# 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

# E1激活流程

1，E1线插入，padr发送sabme请求过来:

|  |
| --- |
|  |

StackMgr发送unblock消息给PADR

1. StackMgr发送ZIPS\_to\_IACS\_UNBLOCK\_MSG给Q921
2. 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; |

4，PADR收到unblock消息，向921发送TEI申请

5，921收到TEI请求帧，解析处理，发送IACS\_ACTIVE\_REQ\_MSG给StackMgr，之后发送BA\_TEI\_I\_MESSAGE给931

|  |
| --- |
| **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 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);  | |-->xmsg\_msgtype(msg\_buf) = IACS\_ACTIVATE\_REQ\_MSG;  | |-->return 0;  |-->teiIdRemove(tei);  | |-->sendQ931MsgtoZips(BA\_TEI\_I\_MESSAGE,tei, 0); |

6，StackMgr发送ZIPS\_to\_IACS\_ACTIVATE\_RESULT\_MSG给921，921处理设置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; |

7，

7,921处理931返回的消息，发送回复UI给PADR

|  |
| --- |
| **IACS::SignallingTick()**  case Q931\_MSG:  ISDNItf\* isdnObject = ISDNItf::isdnItfFind(itfId);  isdnObject->isdnItfProcessQueue4Message(msg\_buf);  |--> entity = checkL2Entity(tei, 1)  |--> entity->l2IACSProcess(msg\_buf);  | |--> case ASP\_UDATA\_REQ\_MESSAGE:  | | |--> frame = new L2UI(ifId\_, tei\_, sapi, data, dataLength);  | | |-->frame->sendMe();  | |-->isdnObj = ISDNItf::isdnItfFind(ifId\_);  | |-->isdnObj->checkL2Result(result, tei\_); |