# 1，Iacs.cc

宏定义，enum定义

方法定义

IACS类：

构造函数（921q,931q,StackMgrq,bufferpool)

1，初始化变量

2，注册trace

3，初始化timer：IACS\_xtm\_inittimer

init()：初始化

1，为每个line创建一个ISDNItf对象：iacsCreateISDNInterface

2，文件内静态变量申请内存空间

SignallingTick()：消息接收死循环

1，接收padr frame，调用对应line上的isdnItfProcessISDNFrame方法处理frame

2，接收StackMgr消息，获取lineid

3，接收Q931消息，获取lineid

4，调用指定line上的isdnItfProcessQueue4Message方法处理SM和931消息

sendTimeoutEvtToQ4();

发送timeout事件给自己本身处理

setCdeValues():

1，设置Cde的值

2，调用init方法初始化

[TxStatisticsFrame](../textNotes/Q921FrameAnalysis.cc)(): 解析Tx frame桢(D信道)

RxStatisticsFrame(): 解析Rx frame桢(B信道)，两者的处理流程基本一致

RxClearStatistics()

TxClearStatistics(): 清除桢分析结果

IACS\_PrintL2L1Buffer：打印frame信息

IACS\_PrintL2L3Buffer：打印buffer信息

[IACS\_PrintQ931Packet](../textNotes/Q931FrameAnalysis.cc)：打印L2L3buffer中Q931 Msg Buffer

//timer相关

IACS\_xtm\_inittimer()：

1，timer\_flag初始化赋值

2，创建新task，运行IACS\_timer\_process

IACS\_timer\_process()：

IACS的timer管理方法，处理timer事件

IACS\_xtm\_createtimer()：创建timer，供其他类调用

创建IACS\_TMR\_FUNC\_CTXT对象，赋值生成新的timer，返回timer id

IACS\_xtm\_deletetimer()：删除timer

# 2，ISDNItf.cc

ISDNItf类：——q921接口，每个line对应一个实例

构造函数（Itf对象Id，931q，StackMgrq……bufferpool）

1. 初始化内部变量
2. 创建执行timeoutT201函数的timer
3. 创建执行timeroutTm01函数的timer

析构函数：

1. 释放ISDNItf对象链表的空间
2. 删除两个timer

isdnItfFind()：根据id获取链表中的对应ISDNItf对象

isdnItfL1Activate(): 判断L1当前状态是否为active，若非赋值active，调用teiIdCheck

isdnItfL1Deactivate：设置L1当前状态为inactive，调用deleteAll删除所有L2IACS对象

l2IACSProcess()：处理ivps消息

isdnItfProcessQueue4Message()：处理StackMgr和Q931消息

isdnItfProcessISDNFrame()：调用L2IACS对象的l2IACSProcess方法处理L1发送的921 frame消息（L2Frame对象，非原始形式）

l2IACSProcess()：

1. 处理teiIdRequestMsg消息
2. 处理teiIdCheckRespMsg消息
3. 处理teiIdVerifyMsg消息

timeroutT201(): 调用指定ISDNItf对象的treatT201Timer方法

treatT201Timer()：创建/删除L2IACS对象

timeroutTm01()：调用指定ISDNItf对象的treatTm01Timer方法

treatTm01Timer()：发送l1 deactive请求到StackMgr

checkL2Entity()：检查L2实体对象的状态

1. l1当前为inactive时，发送l1 active请求到StackMgr
2. l1当前为active状态时，创建L2IACS对象加入到L2对象链表中

checkL2Result()：处理L2对象检查的结果

1. error C/D：删除L2对象
2. error G/H，delconn：调用oneL2Less方法发送no link消息给StackMgr
3. newconn：调用oneL2More发送link establish消息给StackMgr

sendTEIMgmt()：发送921 frame到PADRProxy，继而发给L1

sendQ931MsgtoZips()：发送消息给Q931

# 3，L2MapIACS.cc

L2Frame类：921 frame和对象成员之间的映射处理

Unfold：

解析frame根据消息类型的不同构造L2Frame的对应类型子类对象，传入桢数据

L2Frame：

调用PADRProxy接口获取buffer赋值给内部成员data

sendMe：

1. 调用TxStatisticsFrame解析frame
2. 调用IACS\_PrintL2L1Buffer输出一下buffer内容
3. 调用PADRProxy接口发送data到L1

L2numbered类：继承L2Frame，映射I和S格式桢

L2Unnumbered类：继承L2Frame，映射U格式桢

L2InvalidFrame类：继承L2Frame，映射无效桢

其他继承L2numbered和L2Unnumbered的具体消息类型的类定义

# 4，L2IACS.cc

L2IACS类：

构造函数：

1. 创建timer运行timeoutT200方法
2. 创建timer运行timeoutT203方法

析构函数：

停止删除T200和T203的timer

timeroutT200：发送指定L2IACS对象的T200事件给IACS 处理

timeroutT203：发送指定L2IACS对象的T203事件给IACS 处理

treatT200Timer：处理T200事件

treatT203Timer：处理T203事件

l2IACSProcess：多种重载实现，分别处理对应921帧消息，返回L2CommonIACSResult

initiateConnection：初始化连接

1. 发送sabme命令到padr
2. 启动T200 timer

initiateDisconnection：断开连接

1. 发送disc命令道padr
2. 启动T200 timer

sendAcknowledgment：发送确认信息给padr

sendQ931MsgtoZips：发送消息到Q931

# 5，E1激活

## 5.1 IWU主动下发unblock消息：

1，StackMgr发送unblock消息给PADR;StackMgr发送ZIPS\_to\_IACS\_UNBLOCK\_MSG给Q921

Q921处理ZIPS\_to\_IACS\_UNBLOCK\_MSG消息

|  |
| --- |
| **IACS::SignallingTick()**  case ZIPS\_to\_IACS\_UNBLOCK\_MSG:  |-->itfId = xmsg\_data1(msg\_buf);  ISDNItf\* isdnObject = ISDNItf::isdnItfFind(itfId);  isdnObject->isdnItfProcessQueue4Message(msg\_buf);  |-->case ZIPS\_to\_IACS\_UNBLOCK\_MSG: isdnItfEnable();  | |-->admStatus\_ = enabled; |

2，PADR收到FE201（unblock）消息，执行E1上电操作，之后上报L1active的L1Event事件给StackMgr

3，StackMgr收到L1active事件，进行处理

|  |
| --- |
| L1Event():  |-->case L1ACTIVE:  | |-->sendToIacs = IacsL1Activate;  | |-->ucMsgToServ = SM\_L1ACTIVE\_IND;  | |--> newL1Status = L1activated;  ……  |-->case IacsL1Activate:  | |-->sendMsgtoIACS( ZIPS\_to\_IACS\_ACTIVATE\_RESULT\_MSG); |

4，Q921收到ZIPS\_to\_IACS\_ACTIVATE\_RESULT\_MSG，设置L1状态active

|  |
| --- |
| **IACS::SignallingTick()**  case ZIPS\_to\_IACS\_ACTIVATE\_RESULT\_MSG::  |-->itfId = xmsg\_data1(msg\_buf);  ISDNItf\* isdnObject = ISDNItf::isdnItfFind(itfId);  isdnObject->isdnItfProcessQueue4Message(msg\_buf);  |--> case ZIPS\_to\_IACS\_ACTIVATE\_RESULT\_MSG: isdnItfL1Activate();  | |--> l1Status\_ = active;  | |-->teiCheck();  | | |-->sendTEIMgmt(0, teiIdCheckReqMsg, groupTei\_);  | | |-->startT201(); |

5，padr发送sabme请求，921进行处理

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer); //转换成frame类  | |-🡪 pNewFrame = new L2SABME(data);  | | |--🡪 L2Unnumbered::L2Unnumbered (IRD char\* data) : L2Frame(data)  | | | |--🡪 L2Frame::L2Frame (IRD char \* data): buffer\_(data),data\_((u8 \*) buffer\_)  |u16 ifID = pFrame->getIfId();  |ISDNItf\* isdnObject = ISDNItf::isdnItfFind(ifID);  |ISDNItf::isdnItfProcessISDNFrame ( L2Frame \* fromISDN)  ｜―――＞L2CommonIACS \* entity = checkL2Entity(tei, 0);  ｜　　｜―――＞result = l2List\_[tei];  |--🡪 case L2Frame::SABME:  | |-🡪L2IACS::l2IACSProcess (L2SABME \* msg)  | | |--🡪 sendQ931MsgtoZips(BA\_EST\_IND\_MESSAGE,1, 0, NULL);  | | |-🡪 setMultiframeMode(pf, true);  | | | |--🡪 L2UA l2UA(ifId\_, tei\_, callControlSapi\_, pf); //创建确认信息UA  | | |--> state\_ = multipleFrameEstablished;  |--🡪 checkL2Result(retL2, tei)  | |-🡪case newConn:  | | |-🡪 oneL2More() //发送L2 active消息给isdnstackmgr |

6，StackMgr收到Q921发来的IACS\_LINKESTABLISH\_MSG消息，发送L2active的消息给IWU

## 5.2 底层插线

直接插线的情况下，会从5.1小节的第2步开始执行，PADR发出LOS警告清除消息，同时发送L1Inactive消息给StackMgr，后续流程一致

# 6，outgoing call

1，921收到padr的setup消息，进行处理

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);6  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer); //转换成frame类  | |--> pNewFrame = new L2I(data);  |u16 ifID = pFrame->getIfId();  |ISDNItf\* isdnObject = ISDNItf::isdnItfFind(ifID);  |ISDNItf::isdnItfProcessISDNFrame ( L2Frame \* fromISDN)  | |-->L2CommonIACS \* entity = checkL2Entity(tei, 0);  | |-->case L2Frame::I: retL2 = entity->l2IACSProcess((L2I \*) fromISDN);  | | |-->sendQ931MsgtoZips(BA\_DATA\_MESSAGE,1, dataSize, pData);  | | |-->result = multiframeMode(msg, withAck, false);  | | | |-->sendAcknowledgment(false, pf);  | | | |--> sendQueuedFrames();  | |--> checkL2Result(retL2, tei); |

2，931收到BA\_DATA\_MESSAGE消息，处理发送ASP\_DATA\_REQ\_MESSAGE给921

|  |
| --- |
| **IACS::SignallingTick()**  |-->isdnObject->isdnItfProcessQueue4Message(msg\_buf);  | |--> case Q931\_MSG: entity->l2IACSProcess(msg\_buf);  | | |--> case ASP\_DATA\_REQ\_MESSAGE: sendQueuedFrames();  | | | |--> pL2I->sendMe(vr\_);  | | |-->isdnObj->checkL2Result(result, tei\_); |

3, padr收到setup ack的I帧之后，发送RR帧给921表示接受到ack

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);6  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer);  | |--> pNewFrame = new L2RR(data);  |u16 ifID = pFrame->getIfId();  |ISDNItf\* isdnObject = ISDNItf::isdnItfFind(ifID);  |ISDNItf::isdnItfProcessISDNFrame ( L2Frame \* fromISDN)  | |-->L2CommonIACS \* entity = checkL2Entity(tei, 0);  | |--> case L2Frame::RR: retL2 = entity->l2IACSProcess((L2RR \*) fromISDN); |

4，padr发送information的I帧给921，921处理

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);6  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer); //转换成frame类  | |--> pNewFrame = new L2I(data);  |u16 ifID = pFrame->getIfId();  |ISDNItf\* isdnObject = ISDNItf::isdnItfFind(ifID);  |ISDNItf::isdnItfProcessISDNFrame ( L2Frame \* fromISDN)  | |-->L2CommonIACS \* entity = checkL2Entity(tei, 0);  | |-->case L2Frame::I: retL2 = entity->l2IACSProcess((L2I \*) fromISDN);  | | |-->sendQ931MsgtoZips(BA\_DATA\_MESSAGE,1, dataSize, pData);  | | |-->result = multiframeMode(msg, withAck, false);  | | | |-->sendAcknowledgment(false, pf);  | | |--> sendQueuedFrames();  | |--> checkL2Result(retL2, tei); |

5，931收到BA\_DATA\_MESSAGE消息，处理发送ASP\_DATA\_REQ\_MESSAGE给921

|  |
| --- |
| **IACS::SignallingTick()**  |-->isdnObject->isdnItfProcessQueue4Message(msg\_buf);  | |--> case Q931\_MSG: entity->l2IACSProcess(msg\_buf);  | | |--> case ASP\_DATA\_REQ\_MESSAGE: sendQueuedFrames();  | | | |--> pL2I->sendMe(vr\_);  | | |-->isdnObj->checkL2Result(result, tei\_); |

6,padr收到call proceeding的I帧之后，发送RR帧给921表示接受到proceed

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);6  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer);  | |--> pNewFrame = new L2RR(data);  |u16 ifID = pFrame->getIfId();  |ISDNItf\* isdnObject = ISDNItf::isdnItfFind(ifID);  |ISDNItf::isdnItfProcessISDNFrame ( L2Frame \* fromISDN)  | |-->L2CommonIACS \* entity = checkL2Entity(tei, 0);  | |--> case L2Frame::RR: retL2 = entity->l2IACSProcess((L2RR \*) fromISDN); |

后面alerting和通话建立的过程不需要921的参与

# 7，incoming call

1，有电话打来，931发送连接建立请求给921

|  |
| --- |
| **IACS::SignallingTick()**  |-->isdnObject->isdnItfProcessQueue4Message(msg\_buf);  | |--> case Q931\_MSG: entity->l2IACSProcess(msg\_buf);  | | |--> case ASP\_EST\_REQ\_MESSAGE：  | | | |--> case TEIAssigned: initiateConnection();  | | | | |--> l2SABME.sendMe();  | | | | |-->state\_ = awaitingEstablishment; |

2，padr处理之后发送UA确认给921，921进行处理

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);6  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer);  | |--> pNewFrame = new L2UA(data);  | isdnObject->isdnItfProcessISDNFrame(pFrame);  | |--> case L2Frame::UA: retL2 = entity->l2IACSProcess((L2UA \*) fromISDN);  | | |--> case awaitingEstablishment:  sendQ931MsgtoZips(BA\_EST\_CONF\_MESSAGE, 1, 0, NULL); |

3，931接着通过DL-data发送setup消息给921，921通知padr

|  |
| --- |
| **IACS::SignallingTick()**  |-->isdnObject->isdnItfProcessQueue4Message(msg\_buf);  | |--> case Q931\_MSG: entity->l2IACSProcess(msg\_buf);  | | |--> case ASP\_DATA\_REQ\_MESSAGE：sendQueuedFrames();  | | | |--> pL2I->sendMe(vr\_); |

4，padr收到setup的RR帧之后，发送RR回复给921

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);6  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer);  | |--> pNewFrame = new L2RR(data);  | isdnObject->isdnItfProcessISDNFrame(pFrame);  | |--> case L2Frame::RNR: retL2 = entity->l2IACSProcess((L2RR \*) fromISDN);  | | |--> result = multiframeMode(msg, withAck, false); |

5，padr发送call proceeding的I帧给921处理

|  |
| --- |
| **IACS::SignallingTick():**  |(Zips\_BADR\_proxy::getZips\_BADR\_proxy())->getFrame(&pBuffer))  | |--🡪 retCode = getHdlcFrame(frame);6  | | |--🡪 read\_rngbuff(&greadpointer, &frame))  |IACS\_PrintL2L1Buffer  |pFrame = L2Frame::unfold(pBuffer);  | |--> pNewFrame = new L2I(data);  | isdnObject->isdnItfProcessISDNFrame(pFrame);  | |--> case L2Frame::I: retL2 = entity->l2IACSProcess((L2I \*) fromISDN);  | | |-->sendQ931MsgtoZips(BA\_DATA\_MESSAGE,1, dataSize, pData);  | | |-->result = multiframeMode(msg, withAck, false);  | | | |-->sendAcknowledgment(false, pf); |

后面alerting和通话建立的过程不需要921的参与

# 8，E1去激活

## 8.1 IWU主动下发block消息：

1，StackMgr接收到IWU的block消息，发送block消息给PADR;

|  |
| --- |
| Block():  |-->sendEventToPADR(FE203);  |-->sendMsgtoIACS(ZIPS\_to\_IACS\_BLOCK\_MSG); |

2，Q921收到ZIPS\_to\_IACS\_BLOCK\_MSG消息，设置adminstatus为disable，同时调用isdnItfL1Deactivate函数

|  |
| --- |
| case ZIPS\_to\_IACS\_BLOCK\_MSG:  |-->isdnItfDisable();  | |--> admStatus\_ = disabled;  | |--> isdnItfL1Deactivate(); |

3，PADR收到FE203消息后，首先发出LOS警告，然后对E1进行下电操作，发送L1Inactive的L1Event给StackMgr

4，Stackmgr收到LOS警告上报VappLsm，显示警告

|  |
| --- |
| case PADR\_LOS:  |-->usrP\_m[ulUp]->reportAlarmEvent(ulEventType,ulStateType); |

5，Stackmgr收到L1Inactive消息

|  |
| --- |
| L1Event():  |-->case L1INACTIVE:  | |-->newL1Status = L1deactivated;  | |-->sendToIacs = IacsL1Deactivate;  | |-->ucMsgToServ = SM\_L1DOWN\_IND;  ……  |-->case IacsL1Deactivate:  | |-->sendMsgtoIACS( ZIPS\_to\_IACS\_DEACTIVATE\_RESULT\_MSG); |

6，Q921收到ZIPS\_to\_IACS\_DEACTIVATE\_RESULT\_MSG消息，设置L1状态，删除L2链接

|  |
| --- |
| case ZIPS\_to\_IACS\_DEACTIVATE\_RESULT\_MSG:  |-->isdnItfL1Deactivate();  | |--> l1Status\_ = inactive;  | |--> deleteAll(); |

## 8.2 底层直接拔线

直接拔线的情况下，会从8.1小节的第3步开始执行，PADR发出LOS警告，同时发送L1Inactive消息给StackMgr，后续流程一致