_RX\_10 | 语句覆盖 | 只有SDU，子头长度覆盖2字节和3字节，插入子头为2字节 |
| HENB\_MAC\_RX\_11 | 语句覆盖 | 只有SDU，子头长度覆盖2字节和3字节，普通padding |

#### 测试用例：HENB\_MAC\_RX\_1

|  |  |
| --- | --- |
| **说明** | 输入参数错误情况 |
| **输入** | pdu\_p= NULL；  mac\_subhdr\_lst\_p = NULL；  total\_size = 0x11； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

声明空指针；

调用函数unpack\_subhdr（）。

**测试方法**

输入参数为空，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:1

MAC\_RX: unpack\_subhdr: mac\_subhdr\_lst\_p is null

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_2

|  |  |
| --- | --- |
| **说明** | MAC PDU长度错误 |
| **输入** | msg\_2[7] = {0x3A, 0x3B, 0x1C,5,6,6,7}  初始化的mac\_subhdr\_lst\_p；  total\_size = 1； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_2；

total\_size = 1；

unpack\_subhdr（）;

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:2

MAC\_RX: total\_size error!

the MAC PDU payload len is: -1

the content is:

the lcid of subheader is: 1a

length that subheader corresponds to is: 1

the lcid of subheader is: 1b

length that subheader corresponds to is: 2

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_3

|  |  |
| --- | --- |
| **说明** | 只有CE，没有padding，其中BSR CE为截短BSR CE。 |
| **输入** | msg\_3[7] = {0x3A, 0x3B, 0x1C,5,6,6,7}  初始化的mac\_subhdr\_lst\_p；  total\_size =7； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_3；

total\_size = 7；

show\_memory();

unpack\_subhdr（）;

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:3

the MAC PDU len is: 7

the content is: 3a 3b 1c 5 6 6 7

the MAC PDU payload len is: 4

the content is: 5 6 6 7

the lcid of subheader is: 1a

length that subheader corresponds to is: 1

the lcid of subheader is: 1b

length that subheader corresponds to is: 2

the lcid of subheader is: 1c

length that subheader corresponds to is: 1

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_4

|  |  |
| --- | --- |
| **说明** | 只有CE，没有padding，其中BSR CE为短BSR CE。 |
| **输入** | msg\_4[7] = {0x3A, 0x3B, 0x1D,5,6,6,8}  初始化的mac\_subhdr\_lst\_p；  total\_size = 7； |
| **输出** | 打印信息，并返回 |

**测试修改**

注释原函数中cleanup\_mac\_rx（）中的删除线程的代码。

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_4；

total\_size = 7；

show\_memory();

unpack\_subhdr（）;

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:4

the MAC PDU len is: 7

the content is: 3a 3b 1d 5 6 6 8

the MAC PDU payload len is: 4

the content is: 5 6 6 8

the lcid of subheader is: 1a

length that subheader corresponds to is: 1

the lcid of subheader is: 1b

length that subheader corresponds to is: 2

the lcid of subheader is: 1d

length that subheader corresponds to is: 1

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_5

|  |  |
| --- | --- |
| **说明** | 只有CE，没有padding，其中BSR CE为长BSR CE。 |
| **输入** | msg\_5[9] = {0x3A, 0x3B, 0x1E,5,6,6,4,4,4}  初始化的mac\_subhdr\_lst\_p；  total\_size = 9； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_5；

total\_size = 9；

show\_memory();

unpack\_subhdr（）;

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:5

the MAC PDU len is: 9

the content is: 3a 3b 1e 5 6 6 4 4 4

the MAC PDU payload len is: 6

the content is: 5 6 6 4 4 4

the lcid of subheader is: 1a

length that subheader corresponds to is: 1

the lcid of subheader is: 1b

length that subheader corresponds to is: 2

the lcid of subheader is: 1e

length that subheader corresponds to is: 3

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_6

|  |  |
| --- | --- |
| **说明** | 只有CE，插入padding子头； |
| **输入** | msg\_6[9] = {0x3F, 0x3A, 0x3B, 0x1C, 5,6,6,7}  初始化的mac\_subhdr\_lst\_p；  total\_size\_p = 8； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_6；

total\_size = 8；

show\_memory（）；

unpack\_subhdr（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:6

the MAC PDU len is: 8

the content is: 3f 3a 3b 1c 5 6 6 7

the MAC PDU payload len is: 4

the content is: 5 6 6 7

the lcid of subheader is: 1a

length that subheader corresponds to is: 1

the lcid of subheader is: 1b

length that subheader corresponds to is: 2

the lcid of subheader is: 1c

length that subheader corresponds to is: 1

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_7

|  |  |
| --- | --- |
| **说明** | 只有CE，普通padding； |
| **输入** | msg\_7[11]={0x3A, 0x3B, 0x3C, 0x1F,5,6,6,7,0,0,0}  初始化的mac\_subhdr\_lst\_p；  total\_size=11 ； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg；

total\_size = 11；

show\_memory（）;

unpack\_subhdr（）;

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:7

the MAC PDU len is: 11

the content is: 3a 3b 3c 1f 5 6 6 7 0 0 0

the MAC PDU payload len is: 7

the content is: 5 6 6 7 0 0 0

the lcid of subheader is: 1a

length that subheader corresponds to is: 1

the lcid of subheader is: 1b

length that subheader corresponds to is: 2

the lcid of subheader is: 1c

length that subheader corresponds to is: 1

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_8

|  |  |
| --- | --- |
| **说明** | 只有SDU，子头长度覆盖2字节和3字节，没有padding |
| **输入** | msg\_8[294] = {0x21，0x0A, 0x22, 0x14, 0x23, 0x80, 0x80, 0x04}  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 128；  第一个lcid = 4，len = 127；  初始化的mac\_subhdr\_lst\_p；  total\_size = 294：8字节子头，286字节净荷； |
| **输出** | 打印信息，并返回 |

**测试修改**

注释原函数中cleanup\_mac\_rx（）中的删除线程的代码。

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_8；

show\_memory（）；

total\_size = 294；

unpack\_subhdr（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:8

the MAC PDU len is: 294

the content is: 21 a 22 14 23 80 80 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0

the MAC PDU payload len is: 286

the content is: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the lcid of subheader is: 1

length that subheader corresponds to is: 10

the lcid of subheader is: 2

length that subheader corresponds to is: 20

the lcid of subheader is: 3

length that subheader corresponds to is: 128

the lcid of subheader is: 4

length that subheader corresponds to is: 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_9

|  |  |
| --- | --- |
| **说明** | 只有SDU，子头长度覆盖2字节和3字节，插入子头为1字节 |
| **输入** | msg\_9[294] = {0x3F,0x21，0x0A, 0x22, 0x14, 0x23, 0x80, 0x80, 0x04，  0，0，0，0，0，0, 0，0，0，0}  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 128；  第一个lcid = 4，len = 127；  初始化的mac\_subhdr\_lst\_p；  total\_size= 294（9字节MAC子头，280字节净荷）； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg；

show\_memory（）；

total\_size = 294；

unpack\_subhdr（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:9

the MAC PDU len is: 294

the content is: 3f 21 a 22 14 23 80 80 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0

the MAC PDU payload len is: 280

the content is: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the lcid of subheader is: 1

length that subheader corresponds to is: 10

the lcid of subheader is: 2

length that subheader corresponds to is: 20

the lcid of subheader is: 3

length that subheader corresponds to is: 128

the lcid of subheader is: 4

length that subheader corresponds to is: 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_10

|  |  |
| --- | --- |
| **说明** | 只有SDU，子头长度覆盖2字节和3字节，插入子头为2字节 |
| **输入** | msg\_10[294] = {0x3F, 0x3F,0x21，0x0A, 0x22, 0x14, 0x23，0x7F, 0x04,  0，0，0，0，0，0, 0，0，0，0}  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 127；  第一个lcid = 4，len = 128；  初始化的mac\_subhdr\_lst\_p；  total\_size = 294（9字节MAC子头，280字节净荷）； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_10；

show\_memory（）；

total\_size = 294；

unpack\_subhdr（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:10

the MAC PDU len is: 294

the content is: 3f 3f 21 a 22 14 23 7f 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0

the MAC PDU payload len is: 280

the content is: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the lcid of subheader is: 1

length that subheader corresponds to is: 10

the lcid of subheader is: 2

length that subheader corresponds to is: 20

the lcid of subheader is: 3

length that subheader corresponds to is: 127

the lcid of subheader is: 4

length that subheader corresponds to is: 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_11

|  |  |
| --- | --- |
| **说明** | 只有SDU，子头长度覆盖2字节和3字节，普通padding |
| **输入** | msg\_11[290] = {0x21，0x0A, 0x22, 0x14, 0x23，0x7F, 0x24, 0x80，0x80，  0x1F, 0，0，0，0，0，0, 0，0，0，0}  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 127；  第一个lcid = 4，len = 128；  初始化的mac\_subhdr\_lst\_p；  total\_size = 295（10字节MAC子头，285字节净荷）； |
| **输出** | 打印信息，并返回 |

**测试修改**

注释原函数中cleanup\_mac\_rx（）中的删除线程的代码。

**测试程序**

create\_subhdr\_lst（）；

pdu\_p = msg\_11；

show\_memory（）；

total\_size = 295；

unpack\_subhdr（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:11

the MAC PDU len is: 295

the content is: 21 a 22 14 23 7f 24 80 80 1f 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0

the MAC PDU payload len is: 285

the content is: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the lcid of subheader is: 1

length that subheader corresponds to is: 10

the lcid of subheader is: 2

length that subheader corresponds to is: 20

the lcid of subheader is: 3

length that subheader corresponds to is: 127

the lcid of subheader is: 4

length that subheader corresponds to is: 128

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

## 3.2 被测函数：unpack\_payload（）

### 3.2.1 函数描述：

**功能**

解析MAC PDU净荷，将CE交给相应控制模块，将SDU交给RLC接收模块。

**输入**

UINT16 rnti：RNTI值；

UINT8 \*payload \_p：指向MAC PDU净荷起始位置的指针；

MacSubhdrLst \*mac\_subhdr\_lst\_p：子头信息链表；

INT16 total\_size：MAC PDU净荷的总长度；

**输出**

无。

**算法与处理流程**

检查输入参数；

获取子头信息节点；

while（节点） {

检查lcid是否合法；

switch （lcid） {

case 0：

......

default:

if (len == 0) {

将剩余长度设置为该SDU长度；

}

}

计算MAC PDU剩余长度；

获取下一个子头节点；

### 3.2.2 测试用例：

|  |  |  |
| --- | --- | --- |
| **测试用例** | **测试类型** | **测试说明** |
| HENB\_MAC\_RX\_12 | 等价类划分 | 输入空指针 |
| HENB\_MAC\_RX\_13 | 等价类划分 | 净荷长度小于0 |
| HENB\_MAC\_RX\_14 | 语句覆盖 | CE长度大于MAC PDU剩余长度 |
| HENB\_MAC\_RX\_15 | 语句覆盖 | 解析CE |
| HENB\_MAC\_RX\_16 | 语句覆盖 | 某个SDU长度大于剩余总长度 |
| HENB\_MAC\_RX\_17 | 语句覆盖 | SDU子头都有长度域 |
| HENB\_MAC\_RX\_18 | 语句覆盖 | 最后一个SDU子头没有长度域 |
| HENB\_MAC\_RX\_19 | 语句覆盖 | SDU长度为0，但不是说最后一个节点 |

#### 测试用例：HENB\_MAC\_RX\_12

|  |  |
| --- | --- |
| **说明** | 输入参数错误情况 |
| **输入** | rnti=0xFFFF；  payload\_p= NULL；  mac\_subhdr\_lst\_p = NULL；  total\_size = 50； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

声明空指针；

调用函数unpack\_payload（）。

**测试方法**

输入参数为空，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:12

MAC\_RX: unpack\_mac\_payload: mac\_subhdr\_lst\_p is null

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_13

|  |  |
| --- | --- |
| **说明** | MAC PDU长度为小于或等于0 |
| **输入** | payload\_p；  mac\_subhdr\_lst\_p，有三个节点：  第一个节点：lcid = 0X1A, len = 1；  第一个节点：lcid = 0X1B, len = 2；  第一个节点：lcid = 0X1C, len = 1；  total\_size = 0； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

add\_subhdr\_node（mac\_subhdr\_lst\_p，test\_case）；

switch （test\_case） {

case 13;

total\_size = 0；

payload\_p = ce\_msg；

break；

}

unpack\_payload（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:13

MAC\_RX: unpack\_mac\_payload: total\_size is 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_14

|  |  |
| --- | --- |
| **说明** | CE长度大于MAC PDU剩余长度 |
| **输入** | payload\_p；  mac\_subhdr\_lst\_p，有三个节点：  第一个节点：lcid = 0X1A, len = 1；  第一个节点：lcid = 0X1B, len = 2；  第一个节点：lcid = 0X1C, len = 1；  total\_size =2；第一个CE净荷长度为1，因此在解析第一个CE时返回 |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

add\_subhdr\_node（mac\_subhdr\_lst\_p，test\_case）；

switch （test\_case） {

case 14;

total\_size = 2；

payload\_p = ce\_msg；

break；

}

unpack\_payload（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:14

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

MAC\_RX: unpack\_mac\_payload :total\_size < ce\_len

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_15

|  |  |
| --- | --- |
| **说明** | 解析CE |
| **输入** | payload\_p；  mac\_subhdr\_lst\_p，有三个节点：  第一个节点：lcid = 0X1A, len = 1；  第一个节点：lcid = 0X1B, len = 2；  第一个节点：lcid = 0X1C, len = 1；  total\_size = 4；MAC PDU的净荷占4字节 |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

add\_subhdr\_node（mac\_subhdr\_lst\_p，test\_case）；

switch （test\_case） {

case 15;

total\_size = 4；

payload\_p = ce\_msg；

break；

}

unpack\_payload（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:15

ce len is: 4

the content is: 5 6 6 7

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_16

|  |  |
| --- | --- |
| **说明** | 某个SDU长度大于剩余总长度 |
| **输入** | payload\_p；  mac\_subhdr\_lst\_p，有三个节点：  第一个lcid = 1，len = 5；  第一个lcid = 2，len = 10；  第一个lcid = 3，len = 15；  第一个lcid = 4，len = 20；  total\_size = 7；MAC PDU的净荷占50字节 |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

add\_subhdr\_node（mac\_subhdr\_lst\_p，test\_case）；

switch （test\_case） {

case 16;

total\_size = 7；

payload\_p = sdu\_msg；

break；

}

unpack\_payload（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:16

sdu len is: 50

the content is: 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7

7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:5 5 5 5 5

MAC\_RX:unpack\_payload:total\_size<sdu\_len

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_17

|  |  |
| --- | --- |
| **说明** | SDU子头都有长度域 |
| **输入** | payload\_p；  mac\_subhdr\_lst\_p，有三个节点：  第一个lcid = 1，len = 5；  第一个lcid = 2，len = 10；  第一个lcid = 3，len = 15；  第一个lcid = 4，len = 20；  total\_size = 50；MAC PDU的净荷占50字节 |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

add\_subhdr\_node（mac\_subhdr\_lst\_p，test\_case）；

switch （test\_case） {

case 17;

total\_size = 50；

payload\_p = sdu\_msg；

break；

}

unpack\_payload（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:17

sdu len is: 50

the content is: 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7

7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:5 5 5 5 5

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:6 6 6 6 6 6 6 6 6 6

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:7 7 7 7 7 7 7 7 7 7 7 7 7 7 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_18

|  |  |
| --- | --- |
| **说明** | 最后一个SDU子头没有长度域 |
| **输入** | payload\_p；  mac\_subhdr\_lst\_p，有三个节点：  第一个lcid = 1，len = 5；  第一个lcid = 2，len = 10；  第一个lcid = 3，len = 15；  第一个lcid = 4，len = 0；  total\_size = 50；MAC PDU的净荷占50字节 |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_subhdr\_lst（）；

add\_subhdr\_node（mac\_subhdr\_lst\_p，test\_case）；

switch （test\_case） {

case 18;

total\_size = 50；

payload\_p = sdu\_msg；

break；

}

unpack\_payload（）；

**测试方法**

输入参数为空，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:18

sdu len is: 50

the content is: 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7

7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is: 5 5 5 5 5

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:6 6 6 6 6 6 6 6 6 6

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:7 7 7 7 7 7 7 7 7 7 7 7 7 7 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_19

|  |  |
| --- | --- |
| **说明** | SDU长度为0，但不是说最后一个节点 |
| **输入** | payload\_p；  mac\_subhdr\_lst\_p，有三个节点：  第一个lcid = 1，len =5；  第一个lcid = 2，len = 10；  第一个lcid = 3，len = 0；  第一个lcid = 4，len = 20；  total\_size = 50；MAC PDU的净荷占50字节 |
| **输出** | 打印信息，并返回 |

**测试修改**

注释原函数中cleanup\_mac\_rx（）中的删除线程的代码。

**测试程序**

create\_subhdr\_lst（）；

add\_subhdr\_node（mac\_subhdr\_lst\_p，test\_case）；

switch （test\_case） {

case 19;

total\_size = 50；

payload\_p = sdu\_msg；

break；

}

unpack\_payload（）；

**测试方法**

设置输入参数，调用被测函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:19

sdu len is: 50

the content is: 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7

7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is: 5 5 5 5 5

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:6 6 6 6 6 6 6 6 6 6

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

MAC\_RX:unpack\_payload:total\_size<sdu\_len

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

## 3.3 被测函数：unpack\_mac\_pdu（）

### 3.3.1 函数描述：

**功能**

解析整个MAC PDU。

**输入**

UINT16 rnti：该PDU对应的UE

UINT8 \*msg\_p：指向PDU消息的指针；

UINT32 msg\_len：该PDU消息的长度。

**输出**

无。

**算法与处理流程**

初始化子头信息链表mac\_subhdr\_lst\_p；

unpack\_subhdr（）；

if （total\_size < 0） {

释放链表；

报告错误并返回；

}

unpack\_payload（）；

释放mac\_subhdr\_lst\_p；

### 3.3.2 测试用例：

|  |  |  |
| --- | --- | --- |
| **测试用例** | **测试类型** | **测试说明** |
| HENB\_MAC\_RX\_20 | 等价类划分 | 输入参数错误； |
| HENB\_MAC\_RX\_21 | 等价类划分 | 解析完子头后，剩余长度小于0； |
| HENB\_MAC\_RX\_22 | 语句覆盖 | 只有CE，无padding的MAC PDU解析 |
| HENB\_MAC\_RX\_23 | 语句覆盖 | 只有CE，有插入padding子头的MAC PDU解析 |
| HENB\_MAC\_RX\_24 | 语句覆盖 | 只有CE，有普通padding的MAC PDU解析 |
| HENB\_MAC\_RX\_25 | 语句覆盖 | 有CE和SDU，无padding的MAC PDU解析 |
| HENB\_MAC\_RX\_26 | 语句覆盖 | 有CE和SDU，有1字节插入padding子头的MAC PDU解析 |
| HENB\_MAC\_RX\_27 | 语句覆盖 | 有CE和SDU，有2字节插入padding子头的MAC PDU解析 |
| HENB\_MAC\_RX\_28 | 语句覆盖 | 有CE和SDU，有普通padding的MAC PDU解析 |
| HENB\_MAC\_RX\_29 | 等价类划分 | lcid号非法情况 |

#### 测试用例：HENB\_MAC\_RX\_20

|  |  |
| --- | --- |
| **说明** | 输入参数错误情况 |
| **输入** | rnti=0xFFFF；  msg\_p = NULL；  msg\_len = 0； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_p = NULL；

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:20

MAC\_RX: unpack\_mac\_pdu :msg\_p is null

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_21

|  |  |
| --- | --- |
| **说明** | 解析完子头后，剩余长度小于等于0； |
| **输入** | rnti=0xFFFF；  pdu\_21[7] = {0x3A, 0x3B, 0x1C,5,6,6,7}；  msg\_len = 3； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_p = pdu\_21;

msg\_len = 3;

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:21

MAC\_RX: total size < 0!

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_22

|  |  |
| --- | --- |
| **说明** | 只有CE，无padding的MAC PDU解析 |
| **输入** | rnti =0xFFFF；  pdu\_22[7] = {0x3A, 0x3B, 0x1C,5,6,6,7}  msg\_len = 7； |
| **输出** | 打印信息，并返回 |

**测试修改**

注释原函数中cleanup\_mac\_rx（）中的删除线程的代码。

**测试程序**

msg\_p = pdu\_22;

msg\_len = 3;

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:22

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_23

|  |  |
| --- | --- |
| **说明** | 只有CE，插入padding子头的MAC PDU解析 |
| **输入** | rnti =0xFFFF；  pdu\_23[8] = {0x3F，0x3A, 0x3B, 0x1C,5,6,6,7}  msg\_len = 8； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_p = pdu\_23;

msg\_len = 8;

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:23

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_24

|  |  |
| --- | --- |
| **说明** | 只有CE，普通padding的MAC PDU解析 |
| **输入** | rnti =0xFFFF；  pdu\_24[11] = {0x3A, 0x3B, 0x1C, 0x1F，5,6,6,7,0,0,0}  msg\_len = 11； |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_p = pdu\_24;

msg\_len = 11;

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:24

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_25

|  |  |
| --- | --- |
| **说明** | 有CE和SDU，无padding的MAC PDU解析 |
| **输入** | rnti =0xFFFF；  pdu\_25 [10] = {0x3A, 0x3B, 0x3C, 0x21，0x0A, 0x22, 0x14, 0x23, 0x80, 0x80, 0x04，5，6，6，7}  第一CE:PHR；  第二个CE：C-RNTI；  第三个CE：Truncated BSR；  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 128；  第一个lcid = 4，len = 127；  msg\_len = 303（MAC子头：11字节，净荷：292） |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_len = 303;

msg\_p = pdu\_25；

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:25

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_26

|  |  |
| --- | --- |
| **说明** | 有CE和SDU，有1字节插入padding子头的MAC PDU解析 |
| **输入** | rnti =0xFFFF；  pdu\_26[16] = {0x3F,0x3A, 0x3B, 0x3C, 0x21，0x0A, 0x22, 0x14, 0x23, 0x80, 0x80, 0x04，5，6，6，7 }  第一CE:PHR；  第二个CE：C-RNTI；  第三个CE：Truncated BSR；  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 128；  第一个lcid = 4，len = 127；  msg\_len = 304（MAC子头：12字节，净荷：292） |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_len = 304;

msg\_p = pdu\_26；

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:26

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_27

|  |  |
| --- | --- |
| **说明** | 有CE和SDU，有2字节插入padding子头的MAC PDU解析 |
| **输入** | rnti =0xFFFF；  pdu\_27[16] = {0x3F, 0x3F,0x3A, 0x3B, 0x3C, 0x21，0x0A, 0x22, 0x14, 0x23，0x7F, 0x04，5，6，6，7}  第一CE: PHR；  第二个CE：C-RNTI；  第三个CE：Truncated BSR；  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 127；  第一个lcid = 4，len = 128；  msg\_len = 304（MAC子头：12字节，净荷：292） |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_len = 304;

msg\_p = pdu\_27

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:27

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_28

|  |  |
| --- | --- |
| **说明** | 有CE和SDU，有普通padding的MAC PDU解析 |
| **输入** | rnti =0xFFFF；  pdu\_28[17] = {0x3A, 0x3B, 0x3C, 0x21，0x0A, 0x22, 0x14, 0x23，0x7F, 0x24, 0x80，0x80，0x1F，5，6，6，7}  第一CE: PHR；  第二个CE：C-RNTI；  第三个CE：Truncated BSR；  第一个lcid = 1，len = 10；  第一个lcid = 2，len = 20；  第一个lcid = 3，len = 127；  第一个lcid = 4，len = 128；  msg\_len = 305（MAC子头：13字节，净荷：292） |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_len = 305;

msg\_p = (UINT8 \*)malloc(305);

memcpy(msg\_p, pdu\_28, 17);

memset(msg\_p + 17, 0, 288);

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:28

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_29

|  |  |
| --- | --- |
| **说明** | lcid号非法情况 |
| **输入** | rnti =0xFFFF；  pdu\_29[7] = {0x2B, 0x3B, 0，0，0，0，0}  msg\_len = 7（MAC子头：2字节，净荷：5） |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

msg\_p = pdu\_29;

msg\_len = 7;

unpack\_mac\_pdu（）；

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:29

MAC\_RX: reserved lcid

MAC\_RX: unpack\_subhdr error!

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

# 4 功能单元测试

本模块主要包括三个功能单元：解析子头功能、解析净荷功能和MAC PDU的解包功能。

这三个功能单元的测试已在各函数测试中完成。

# 5 功能组合测试

## 5.1 MAC接收功能测试

### 5.1.1 测试说明：

在创建的进程中，收取消息队列中的消息，并解析。

### 5.1.2 所涉及功能

ULSCH\_TB子头解析功能；

MAC PDU的解析功能。

### 5.1.3 测试用例：

|  |  |  |
| --- | --- | --- |
| **测试用例** | **测试类型** | **测试说明** |
| HENB\_MAC\_RX\_30 | 语句覆盖 | 接收只有ce，无padding的MAC PDU消息； |
| HENB\_MAC\_RX\_31 | 语句覆盖 | 接收只有CE，有插入padding子头的MAC PDU； |
| HENB\_MAC\_RX\_32 | 语句覆盖 | 接收只有CE，有普通padding的MAC PDU； |
| HENB\_MAC\_RX\_33 | 语句覆盖 | 接收有CE和SDU，无padding的MAC PDU |
| HENB\_MAC\_RX\_34 | 语句覆盖 | 接收有CE和SDU，有1字节插入padding子头的MAC PDU； |
| HENB\_MAC\_RX\_35 | 语句覆盖 | 接收有CE和SDU，有2字节插入padding子头的MAC PDU； |
| HENB\_MAC\_RX\_36 | 语句覆盖 | 接收有CE和SDU，有普通padding子头的MAC PDU； |
| HENB\_MAC\_RX\_37 | 语句覆盖 | 错误信息的接收。 |

#### 测试用例：HENB\_MAC\_RX\_30

|  |  |
| --- | --- |
| **说明** | 接收只有ce，无padding的MAC PDU消息； |
| **输入** | pdu\_30[7] = {0x3A, 0x3B, 0x1C, 5, 6, 6, 7}; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:30

a msg queue is created: 9e2f090 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_31

|  |  |
| --- | --- |
| **说明** | 接收只有CE，有插入padding子头的MAC PDU； |
| **输入** | pdu\_31[8] = {0x3F,0x3A, 0x3B, 0x1C, 5, 6, 6, 7}; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:31

a msg queue is created: 89e8090 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_32

|  |  |
| --- | --- |
| **说明** | 接收只有CE，有普通padding的MAC PDU； |
| **输入** | pdu\_32[9] = {0x3A, 0x3B, 0x3C, 0x1F, 5, 6, 6, 7,0}; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:32

a msg queue is created: 86f8090 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_33

|  |  |
| --- | --- |
| **说明** | 接收有CE和SDU，无padding的MAC PDU |
| **输入** | pdu\_33[303] = {0x3A, 0x3B, 0x3C, 0x21, 0x0A, 0x22, 0x14,0x23, 0x80, 0x80,  0x04, 5, 6, 6, 7, }; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:33

a msg queue is created: 80fa2e0 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_34

|  |  |
| --- | --- |
| **说明** | 接收有CE和SDU，有1字节插入padding子头的MAC PDU； |
| **输入** | pdu\_34[304] = {0x3F, 0x3A, 0x3B, 0x3C, 0x21, 0x0A, 0x22, 0x14, 0x23, 0x80,  0x80, 0x04, 5, 6, 6, 7}; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:34

a msg queue is created: 8fac2e0 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_35

|  |  |
| --- | --- |
| **说明** | 接收有CE和SDU，有2字节插入padding子头的MAC PDU； |
| **输入** | pdu\_35[304] = {0x3F, 0x3F, 0x3A, 0x3B, 0x3C, 0x21, 0x0A, 0x22, 0x14,0x23,  0x7F, 0x04, 5, 6, 6, 7}; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:35

a msg queue is created: 8da22e0 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_36

|  |  |
| --- | --- |
| **说明** | 接收有CE和SDU，有普通padding子头的MAC PDU； |
| **输入** | pdu\_36[305] = {0x3A, 0x3B, 0x3C, 0x21, 0x0A, 0x22, 0x14, 0x23, 0x7F, 0x24,  0x80, 0x80, 0x1F, 5, 6, 6, 7}; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:36

a msg queue is created: 8d462e0 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

\*the PHR CE hase recieved by power control module\*

the data of the PHR CE is 5

\*the C\_RNTI CE hase recieved by RA module\*

the data of the C\_RNTI CE is 66

\*the BSR CE hase recieved by ul schedule module\*

the data of the short or truncated BSR CE is 7

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 1 is:0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 2 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 3 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\*the RLC SDU hase recieved by RLC\*

the data of the SDU corresponding to lcid 4 is:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

#### 测试用例：HENB\_MAC\_RX\_37

|  |  |
| --- | --- |
| **说明** | 错误信息的接收。 |
| **输入** | pdu\_37[7] = {0x2B, 0x3B, 0, 0, 0, 0, 0}; |
| **输出** | 打印信息，并返回 |

**测试修改**

**测试程序**

create\_mac\_msg( ) {

get\_pdu\_len（）；

create\_ulschtb\_ind（）；

create\_mac\_pdu（）；

}

create\_msgq（）；

init\_pdu\_rx();

**测试方法**

设置输入参数，直接调用该函数。

**测试输出**

##################################

#### mac\_pdu\_rx.c test start ####

##################################

the test case is:37

a msg queue is created: 9008090 !

begin to init\_pdu\_rx

end in init\_pdu\_rx

send msgq succese!

\*DO Ul HARQ\*

MAC\_RX: reserved lcid

MAC\_RX: unpack\_subhdr error!

MAC\_RX: unpack\_mac\_pdu error!

##################################

#### mac\_pdu\_rx.c test end ####

##################################

**BUG描述**

无。

# 6 其他测试

无。

# 附录1 测试类型

## 1. 等价类划分

等价类划分是一种典型的黑盒测试方法。等价类是指某个输入域的集合。它表示对揭露程序中的错误来说，集合中的每个输入条件是等效的。因此我们只要在一个集合中选取一个测试数据即可。等价类划分的办法是把程序的输入域划分成若干等价类，然后从每个部分中选取少数代表性数据当作测试用例。这样就可使用少数测试用例检验程序在一大类情况下的反映。

在考虑等价类时，应该注意区别以下两种不同的情况：

有效等价类：有效等价类指的是对程序的规范是有意义的、合理的输入数据所构成的集合。在具体问题中，有效等价类可以是一个，也可以是多个。

无效等价类：无效等价类指对程序的规范是不合理的或无意义的输入数据所构成的集合。对于具体的问题，无效等价类至少应有一个，也可能有多个。

确定等价类有以下几条原则：

如果输入条件规定了取值范围或值的个数，则可确定一个有效等价类和两个无效等价类。例如，程序的规范中提到的输入条包括“……项数可以从1到999……”，则可取有效等价类为“l考项数＜999”，无效等价类为“项数＜l，，及“项数＞999”。

**输入条件规定了输入值的集合，或是规定了“必须如何”的条件，则可确定一个有效等价类和一个无效等价类。如某程序涉及标识符，其输入条件规定“标识符应以字母开头……”则“以字母开头者”作为有效等价类，“以非字母开头”作为无效等价类。**

如果我们确知，已划分的等价类中各元素在程序中的处理方式是不同的，则应将此等价类进一步划分成更小等价类。

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| 。。。。。。  。。。。。。 | 。。。。。。  。。。。。。 | 。。。。。。  。。。。。。 |

根据已列出的等价类表，按以下步骤确定测试用例：

为每个等价类规定一个唯一的编号；

设计一个测试用例，使其尽可能多地覆盖尚未覆盖的有效等价类。重复这一步，最后使得所有有效等价类均被测试用例所覆盖；

设计一个新的测试用例，使其只覆盖一个无效等价类。重复这一步，使所有无效等价类均被覆盖。这里强调每次只覆盖一个无效等价类。这是因为一个测试用例中如果含有多个缺陷，有可能在测试中只发现其中的一个，另一些被忽视。等价类划分法能够全面、系统地考虑黑盒测试的测试用例设计问题，但是没有注意选用一些“高效的”、“有针对性的”测试用例。后面介绍的边值分析法可以弥补这一缺点。

## 2. 边值分析法

边值分析法是列出单元功能、输入、状态及控制的合法边界值和非法边界值，设计测试用例，包含全部边界值的方法。典型地包括IF语句中的判别值，定义域、值域边界，空或畸形输入，末受控状态等。边值分析法不是一类找一个例子的方法，而是以边界情况的处理作为主要目标专门设计测试用例的方法。另外，边值分析不仅考查输入的边值，也要考虑输出的边值。这是从人们的经验得出的一种有效方法。人们发现许多软件错误只是在下标、数据结构和标量值的边界值及其上、下出现，运行这个区域的测试用例发现错误的概率很高。

用边值分析法设计测试用例时，有以下几条原则：

如果输入条件规定了取值范围，或是规定了值的个数，则应以该范围的边界内及刚刚超出范围的边界外的值，或是分别对最大、最小及稍小于最小、稍大于最大个数作为测试用例。如有规范“某文件可包含l至200”个记录……“，则测试用例可选1和200及0和206等。

针对规范的每个输出条件使用原则〔a〕。

如果程序规范中提到的输入或输出域是个有序的集合(如顺序文件、表格等)就应注意选取有序集的第一个和最后一个元素作为测试用例。

分析规范，尽可能找出可能的边界条件。一个典型的边值分析例子是三角形分类程序。选取a，b，c构成三角形三边，“任意两边之和大于第三边”为边界条件。边值分析相等价类划分侧重不同，对等价类划分是一个补充。如上述三角形问题，选取a＝3，b＝4，c＝0，a＝2，b＝4，c＝7则覆盖有效和无效等价类。如果能在等价类划分中注入边值分析的思想。在每个等价类中不只选取一个覆盖用例，而是进而选取该等价类的边界值等价类划分法将更有效，最后可以用边值分析法再补充一些测试用例。

## 3. 语句覆盖

程序的某次运行一般并不能执行到其中的每一个语句，因此，如果某语句含有一个错误，而它在测试中没执行，这个错误就不可能被发现。为了提高发现错误的可能性，应该在测试时至少要执行程序中的每一个语句。

所谓“语句覆盖”测试标准，它的含义是：选择足够的测试用例，使得程序中每个语句至少都能执行一次。

例子：

Procedure Example(Var A,B,C:real)

begin

if(A>1)and(B=0)

then x:=x/A；

if(A=2)or(x>1)

then x:=x+l

end;

为了使程序中每个语句至少执行一次，只需设计一个能通过路径ace的例子就可以了。例如选择输入数据为：

A=2，B=0，x=3

就可达到“语句覆盖”标准。

显然，语句覆盖是一个比较弱的覆盖标准。如果第一个条件语句中的and错误地写成or，上面的测试用例是不能发现这个错误的，或者是第二个条件语句中x>1误写成x>0，这个测试用例也不能暴露它。我们还可以举出许多错误情况是上述测试数据不能发现的。所以，一般认为“语句覆盖”是很不充分的最低的一种覆盖标准。

## 4. 判定覆盖

比“语句覆盖”稍强的覆盖标准是“判定覆盖”(或称分支覆盖)。这个标准是：执行足够的测试用例，使得程序中每个判定至少都获得一次“真”值和“假”值，即使得程序中的每一个分文至少都通过一次。

对上面那个例子，如果设计两个测试用例，就可以达到“判定覆盖”的标难。为此，我们可以选择输人数据为：

(1)A=3，B=0，x=l

(2)A=2，B=1，x=3

“判定覆盖”比“语句覆盖”严格，因为如果每个分支都执行过了，自然每个语句也就执行了。

## 5. 条件覆盖

它的含义是：执行足够的测试用例，使得判定中每个条件获得各种可能的结果。

对于例子程序,我们只需设计以下两个测试用例就可满足这标准：

(1)A＝2，B＝o，x＝4(沿路径ace执行)

(2)A＝1，B＝l，x＝l(沿路径aN执行)

虽然同样只要两个测试用例，但它比判定覆盖中两个测试用例更有效。一般来说，“条件覆盖”比“判定覆盖”强，但是，并不总是如此，满足“条件覆盖”不一定满足“判定覆盖”。例如对语句。

IF(A AND B)THEN S

设计两个测试用例：A“真”B“假”和A“假”B“真”。对于上例我们设计两个测试用例为：

(1)A＝1，B＝o，x＝3

(2)A＝2，B＝l，x＝1

亦是如此，它们能满足“条件覆盖”但不满足“判定覆盖”。

## 6. 判定／条件覆盖

针对上面的问题引出了另一种覆盖标准，这就是“判定／条件覆盖”，它的含义是：执行足够的测试用例，同时满足判定覆盖和条件覆盖的要求。显然，它比“判定覆盖”和“条件覆盖”都强。

对于例子程序，我们选取测试用例：

(1)A=2，B=0，x=4

(2)A=1，B=l，x=l

它满足判定／条件覆盖标准。

值得指出，看起来“判定／条件覆盖”似乎是比较合理的，应成为我们的目标，但是事实并非如此，因为大多数计算机不能用一条指令对多个条件作出判定，而必须将源程序中对多个条件的判定分解成几个简单判定。这个讨论说明了，尽管“判定／条件覆盖”看起来能使各种条件取到所有可能的值，但实际上并不一定能检查到这样的程度。针对这种情况，有下面的条件组合覆盖标准。

## 7. 条件组合覆盖

“条件组合覆盖”的含义是：执行足够的测试用例，使得每个判定中条件的各种可能组合都至少执行一次。这是一个最强的逻辑覆盖标准。

再看例子程序，必须使测试用例覆盖八种组合结果

(1)A>1，B=0 (0)A=2，x>1

(2)A>1，B<>0 (6)A=2，x<1

(3)A<l，B=0 (7)A<>2，x>1

(4)A<1，B<>0 (8)A<>2，x<1

必须注意到，(0)、(6)、(7)、(8)四种情况是第二个条件语句的条件组合，而x的值在该语句之前是要经过计算的，所以我们还必须根据程序的逻辑推算出在程序的人口点x的输入值应是什么。

要测试八个组合结果并不是意味着需要八种测试用例，事实上，我们能用四种测试用例来覆盖它们：

(1)A＝2，B＝o，x＝4；

(2)A＝2，B＝1，x＝l；

(3)A＝l，B＝o，x＝2；

(4)A＝1，B＝1，x＝l。

上面四个例子虽然满足条件组合覆盖，但并不能覆盖程序中的每一条路径，可以看出条件组合覆盖仍然是不彻底的，在白盒测试时，要设法弥补这个缺陷。