

CMUX 协议文档

修订情况记录：

版本号	修订人	修订原因	修订日期	审核	测试	签收
V0.8	徐兵林	新建	2008-11-4			
V1.0	高金山	完成	2009-2-11			

目录

CMUX 协议文档 1

1. 引言 4

1.1. 编写目的 4

1.2. 编写背景 4

1.3. 参考资料 4

1.4. 名字解释 4

2. CMUX 协议框架 5

3. Non Error Recovery Mode 8

3.1. 服务接口定义 8

 3.1.1. 服务模型 8

 3.1.2. 启动 CMUX 服务模式 9

 3.1.3. 建立 DLC 服务 9

 3.1.4. 数据服务 10

 3.1.5. 功耗控制 10

 3.1.5.1. Sleep 模式 11

 3.1.5.2. Wake up 模式 11

 3.1.6. 释放 DLC 服务 11

 3.1.7. 关闭服务 11

 3.1.8. 控制服务 11

 3.1.8.1. 07.10 协议服务 12

 3.1.8.2. 虚拟端口服务 13

3.2. 数据帧结构 14

 3.2.1. 帧域 14

 3.2.1.1. 标志域 (Flag Sequence Field) 14

 3.2.1.2. 地址域 (Address Field) 14

 3.2.1.3. 控制域 (Control Field) 15

 3.2.1.4. 信息域 (Information Field) 15

 3.2.1.5. 长度指示域 (Length Indicator) 15

 3.2.1.6. FCS 域 (Frame Checking Sequence Field) 15

 3.2.2. 格式约定 16

 3.2.3. 帧有效性 16

 3.2.4. 帧中止 16

 3.2.5. 数据帧之间的填充 16

 3.2.6. 基本 Basic 16

 3.2.6.1. 约束 17

 3.2.7. 高级 Advanced 17

 3.2.7.1. 控制字节透明 17

 3.2.7.2. 开始 /停止传输 -扩展透明 17

 3.2.7.3. 流控 (Flow-control) 透明 18

 3.2.7.4. 帧的结构 18

3.3. 帧类型 18

3.4. 过程 and 状态 20

 3.4.1. 建立 DLC 链路 20

 3.4.2. 释放 DLC 链路 20

 3.4.3. 信息传输 21

 3.4.4. 帧变量 21

 3.4.5. 超时的考虑 22

 3.4.6. 多路控制通道 22

 3.4.6.1. 控制消息格式 22

 3.4.6.2. 控制消息类型参数 23

 3.4.7. 电源控制与唤醒机制 32

 3.4.8. 流控 32

3.5. 集成层 Convergence Layer 34

 3.5.1. 类型 1-未结构化的字节流 34

 3.5.2. 类型 2-带参数的未结构化的字节流 34

 3.5.3. 类型 3-不可中断的帧数据 36

 3.5.4. 类型 4-可中断的帧数据 36

3.6. DLCI 值 37

3.7. 系统参数 37

 3.7.1. 确认时间 T1 37

 3.7.2. 帧的最大长度 N1 38

 3.7.3. 最大重发次数 N2 38

 3.7.4. 窗口大小 k 38

 3.7.5. 控制通道的响应时间 T2 38

 3.7.6. 唤醒流程的响应时间 T3 38

3.8. 启动和关闭 MUX 38

4. Error Recovery Mode 39

1. 引言

1.1. 编写目的

本文档根据 GSM07.10 协议同时结合 E901 Mobile 参考代码，对 CMUX 多串口协议进行了大致的描述。由于 7.10 协议可以使用两种传输模式 non-error 和 error 模式，本文档只介绍 non-error 模式。

1.2. 编写背景

1. 希望 CMUX 协议有一个总体框架；
2. 想要了解 CMUX 多串口协议的通讯设计；

1.3. 参考资料

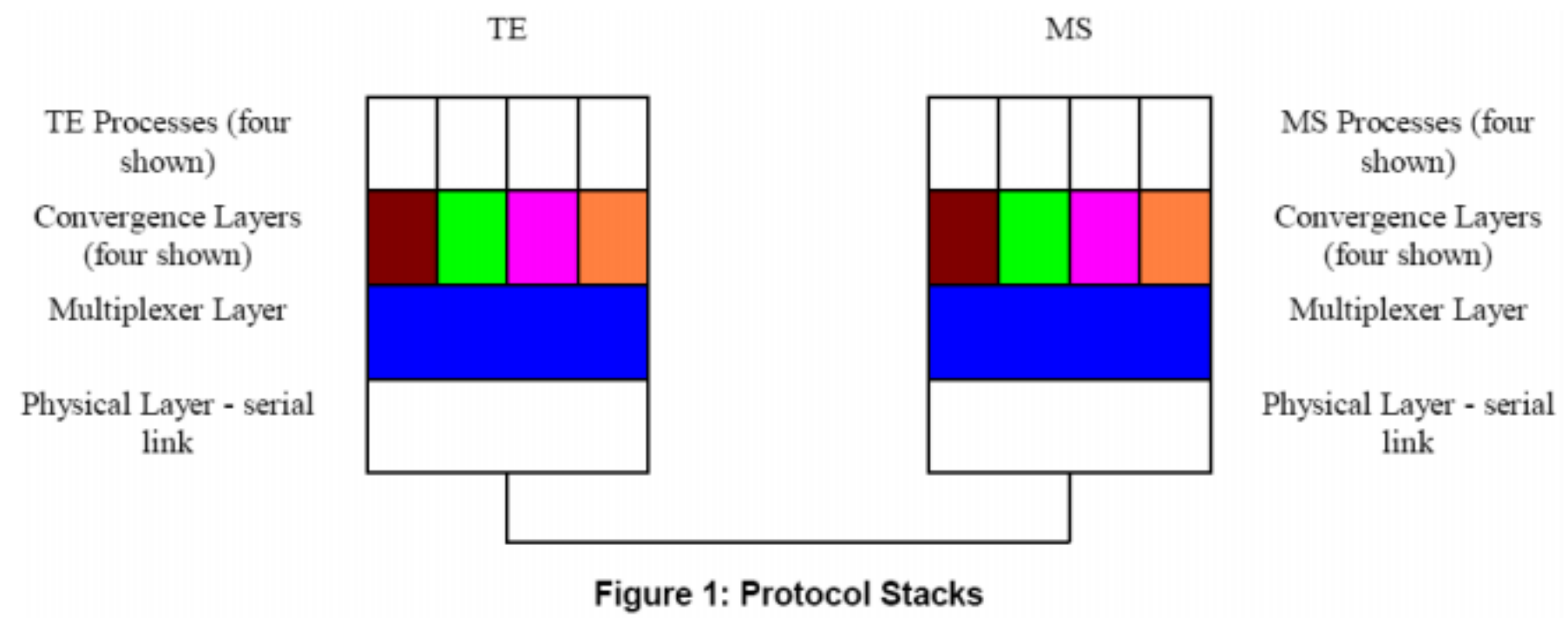
1. GSM07.10 协议
2. E901 Mobile 参考 CMUX 通讯代码

1.4. 名词解释

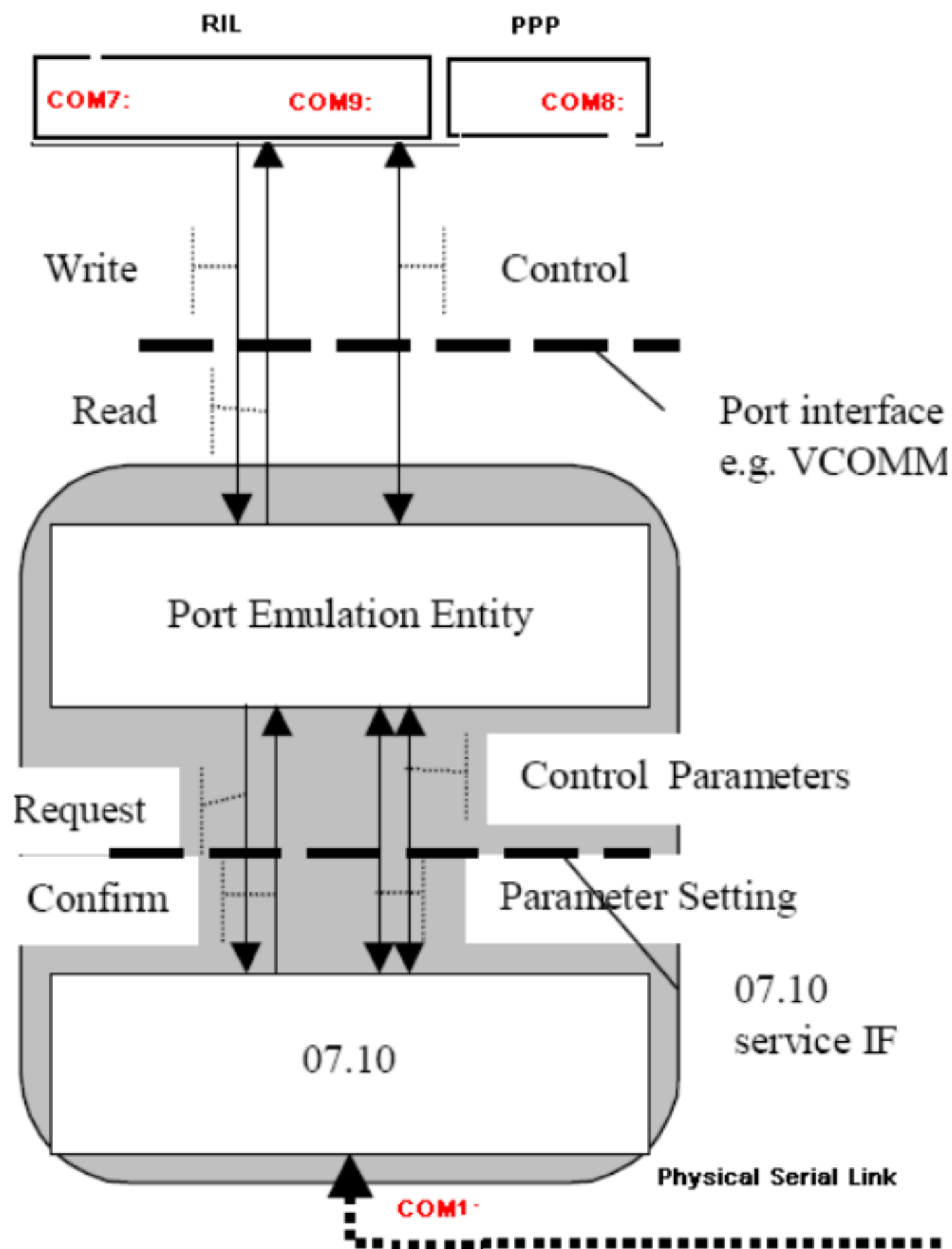
- | | |
|----------|--|
| 1. CMUX | 多串口协议 |
| 2. ABM | Asynchronous Balanced Mode 异步平衡模式 |
| 3. ERM | Error-Recovery Mode 错误校验模式 |
| 4. DLC | Data Link Connection 数据链路连接 |
| 5. FCS | Frame Check Sequence 帧校验序列 |
| 6. SARM | Set Asynchronous Balanced Mode 设置异步平衡模式 |
| 7. UAU | Unnumbered Acknowledgement 未编号的确认信息 |
| 8. DM | Disconnected Mode 断开模式 |
| 9. UIH | Unnumbered Information with Header Check 未编号的带校验头的信息 |
| 10. UI | Unnumbered Information 未编号的信息 |
| 11. PSC | Power Saving Control 省电控制 |
| 12. MSC | Modem 状态命令 |
| 13. HDLC | High-level data link control 高级数据链路控制 |

2. CMUX 协议框架

多路复用协议提供在单个物理通信通道之上虚拟出多个并行的逻辑通信通道的能力，一般应用于 TE(Terminal Equipment) 与 MS(Mobile Station) 之间，TE 相当于智能手机的 AP 端，MS 相当于智能手机的 MODEM 端，下图给出了典型的协议层次关系：



蓝色部分就是 MUX 多路复用层，它利用底层的物理串口链接收发数据，同时向上层提供若干个逻辑上独立使用的收发通道（上图中提供了四个逻辑通道，不同颜色表示）。每个逻辑通道独立创建，可以拥有软件流控制。在实际使用中，TE 端的 MUX 向 MS 端的 MUX 发起通道建立请求，设置通道参数等，是主动的一方；MS 端的 MUX 等待 TE 端的服务请求，根据自身能力提供相应服务。也就是说，两者的角色是不对称的。



GSM07.10 协议就是一个提供 MUX 功能的标准协议，也是实际中较广泛使用的协议。在 E901 项目中，NXP 的 5209 MODEM 软件提供标准的 07.10 MUX 功能，AP 端开发了 07.10 MUX 的驱动，如上图所示，AP 端的 MUX 驱动利用了物理串口 FFUART(COM1:), 同时向系统提供了三个逻辑串口 (COM7 : COM9 : COM8 :), 其中 COM7: 是 AT 命令口 TE< MS ; COM9: 是 URC 口 TE MS ; COM8: 是数据端口 TE< MS , 当建立 GPRS/EDGE 数据连接时使用。这样对 RIL 驱动而言，只看到 COM7 和 COM9 这两个虚拟串口，对 PPP 模块而言，只看到 COM8 虚拟口，真正的物理串口 COM1 只对 MUX 驱动有意义，其他驱动看不到也不关心 COM1。

TE 终端和 MS 终端之间的每个通道，称作一个 DLC 链路，它们是先后相互独立的建立起来的。其数据传输采用 8BIT 字符的 Start-Stop 的方式进行传输，双方的数据交互使用了后面定义的帧结构。出于缓冲区管理的目的，每条 DLC 链路都有各自的流控。总的链路也有全面的流控的机制。

DLC 有两种工作模式 Error-Recovery Mode (ERM) 错误校验模式和 Non-Error-Recovery

Mode (non-ERM) 模式。在 DLC 链路上可以同时使用 ERM 和 nor-ERM 模式。采用哪种模式，在 DLC 建立的时候就决定好了。如果使用 ERM 校验模式，必须把 DLC 配置成 ISO HDLC 透明机制，但是否使用 ERM 校验模式，是可选的。

non-ERM 模式使用 UI 帧和 UIH 帧来装载用户数据。而 ERM 模式使用 I 帧装载用户数据。

CMUX 支持三种操作选项，分别是 Basic, Advanced without Error Recovery 和 Advance with Error Recovery。其中前两种不支持错误纠正。

Basic :

- 1) 数据长度指示器代替 HDLC 的透明机制；
- 2) 与 HDLC 的 flag 标志位不同；
- 3) 不能在数据链路上使用 XON/XOFF 流控
- 4) 从数据同步丢失中恢复，需要比较长的时间。

Advanced without Error Recovery

- 1) 异步的 HDLC 跟 ISO/IEC 13239 .协议一致；
- 2) 可以在数据链路上使用 XON/XOFF 流程控制；
- 3) 从数据同步丢失中恢复，很快。

Advance with Error Recovery

- 1) 使用 HDLC ERM 流程。

3. Non Error Recovery Mode

Non Error Recovery 用在 Basic模式和 Advanced 模式，它是一种不可靠的数据传输，所以在链路很可靠的情况下才使用它。它有一套简单的流程；Advanced 提供了数据透明机制：它可以传输管理和控制信息；它允许 MS或TE 进入省电模式；它分了几个层，可以原封不动的装载不同类型的数据。

透明机制必须在 MUX 会话开始的时候建立，它是整个 MUX 会话的一个特征。

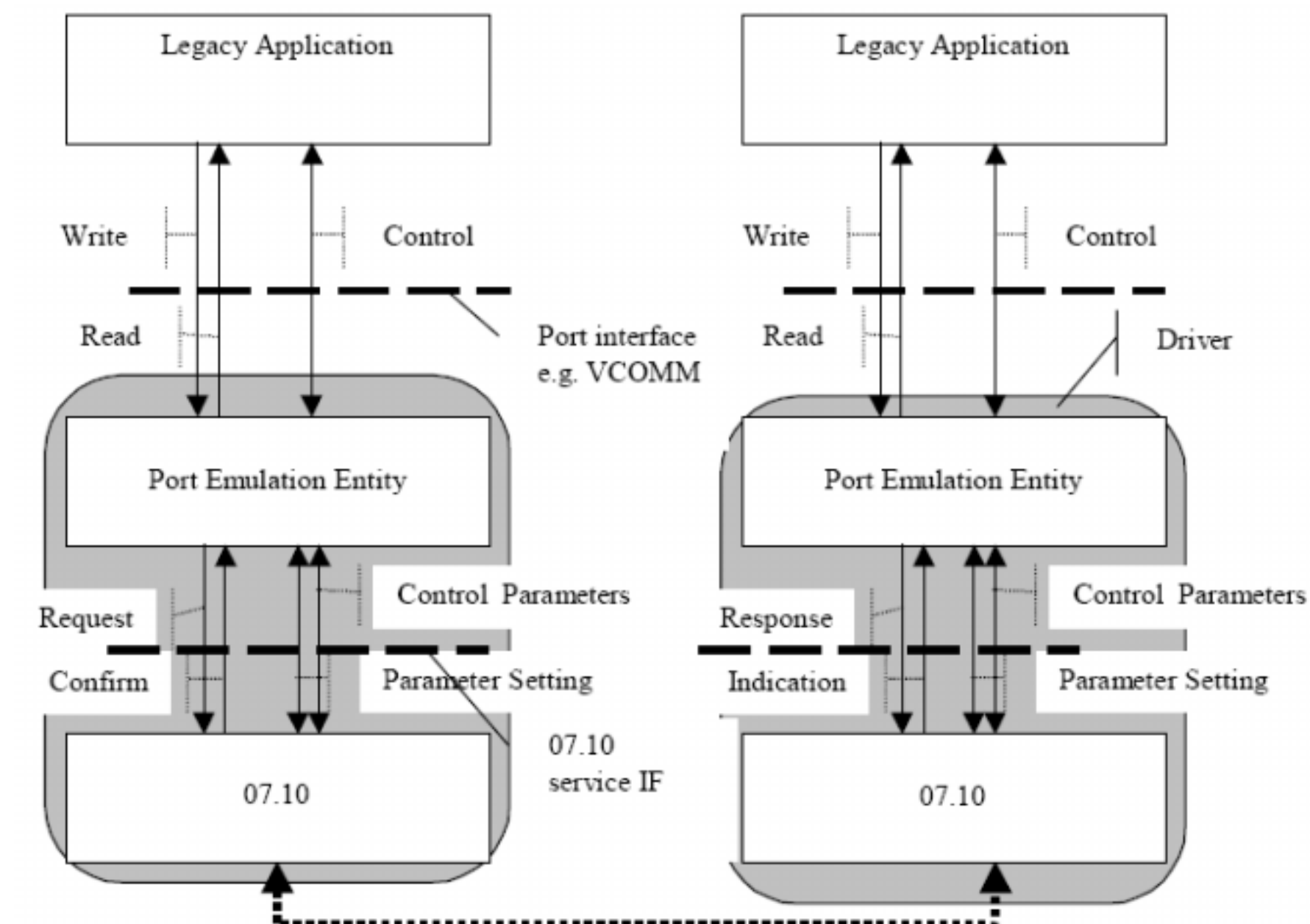
这套简单的流程，使用 UIH 帧装载传输数据。这些帧比较容易处理，因为它使用的 Frame Check Sequence (FCS) 校验机制，可以事先计算，而不需要一个字节一个自己的去计算。这套流程的使用是很直接的，不需要实现 HDLC 的状态机。

与可靠性相比，当及时的传递数据更重要时，该通道 UI 帧或 UIH 帧，都可以用。当对可靠性要求比较高时，那就采用 UI 帧吧。

3.1. 服务接口定义

服务接口用在数据链路层与上层应用之间。这些接口由原语和参数界定。

3.1.1. 服务模型



从图中可以看到双方通讯的两端都需要串口模拟实体，在实体和逻辑层之间建立的数据流通道和控制通道。图中的左端是主动请求发起端。右端是接收方。

3.1.2. 启动 CMUX 服务模式

mode = [Basic | HDLC - UIH frames | HDLC - UI frames | HDLC - frames]. , 如果控制通道用了 HDLC 模式，那么其它用了 DLC 模式的参数配置可以忽略。

system_parameters = Port speed [9,6 | 19,2 | 38,4 | 57,6 | 115,2 |

230,4 kBit/s],

Maximum Frame Size [1 –128 in Basic mode,

1 –512 in HDLC modes

default: 31 for the basic

option and 64 for the

advanced option]

Acknowledgement Timer [0,01s-2,55s,

default: 0,1s]

Maximum number of retransmissions [0 –100,

default : 3]

Response timer for the multiplexer control channel [0,01s-2,55s,

default: 0,3s]

Wake up response timer [1s –255s, default 10s]

Window size for error recovery mode [1 –7,

default : 2]

accept = [true | false]

一点要注意的是各个虚拟通道的一些参数可以不同。根据（图 2）中，启动 CMUX 服务，需要双方（左边是发起方，右边是接收方）建立 4 条消息，

1) 发起方： TS0710_START.request (mode, system_parameters)

2) 发起方： TS0710_START.indication (mode, system_parameters

3) 接收方： TS0710_START.response (mode, system_parameters, accept)

4) 接收方： TS0710_START.confirm (mode, system_parameters, accept)

这 4 条消息建立过程大致如下：

首先发起方发起 request (请求) 消息，接着，接收方接到 indication (指示) 消息，马上接收方发出 response (反馈) 消息，最后，发起方收到 confirm (确认) 消息。这个过程做一些系统参数配置。

3.1.3. 建立 DLC 服务

DLCI = 1-63 (DLCI number)

System parameters= Type of frame [UIH | UI | I, default: UIH],

Convergence layer [1 - 4, default: 1]

Priority [0-63]

Acknowledgement Timer [0,01s-2,55s,
default: 0,1s]

Maximum Frame Size [1 –32768,
default: 31 for the basic option and
64 for the advanced option]

Maximum number of retransmissions [0 –255,
default : 3]

Window size for error recovery mode [1 –7,
default : 2]

Accept= [true | false]

建立服务，是配置各个通道的参数，让通道能工作。它需要 4条消息：

- 1) 发起方： TS_0710_DLC_ESTABLISHMENT.request(DLCI, system_parameters)
- 2) 发起方： TS_0710_DLC_ESTABLISHMENT.indication(DLCI, system_parameters)
- 3) 接收方： TS_0710_DLC_ESTABLISHMENT.response(DLCI, system_parameters, accept)
- 4) 接收方： TS_0710_DLC_ESTABLISHMENT.confirm(DLCI, system_parameters, accept)

大致的过程和启动服务的流程差不多。 建立服务是建立通道， 当然是配置通道参数， 后者是配置系统参数。

3.1.4. 数据服务

数据服务使用了两条消息：

- 1) TS_0710_DATA.request(DLCI, User_data)
- 2) TS_0710_DATA.indication(DLCI, User_data)

DLCI = [1 –63] DLC over which the data is to be transmitted.

User_data= Data to be transferred organised in accordance with the
convergence layer of the DLC

3.1.5. 功耗控制

功耗控制包括 sleep模式和 wake up模式

3.1.5.1. Sleep 模式

有三条消息：

- 1) TS_0710_SLEEP.request
- 2) TS_0710_SLEEP.indication
- 3) TS_0710_SLEEP.confirm

注意的是这里不需要接收方发出反馈（Response）消息，当然有这种情况，就是接收方没有进入 Sleep 模式，也被认为是进入 Sleep 模式。

3.1.5.2. Wake up 模式

有两条消息：

- 1) TS_0710_WAKEUP.indication
- 2) TS_0710_WAKEUP.response

3.1.6. 释放 DLC 服务

用来断开 DLC 通道，有两条消息

- 1) TS_0710_DLC_RELEASE.request(DLCI)
- 2) TS_0710_DLC_RELEASE.indication(DLCI)

参数：

DLCI = [1 -63] Number of the DLC to be released.

同建立通道一样，最大的可以有 63 个通道。

3.1.7. 关闭服务

两条消息：

- 1) TS_0710_CLOSE.request
- 2) TS_0710_CLOSE.indication

关闭 CMUX 服务，和启动 CMUX 对应。

3.1.8. 控制服务

这里把控制服务分成独立的一段。主要是控制服务比较复杂。分作两部分：07.10 协议服务和 Port Emulation 服务。

3.1.8.1. 07.10 协议服务

DLC 参数判定

DLC 参数判定用来设置具体 DLC 参数，有四个消息；

- 1) TS0710_PARNEG.request (DLC, DLC parameters)
- 2) TS0710_PARNEG.indication (DLC , DLC_parameters)
- 3) TS0710_PARNEG.response (DLC , DLC_parameters , accept)
- 4) TS0710_PARNEG.confirm (DLC, DLC _parameters, accept)

用来改变一个具体的 DLC 通道的参数。

DLC_parameters = frame type [UIH | UI | I ,

default: UIH]

Convergence Layer Type [Type 1 | Type 2 | Type 3 | Type 4,

default: Type 1]

Priority [1-63,

default: according to table in subclause 5.6]

Acknowledgement timer [10 ms - 25.5 sec,

default: 100 ms]

Maximum Frame Size [1 –32768,

default: 31 for the basic option and

64 for the advanced option]

Maximum number of retransmissions [0 –100,

default : 3]

Response timer for the multiplexor control

channel [0,01s-2,55s, default: 0,3s]

Wake up response timer [1s –255s, default 10s]

Window size for error recovery mode [1 –7,

default : 2]

accept = [true | false]

DLC 服务判定服务

用来判别 DLC 服务是那一项具体的服务。

- 1) TS0710_SERVNEG.request (DLC, Service_parameters)
- 2) TS0710_SERVNEG.indication (DLC , Service_parameters)
- 3) TS0710_SERVNEG.response (DLC , Service parameters , accept)
- 4) TS0710_SERVNEG.confirm (DLC, Service_parameters, accept)

service_parameters = **Service**[data | voice 64kbit/s A-law PCM | reserved 1 | reserved 2],

voice codec[GSM 06.21 | 64kbit/s u-law PCM | coded ADPCM 32kbit/s | coded half rate |

128 kbit/s PCM | reserved]

测试服务

- 1) TS0710_TEST.request (Test data)
- 2) TS0710_TEST.confirm (Test data)

流程控制

- 1) TS_0710_FLOW.request(DLCI,State)
 - 2) TS_0710_FLOW.indication(DLCI, State)
- DLCI = [1 –63] DLC over which the data is to be transmitted.
- State = enabled (data may be transferred), disabled (data may not be transferred)
- 控制使能或者不使能数据传输。

3.1.8.2 虚拟端口服务

移动DLC 参数判定服务

- 1) TS0710_PORTNEG.request (DLC, Port_parameters)
- 2) TS0710_PORTNEG.indication (DLC , Port_parameters)
- 3) TS0710_PORTNEG.response (DLC , Port parameters , accept)
- 4) TS0710_PORTNEG.confirm (DLC, Port_parameters, accept)

port_parameters= Port speed [2,4 | 4,8 | 7,2 | 9,6 | 19,2 | 38,4 | 57,6 | 115,2 | 230,4 kBit/s],

Data bits [5 | 6 | 7 | 8,
default: 8 bits |

Stop bits [1 | 1,5,
default: 1 bit |

Parity [no parity | parity,
default: no parity |

Parity Type [odd | even | mark | space]

accept= [true | false]

控制参数服务

- 1) TS0710_CONTROL.request (DLC, Control_parameters)
- 2) TS0710_CONTROL.indication (DLC , Contol_parameters)
- 3) TS0710_CONTROL.response (DLC , Contro_parameters)
- 4) TS0710_CONTROL.confirm (DLC, Control_parameters)

system_parameters= Modem Signal [DTR/DSR | RTS/CTS | RI | DCD],

Break Signal [0—3 s in steps of 200 ms,
default 0ms],

Buffers [do not discard buffers, discard buffer
default: do not discard buffers],

Break signal sequence[as soon as possible | in sequence,

default: in sequence]

DLC 通道指示服务

- 1) TS0710_PORTNEG.request (DLC , Line Status parameter)
- 2) TS0710_PORTNEG.indication (DLC , Line Status parameter)

Line status parameter= Port speed [no errors, overrun error, parity error, framing error]

3.2. 数据帧结构

3.2.1. 帧域

Flag(1B) Address (1B) Control(1B) Length(1~2B) Info(Length 长度, 可变) FCS(1B) Flag(1B)
注意：数据长度指示域, 可能有, 可能没有, 如果没有采用透明机制, 则有数据长度指示域。

3.2.1.1. 标志域 (Flag Sequence Field)

每一帧开始和结束都有一个标识 flag , 基本模式 Flag固定为 0xF9。高级模式为 0x7E。

3.2.1.2. 地址域 (Address Field)

Bit No.	1	2	3	4	5	6	7	8
Signal	EA	C/R	DLCI					

图 2：地址域的格式

DLCI 是 DLC (Data Link Control 数据链路控制) 的标识号 (identify)。
EA 位表示该域是否只有本字节。为 1 表示本字节就是长度域, 为 0 表示后续还有一个字节。这里只能为 1, 以后扩展协议后, 会有为 0 的情况。
C/R (Command/Response) 用来指示是命令还是响应。 Initiator 是指 MUX 初始化的那一方, 而 responder是响应 MUX 初始化的那一方。

表 1：C/R 位的用法

Command/response	Direction	C/R value
Command	Initiator -> Responder	1
	Responder -> Initiator	0
Response	Initiator -> Responder	0
	Responder -> Initiator	1

如主机建立的 MUX, 主机发命令包时, Address为 0x7, 猫响应 Address也是 0x7。

3.2.1.3. 控制域 (Control Field)

表2：控制域的编码

Frame Type	1	2	3	4	5	6	7	8	备注
SABM (Set Asynchronous Balanced Mode)	1	1	1	1	P/F	1	0	0	
UA (Unnumbered Acknowledgement)	1	1	0	0	P/F	1	1	0	
DM (Disconnected Mode)	1	1	1	1	P/F	0	0	0	
DISC (Disconnect)	1	1	0	0	P/F	0	1	0	
UIH(Unnumbered Information with Header check)	1	1	1	1	P/F	1	1	1	
UI (Unnumbered Information)	1	1	0	0	P/F	0	0	0	可选

P/F是 Poll/Final 位，表示测试 /返回。后续会介绍。

比如建立 DLC 的时候，主机发 SABM 帧，并把 P置 1，如果成功，对方返回 UA 帧，并把 F置 1，否则返回 DM 帧，并把 P置 1。

3.2.1.4. 信息域 (Information Field)

信息域，它是实际数据域。 UI 帧或 UIH 帧，才有该域。

3.2.1.5. 长度指示域 (Length Indicator)

只有在基本模式下，该域才存在。可以是 1 个字节，也可以是 2 个字节长度；

Bit No.	1	2	3	4	5	6	7	8
Signal	E/A	L1	L2	L3	L4	L5	L6	L7

图3：长度域的首字节

Bit No.	1	2	3	4	5	6	7	8
Signal	L8	L9	L10	L11	L12	L13	L14	L15

图4：长度域的第二个字节

EA 位表示该域是否只有本字节。为 1 表示本字节就是长度域，为 0 表示后续还有一个字节。

3.2.1.6. 校验域 (Frame Checking Sequence Field)

帧的校验序列。它是两个 CRC校验码，异或后，再取反的结果。这两个 CRC校验码，都是8位的，生成的多项式为 $x^8 + x^2 + x + 1$ 。一个是 $x^k \cdot (x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$ ，其中 k是校验数据的 BIT 位的数目，一个是 $x^8 \cdot (校验数据)$ ，校验数据，就是该帧的实际数据。由于比较复杂，可以参考 CRC校验原理。在包为 UIH 帧时，实际数据域，不参与校验，只校验 Address，Control 和 Length这几个域。

3.2.2. 格式约定

所有传输的字符以 “ 一个起始位， 八个数据位， 无奇偶校验位和一个停止位 ” 进行传输。
所有的域，先传输域的第 1 位。对于字节而言，也就是低位先传。
地址，命令，响应和序列数，都先传低位。
FCS序列，先传高次方的项的系数。
注意：这些约定意味着常常需要转换 BIT 位顺序。

3.2.3. 帧有效性

检验数据传输是否正确，可以根据下面的条件判断：

- 1) is not properly bounded by two flags 不正确的边界条件位；
- 2) 在两个边界条件位 (flags) 之间，至少有三个字节。
- 3) FCS的 CRC校验错误；
- 4) 地址域大于一个字节。

无效帧，通常会在不通知发送方的情况下被丢弃。 Multiplexer 的 MS 或 TE 端接收到无效帧后，所要做的处理， 由它的实现者去考虑。 然而， 接收到 FCS校验域错误的帧的提示信息，对音频 /视频的 DLC 或许很有用。

3.2.4. 帧中止

不支持帧的中止。

3.2.5. 数据帧之间的填充

除了省电模式唤醒的过程，在数据帧与数据帧之间，需要填充一些 stop 标识。这种填充了标识字符的情况下，接收方应该能正确处理。如果接收端，收到了多于 3 个连续的标识，它在第一时间开始传输连续的标识（参见省电部分）。

3.2.6. 基本 Basic

Flag	Address	Control	Length Indicator	Information	FCS	Flag
1B	1B	1B	1B/2B	长度不确定	1B	1B

图 5：basic 模式的帧的结构

在基本模式下的 flag 为下面的格式： =0xF9

Bit	1	2	3	4	5	6	7	8
Data	1	0	0	1	1	1	1	1

图 6：basic 模式的标识域

3.2.6.1. 约束

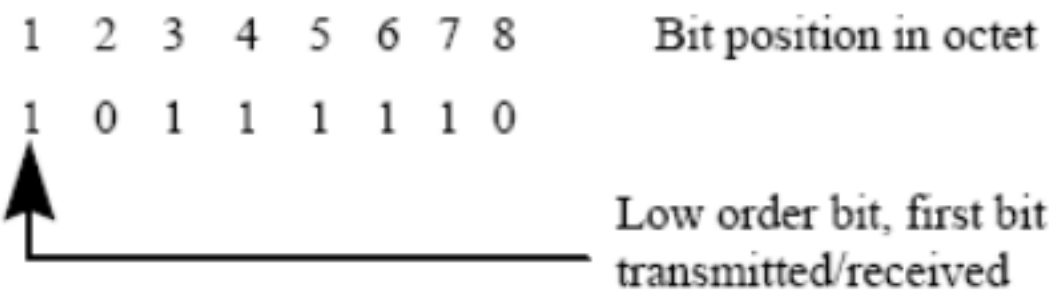
- 1) 结束 flag 也可能是下一帧的开始标志。
- 2) 在高级的模式中 flag 不一样。
- 3) 不支持 DC1/XON 和 DC3/XOFF 的控制字符。

3.2.7. 高级 Advanced

如果 MUX 会话开始，就使用了高级模式，那么，它就可以用于各种数据帧了。这种机制基于控制字节透明（a control octet transparency）。它基于每帧的开（Opening）和闭（Closing）的Flag标志的出现是唯一的。 这些标志不可能出现在数据帧的信息域。 如果在 TE-MS 链路上，出现了同步的丢失，这种机制允许快速的同步恢复。

3.2.7.1. 控制字节透明

下面的透明机制应用于每个数据帧，从地址域到 FCS 域（含）。 ESC 控制字节，是一个透明标识符，它标识了应用了下面透明流程的数据帧的一个字节。 ESC 控制字节编码如下：



传输器（transmitter）将检查从开标识到闭标识的一个数据帧，包括地址域，控制域，FCS 域。FCS 域是这样计算的：

- 当出现了标识或 ESC 控制字节时，补足该字节的第 6 位，并且
- 在传输的前一个字节与本字节中间，插入一个 ESC 控制字节（Insert a control escape octet immediately preceding the octet resulting from the above prior to transmission）。

接收器（receiver）将检查两个标识字节之间的数据帧，收到一个 ESC 控制字节，并且预先进行 FCS 计算：

- 丢弃 ESC 控制字节，并且
- 通过补足它的第 6 位，恢复紧接着的字节。

其他字节值也可通过传输器，包含在透明流程中。 因此，包含哪些东西，取决于前述的系统和应用程序的约定。

3.2.7.2. 开始/停止传输 -扩展透明

传输器（transmitter）对追加到标志域和 ESC 控制字节上的其他字节应用上述的透明流程。目前，仅有的其他字节是流控（flow-control）字符。流程参见 3.2.6.3。

3.2.7.3. 流控 (Flow-control) 透明

流控 (Flow-control) 透明选项，为 ISO/IEC646 （例如 1000100x 和 1100100x 分别表示 x可以是 0或者 1 ）DC1/XON 和 DC3/XOFF 控制字符，提供了透明的流程。 这就确保了字节流不包含像流控字符（不管奇偶校验）那样，能被中间设备打断的数值。

3.2.7.4. 帧的结构

帧的结构如图 7所示。注意，这种结构不包括同步（例如，开始和停止位）或者透明目的所增加的信息。传输的顺序从左往右。

透明机制被激活的情况下，帧的结构如下所示：

Flag	Address	Control	Information	FCS	Flag
1B	1B	1B	长度不确定	1B	1B

图 7：advanced 模式的帧结构

在高级模式下的 flag 为下面的格式： =0x7E

Bit	1	2	3	4	5	6	7	8
Data	0	1	1	1	1	1	1	0

图 8：advanced 模式的标识域

注意：结束 flag 也可能是下一帧的开始标志。

3.3. 帧类型

- 1) Set Asynchronous Balanced Mode (SABM) command
- 2) Unnumbered Acknowledgement (UA) response
- 3) Disconnected Mode (DM) response
- 4) Disconnect (DISC) command
- 5) Unnumbered information with header check (UIH) command and response
- 6) Unnumbered Information (UI) command and response

3.3.1. SABM command

SABM 命令帧用来把指定的一端设置成异步平衡模式（ ABM ），这种模式下，所有的控制域的长度都是一个字节。另外一端接收到 SABM 命令帧后，第一时间发送一个 UA 回应帧，作为确认。接收到该命令， DLC 发送和接收的状态，都将被置 0。

3.3.2. UA response

UA 回应帧，是对 SABM 和 DISC 这两个命令帧的确认。

3.3.3. DM response

它用于报告一端（ station ）逻辑上从数据链路上断开的状态。链路断开后，将收不到任何命令，直到收到 SABM 重新建立通道为止。如果在链路断开状态，收到 DISC 命令，就应该发一个 DM 作为响应。

3.3.4. DISC command

DISC 命令用于终止通道。一端用它通知另外一端，它正在挂起操作并且将进入逻辑断开模式。在处理这个命令以前，接收一端，用 UA 命令去回应，作为对收到 DISC 命令的确认。

用 DLCI 0 通道传送的 DISC 命令，跟 MUX 的关闭命令（见 3.4.6.3 ）具有相同的意思。关于关闭流程的更多信息，参加 3.8.2 。

3.3.5. UIH command/response

UIH 命令帧 / 响应帧，用来传递不带有能影响任何一端的 $V(S)$ 和 $V(R)$ 变量的信息。被传输的信息的完整性与传输到正确的 DLCI 相比，不太重要时，可以用 UIH。UIH 的帧，FCS 只对地址域，控制域和长度域，进行计算。

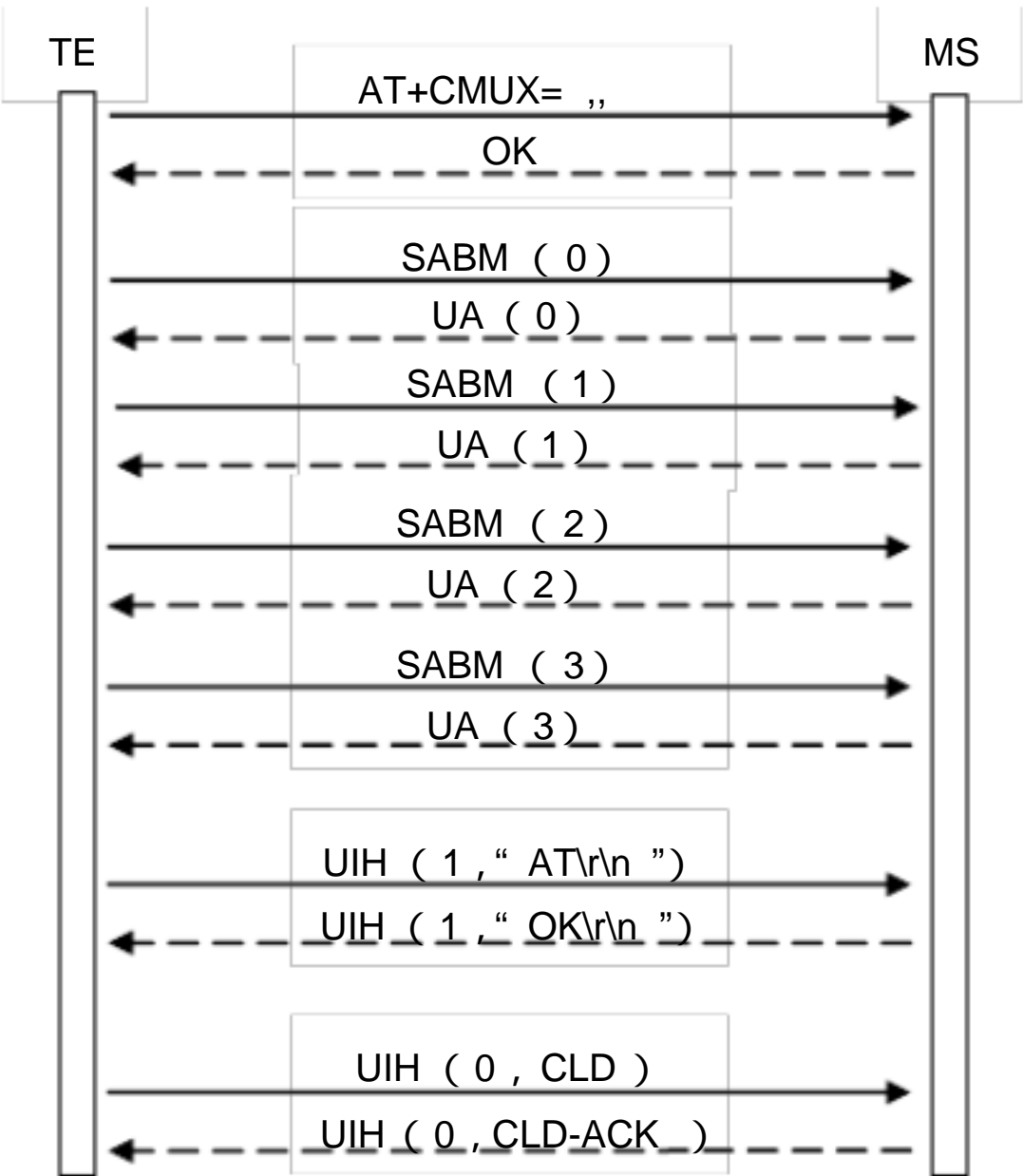
UIH 命令帧 / 响应帧的接收，不是顺序的编号（被数据链路流程检验）。因此，假如数据链路在受保护的该命令部分的传输过程中发生了异常，UIH 数据帧可能丢失；假如对该命令的回复过程中出现了异常，UI 数据帧可能会重复。UIH 命令帧 / 响应帧，没有指明回应帧。

3.3.6. UI command/response

UI 命令帧 / 响应帧，用来传递不带有能影响任何一端的 $V(S)$ 和 $V(R)$ 变量的信息。UI 命令帧 / 响应帧的接收，不是顺序的编号（被数据链路流程检验）。因此，假如数据链路在受保护的该命令部分的传输过程中发生了异常，UI 数据帧可能丢失；假如对该命令的回复过程中出现了异常，UI 数据帧可能会重复。UI 命令帧 / 响应帧，没有指明回应帧。

对 UI 数据帧来说，FCS 对所有域（地址域，控制域，长度域，信息域）进行计算。是否支持 UI 数据帧，是可选的。

3.4. 过程和状态



3.4.1. 建立 DLC 链路

多数情况，DLC 链路是由 TE 建立的，然而，协议上是对等的，MS 也可以建立它，但本文不叙述这种情况。

建立过程如下：首先发起方发送 P 位为 1 的 SABM 命令，地址域是通道的编号。响应方发 F 位为 1 的 UA 作为回应。如果响应方没准备好，或者不愿意建立，那么，它就回一个 F 位为 1 的 DM 帧。

一旦 DLC 建立成功，就意味着进入了连接状态，可以传输数据了。

如果 T1 时间以内，收不到 UA 或 DM 的响应，就重发 SABM。这个动作可以一直重复，直到得到响应，或者上层应用采取了其他措施。

如果没有协商过程，DLC 的参数就是默认的。

3.4.2. 释放 DLC 链路

双方都可以发送 P 位为 1 的 DISC 命令，来释放 DLC 链路。另外一方回应 F 位为 1 的 UA 帧。如果释放成功，DLC 就进入断开状态。

如果收到 DISC 命令时，发现已经是断开状态了，就发一个 DM 作为回应。

如果 T1 时间内，没有收到 UA 或者 DM 的响应，就重发 DISC 命令。这个动作可以一直重复，直到得到响应，或者上层应用采取了其他措施。

3.4.3. 信息传输

3.4.3.1. 信息数据

信息使用 UI 或 UIH 进行传输。UIH 一定要支持，UI 是可选的。

当知道数据被正确的接收非常重要时，就采用 UI，例如 IP 业务包。传输出错无所谓时，使用 UIH。由于及时性的关系，不能采用差错恢复的机制时，可以使用 UI 或 UIH，比如音频数据。

双方都要把 P 位置 0。C/R 位，参见上面的描述。

信息的最大长度，受系统的参数所限制。

3.4.3.2. 优先级

每个数据流，都有对应的优先级。总共有 0-63 级别的优先级，数字越小，表示优先级越高。TE 位每个 DLC 指定一个优先级，并且，用控制通道，通知 MS。

如果一条消息缺省优先级，DLC 将根据 3.6 中的 DLCI 分配表指定它的优先级。传输器 (transmitter) 控制哪些帧要传输，以及对它们按何种结构进行封装，并且不描述这些工作是如何完成的。假如比现在正在传输的数据的优先级更高的数据正在等待，传输器有如下几个可以采用的选项：

- a) 先传输完当前帧。
- b) 中止对当前帧的打包，传输当前的 FCS 和终止标识 Flag (只对 advanced)，并且开始传输更高优先级的数据。

带有更高优先级数据的 DLC，不会阻止任何低优先级的 DLC。高优先级和低优先级的帧的交叉，是必要的，因为这能避免低优先级通道被永远堵塞。

3.4.4. 帧变量

P 位置 1，表示通道的一端向另外一端请求一个或者一组回应 (response)。F 位置 1，表示通道一端指明发一个响应帧，对另外一端的 P 命令进行回应。

在命令帧和响应帧里，P/F 位，最好用一个函数进行设置。

3.4.4.1. P 位函数

P 位置 1，向另外一端请求 F 位置 1 的响应帧。

在一个特别的 DLCI，在给定的时间和给定的方向，只有一个 P 位置 1 的帧。

接收到一个 P 位置 0 的 SABM 或者 DISC 命令帧，接收到的帧，将被丢弃。

如果接收到未被请求的 DM 响应，这个帧要做与 P/F 设置无关的处理。

通道的一端在发送另外一个 P 位置 1 的帧之前，它会从另外一端收到一个 F 位置 1 的回应帧。如果在系统定义的时间段以内，没有收到回应帧，重发一个 P 位置 1 的帧，用于错误恢复的目的，是允许的。

3.4.4.2. F 位函数

F位置 1 的回应帧，一端回应另外一端的 P位置 1 的命令帧时，使用它。要求在第一时间回应。

一端可以随时异步发送 F位置 0 的回应帧。但如果接收到的是 F位置 0 的UA 回应帧，接受的帧，将被丢弃。

如果接收到未被请求的 DM 响应，这个帧要做与 P/F设置无关的处理。

假如一端接收到一个 P位置 1 的命令帧，F位置 1 的回应帧的传输，将优先于其他命令。但模式设置命令（ SABM 或者 DISC ）是个例外。

3.4.5. 超时的考虑

为了检测无回应或者回应丢失的情况，每端都应该提供一个响应超时的函数（ T1 ）。超时后，应该启动错误恢复的流程。

为了解决争夺的情况，超时函数的时长，在两端最好不要相等。

一端传输过一帧，等待回应时，超时函数就应该启动。收到回应，则停止计时。

如果响应超时函数超时了，重发 P位置 1 的命令帧，超时函数重新开始计时。

3.4.6. 多路控制通道

在TE和MS 通信开始时，使用 3.8.1 的流程，建立了 DLCI 0 通道作为控制通道。它用来在两个 MUX （ multiplexer ）之间传递信息。它用 ERM 模式或者 non-ERM 模式的流程，这个在用+CMUX 命令的时候，就决定了的。如果 ERM 的流程是可用的，就用它们。

3.4.6.1. 消息格式

Type	Length	Value1	Value2	...	Value n
------	--------	--------	--------	-----	---------

所有在多个 MUX 之间被传递的信息，都使用上述的 type，length，value 的格式。

上述的每个单元格至少占用 1 个字节，type 和 length 有扩展位，所以，它们有可能多于 1 个字节。

type 域的第一个字节的 bit 位格式如下：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C\R	T1	T2	T3	T4	T5	T6

考虑 EA=1 时，长度域的扩展字节的 bit 位格式如下：

Bit No	1	2	3	4	5	6	7	8
Value	EA	T7	T8	T9	T10	T11	T12	T13

EA 位就是扩展位，其值为 1 表示当前就是本域的最后一个字节。

C/R 位指示该信息是命令帧，还是回应帧。

T位表示类型编码。 每个命令都有自己独特的编码方法。 $2^6=64$ ，这就是说一个字节的 type 域，能表式 63 种不同的信息类型。单字节的信息类型，在本文档中已经定义。

length的第一个字节的 bit 位格式如下：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	L1	L2	L3	L4	L5	L6	L7

后续字节，也是类似的格式。 EA的意义跟上面的相同。

L1 是最低位， L7 是最高位。 $2^7=128$ ，也就是一条信息最大可以有 127 个字节的 value 值。

在不超过最大长度的情况下， 一帧可以包含多条信息。 但一条信息不能分割封装在多帧中。

3.4.6.2. 操作流程

消息都是成对存在的，一个命令消息和一个相应的回应消息。假如 C/R 位置为 1，消息是命令，假如它置 0，消息是回应。 一个回应消息， 跟引发它的命令一样， 拥有同样的 T 位。

假如在 T2 的时间内，一个命令没有产生一个回应，命令将被重新发送 N2 次。假如 N2 次的发送，都没有收到回应， MUX（multiplexer）的控制通道被认为有错，并且发出一个警报。错误情况的解决方法，相应的执行。

3.4.6.3. 消息类型和处理（ Action ）

3.4.6.3.1. 参数协商 PN（Parameter Negotiation）

这个流程是可选的。如果这个命令不支持， DLC 都使用默认值。

在一个 DLC 用 3.4.1 的机制建立之前， TE 和 MS 必须对本 DLC 使用的参数进行协商。 这些参数都由 PN 决定。

PN使用下面的 type 域字节：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C\R	0	0	0	0	0	1

Length 域字节包括值 8和下面 8个值字节。值字节包括表 3的信息。

表 3：PN

Value Octet	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8
1	D1	D2	D3	D4	D5	D6	0	0
2	I1	I2	I3	I4	CL1	CL2	CL3	CL4
3	P1	P2	P3	P4	P5	P6	0	0
4	T1	T2	T3	T4	T5	T6	T7	T8
5	N1	N2	N3	N4	N5	N6	N7	N8
6	N9	N10	N11	N12	N13	N14	N15	N16
7	NA1	NA2	NA3	NA4	NA5	NA6	NA7	NA8
8	K1	K2	K3	0	0	0	0	0

各个域的编码如下：

D位 定义了其他信息提及的 DLCI，D1 是最低有效位。

I位 定义了用来在特殊 DLC 装载信息的 数据帧的类型。见表 4。

表 4：I 位的含义

Meaning	I1	I2	I3	I4
Use UIH frames	0	0	0	0
Use UI frames	1	0	0	0
Use I frames (note)	0	1	0	0

其他值保留。默认值是 0000。缺省协商的情况下，（为 DLCI>0）使用的帧类型，跟 MUX 控制通道使用的相同。

CL 位定义了特殊 DLC 使用的集成层的类型。参见表 5。

表 5：CL 位的含义

Meaning	CL1	CL2	CL3	CL4
Type1	0	0	0	0
Type2	1	0	0	0
Type3	0	1	0	0

其他值保留。默认值是 0000。

P 位定义了指派给特殊 DLC 的优先级。范围是 0-63，0 是最低的优先级。P1 是最低有效位。默认值由 DLCI 的值给定，参见 3.6。

T 位定义了确认时间 T1，参见 3.7.1。单位是毫秒，T1 是最低有效位。

N 位定义了最大帧的大小 N1，参见 3.7.2。该参数是六位的数字，N1 是最低有效位。

NA 位定义了重发的最大次数 N2，参见 3.7.3。该参数是八位数字，K1 是最低有效位。

TE 向 MS 传输一个参数协商（PN）命令，该命令的域里，赋值成 TE 想要为特殊 DLCI 使用的值。MS 回应一个参数协商（PN）回应，该回应带有 MS 提议的值。MS 构建回应时，参照下面的规则：

- DLCI 值不变。
- 使用 I 帧或 UI 帧，是可选的，因此，没实现 UI 帧或 I 帧，MS 采用 UIH 帧回应。
- MS 不会改变被 TE 提议的集成层。
- MS 不会改变被 TE 提议的优先级。
- T1 值是 TE 使用的，是不可协商的；MS 将插入它自己的 T1。建议每方使用不同的 T1。
- 假如没有足够的内存，MS 将为最大帧的大小 N1，提议一个更小的值。
- N2 是 TE 使用的，是不可协商的；MS 将插入它自己的 N2 值。
- 假如没有足够的内存，MS 将为窗口大小 k，提议一个更小的值。

假如 TE 认为 MS 的回应是可以接受的，TE 将根据 3.3.1 的流程，建立 DLC。假如回应是不可接受的，TE 将发起另一个带有新参数的 PN 命令，或者向上层传递失败信息。

假如网络上有一个来电到达 MS，这时没有已经建立好的 DLC，MS 将发起 PN 流程，并建立一个 DLC。这种情况实际上是不会发生的，因为 TE 通常为各种功能建立 DLC，在容量交换后，它与 MS 所共享。来电的提示，将抛出（be through？）一个 07.07 或者 07.05 的结果代码（result code）。

3.4.6.3.2. 省电控制 PSC（Power Saving Control）

参见 3.4.7。省电控制消息使用如下的 type 域字节：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C\R	0	0	0	0	1	0

Length 字节包含值 0，没有无值字节。

假如一端想进入省电状态，它传输一个 PSC 命令；另一端回复一个 PSC 回应。

假如一端想请求让另外一端进入省电状态，它传输一个 PSC 命令；另外一端回复一个 PSC 回应。响应端可以进入省电状态，但这不是必须的。

3.4.6.3.3. 关闭 MUX (Multiplexer Close Down)

参见 3.8.2。
MUX 的 CLD 命令用于重置链路为非多串口的正常 AT 命令模式。它的 type 域字节如下：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C/R	0	0	0	0	1	1

Length 字节包含 0 值，没有值 (value) 字节。

3.4.6.3.4. 测试命令

测试命令用来测试 TE 和 MS 的链接。Length 字节描述了值的字节数量，它用于验证模型。对端将回应具备相同的值的字节。它的 type 域字节的格式如下：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C\R	0	0	0	1	0	0

3.4.6.3.5. 流控开 FCon (Flow Control On)

流控命令用来处理聚集的流 (aggregate flow)。当任何一端可以接收新信息的时候，它发送这个命令。

Length 字节包含值 0，没有值 (value) 字节。它的 type 域字节如下：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C\R	0	0	0	1	0	1

3.4.6.3.6. 流控关 FCoff (Flow Control Off)

流控命令用来处理聚集的流 (aggregate flow)。当任何一端不能接收新信息的时候，它发送这个命令。另外一端不能发送数据帧，除了用控制通道 (DLC=0)。

Length 字节包含 0 值，没有值 (value) 字节。它的 type 域字节如下：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C/R	0	0	0	1	1	0

3.4.6.3.7. 猫状态命令 MSC (Modem Status Command)

想往一个数据流中传送 V.24 控制信号，这是通过发送 MSC 命令来完成的。MSC 命令有一个强制的控制信号字节和一个可选的中止信号字节。当 basic 模式时，这个命令才有效 (relevant)。

这个命令将在 DLC 建立之后，优先于任何用数据，进行传输。

Command	Length	DLCI	V.24 signals	Break Signals(optional)
---------	--------	------	--------------	-------------------------

这个命令包含值 2 或值 3，并且有 2 或 3 个值字节。

DTE 和 DCE 都使用这个命令，相互通知它们自己 V.24 控制信号的状态。MSC 命令的长度是 4 或 5 字节，这依赖于中止信号。

Command 域字节的格式如下：

Bit No.	1	2	3	4	5	6	7	8
Value	EA	C\R	0	0	0	1	1	1

C/R 位用来指示它是命令还是回应。

每当信号改变，DTE 或 DCE 发送这个命令，来指明每个信号的当前的状态。当 DTE 或 DCE 接收到猫命令，它总是发送回一个回应。接收方或发送方的 V.24 信号到控制信号字节里的 BIT 位的映射表，分别在表 6 和表 7 中给出。

MSC 命令中，发送方自己的 V.24 信号，将被发送；但在回应中，它复制了从将被送回

的命令帧里，接收到的 V.24 信号。

DLCI 域标识了该命令应用的 (specific) DLC。第 2 位总被置 1，EA 位根据 3.2.1.2 的描述设置。

Bit No.	1	2	3	4	5	6	7	8
Value	EA	1	DLCI					

图9：地址域的格式

紧跟着 DLCI 域的是控制信号域，它包含信号状态的表现 (representation)，见图 10。扩展位的使用，允许添加其他字节，以满足其他情况。目前，可选的第二个字节被定义为用来处理中止信号的传输。

Bit No.	1	2	3	4	5	6	7	8
Value	EA	FC	RTC	RTR	Reserved0	Reserved0	IC	DV

图10：控制信号字节的格式

控制信号字节的描述如下：

EA 位就是扩展位，其值为 1，表示当前就是本域的最后 一个字节。

FC (Flow Control) 流控位，其值为 1，表示不能接收数据帧。

RTC (Ready To Communicate) 位。其值为 1，表示设备准备好发送数据。

RTR (Ready To Receive) 位。其值为 1，表示设备准备好接收数据。

Bit5/Bit6，保留位，以后使用。发送方置 0，接收方忽略它。

IC (Incoming Call) 位。其值为 1，表示一个来电。

DV (Data Valid) 位。其值为 1，表示有效数据正在发送。

控制字节映射到 V.24 信号，根据下表：

表6：接收 时转 V.24 信号的映射表

Control Signal Byte	DTE receiving		DCE receiving	
bit number, name	signal V.24	circuit	signal	V.24 circuit
3, RTC	DSR	107	DTR	108/2
4, RTR	CTS	106	RFR (note)	133
7, IC	RI	125	-ignored	-
8, DV	DCD	109	-ignored	-

NOTE Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.

表7：发送时转 V.24 信号的映射表

Control Signal Byte	DTE sending		DCE sending	
bit number, name	signal	V.24 circuit	signal	V.24 circuit
3, RTC	DTR	108/2	DSR	107
4, RTR	RFR (note)	133	CTS	106
7, IC	always 0-	-	RI	125
8, DV	always 1-	-	DCD	109

NOTE Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.

假如一端由于流控的原因，不能传输数据帧，但它想自己停止接收更多的帧，它为了发送流控信号，可以继续发送不包含用户数据的帧（比如，只有控制信号字节和可选的中止信号字节）。

EA 位就是扩展位，其值为 1，表示当前就是本域的最后 一个字节。

Bit No	1	2	3	4	5	6	7	8
signal	EA	B1	B2	B3	L1	L2	L3	L4

图 11：中止信号字节的格式（可选的）

中止信号字节通过数据流能携带被 DLC 检测到的中止信号的信息。每个位的含义如下表：

表 8：中止信号字节的含义

Bit	Value	Meaning
B1	1	字节编码了一个中止信号
	0	字节没编中止信号
B2	0	保留，发送方置 0，接收方忽略
B3	0	保留，发送方置 0，接收方忽略
L1-L4	4-bit value	Length of break in units of 200ms

L1-L4 是中止的长度（ break length ）， L1是最低有效位， L4 是最高有效位。

当一端接收到一个中止字节， 它将处理接收到的信息， 并且通过适当的途径传递它。 这不在本文档的描述范围内。

3.4.6.3.8. 不支持命令 NSC（ Non Support Command）

当接收端不支持某种类型的命令时，发该命令帧作为回应。

Length 字节不含值 1，并且只有一个值字节。

Type域字节格式如下：

Bit No.	1	2	3	4	5	6	7	8
signal	EA	C\R	0	0	1	0	0	0

值字节包含不支持命令的命令格式。

值字节（ value octet ）格式如下：

Bit No.	1	2	3	4	5	6	7	8
signal	EA	C\R	Command type					

3.4.6.3.9. 远程端口协商 RPN（ Remote Port Negotiation ）

这个命令是可选的。这个命令用来配置远程端口通信设置的。

所以的设备都要保证在发送数据前， 通信设置被正确的配置。 如果协商没完成， 将对所有参数采用默认值。

在连接状态，当通信设置变动后，设备必须发送 PRN。端口线路状态（ Port Line Status ）变动后，也一样要发 PRN。

Command	Length	Value	Value	Value	Value	Value	Value	Value	Value
RPN	1 or 8	octet1	octet2	octet3	octet4	octet5	octet6	octet7	octet8
		optional	optional	optional	optional	optional	optional	optional	optional
		(DLCI)							

RPN的type域字节格式如下：

Bit No.	1	2	3	4	5	6	7	8
signal	EA	C\R	0	0	1	0	0	1

Length 字节包含值 1或8，并且它有 1个或 8个值字节。

DLCI 字节格式如下：

Bit No.	1	2	3	4	5	6	7	8
signal	EA	1	DLCI					

DLCI 字节的第 2 位没用，固定为 1。EA 位，依据 3.2.1.2 的描述。 Value 数组：端口值字节（ Port Value Octet）格式如下：

Value Octet	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8
2	B1	B2	B3	B4	B5	B6	B7	B8
3	D1	D2	S	P	PT1	PT2	res	res
4	FLC1	FLC2	FLC3	FLC4	FLC5	FLC6	res	res
5	XON1	XON2	XON3	XON4	XON5	XON6	XON7	XON8
6	XOF1	XOF2	XOF3	XOF4	XOF5	XOF6	XOF7	XOF8
7	PM1	PM2	PM3	PM4	PM5	PM6	PM7	PM8
8	PM9	PM10	PM11	PM12	PM13	PM14	PM15	PM16

一端向另外一端发送一条 RPN 命令，它的各个域，都设置了想要的值的，参数掩码指明了哪些参数被设置了值。

当另一端接收到 RPN 命令后，将根据下面的规则，作回应：

DLCI 值不能变。

接收方将接收到发送方提议（ proposed）的端口值字节，并且回复一个把所有能接受参数的参数掩码置 1 的数据帧。假如接收方不接受任何提议的值，它回复一个把不支持参数的参数掩码都置 0 的帧。对这些参数掩码置 1 的参数，新的值，将被接受并且被使用。

假如命令里只包含了一个值字节，它被解释位一个请求，接收方将用当前的端口值的设置，进行回应。

假如发送方认为回应是可以接受的，也就是说，各比特位相匹配（ match），发送方将开始根据端口值字节使用 DLC。假如回应是不可接受的，发送方将启动另一个参数修改了的 RPN 命令，直到最终协商好，或者向更高层传递失败的信息。

B1-B8 指明了波特率，见下表。

表 12：B 位的含义

Meaning	B1	B2	B3	B4	B5	B6	B7	B8
2400bit/s	0	0	0	0	0	0	0	0
4800bit/s	1	0	0	0	0	0	0	0
7200bit/s	0	1	0	0	0	0	0	0
9600bit/s	1	1	0	0	0	0	0	0
19200bit/s	0	0	1	0	0	0	0	0
38400bit/s	1	0	1	0	0	0	0	0
57600bit/s	0	1	1	0	0	0	0	0
115200bit/s	1	1	1	0	0	0	0	0
230400bit/s	0	0	0	1	0	0	0	0

所有其他的 B 位的值都是保留的。默认值是 1100 0000（ 9600 ）。

D1-D2 指示了数据位的数量。

D1	D2
00	5 bits
01	6 bits
10	7 bits
11	8 bits - default

S 位指明了停止位的数量。 S=0：1 停止位， S=1：1、5 停止位。默认值 S=0。

P 位指明了奇偶校验。 P=0：无校验， P=1，有校验。默认值 P=0。

PT1-PT2 指明了了校验的类型。

PT1	PT2
00	odd parity
01	even parity
10	mark parity
11	space parity

FLC1-FLC6 ，默认值为 0，没有流控。

Bit1	XON/XOFF on input
Bit2	XON/XOFF on output
Bit3	RTR on input
Bit4	RTR on output
Bit5	RTC on input
Bit6	RTC on output

Note. The RTR is mapped to either CTS (circuit 106) or RFR (circuit 133). The RTC is mapped to either DTR (circuit 108/2) or DSR (circuit 107). (Circuit 133, RFR(Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name)

XON1-XON8 ，XON 字符（默认值 DC1 ）。

XOF1-XOF8 ，XOFF 字符（默认值 DC3 ）。

PM1-PM8 ，参数掩码。

参数掩码用来指明哪些参数被 RPN 命令所协商。对一个命令来说，参数掩码将被理解为：0=没改变；1=有改变。对应一个回应来说，0=不可接收的；1=可以接收并且使用的。值字节 7 和 8 的比特位掩码如下：

Bit1	bit rate
Bit2	data bits
Bit3	stop bits
Bit4	Parity
Bit5	parity type
Bit6	XON character
Bit7	XOF character
Bit8	reserved

PM9-PM16 ，也是参数掩码。

Bit1	XON/XOFF on input
Bit2	XON/XOFF on output
Bit3	RTR on input
Bit4	RTR on output
Bit5	RTC on input
Bit6	RTC on output

所以保留的值，都被发送方设置为 0，被接收方所忽略。

远程线路状态 RLS （ Remote Line Status ）

3.4.6.3.10. 远程线路状态 RLS （ Remote Line Status ）

这个命令是可选的。

这个命令指明了远程端口线路的状态。
连接状态下，当远程端口线路状态改变时，设备必须发送 RLS 命令。
RLS 命令使用如下的 type 域格式：

表 13：type 域字节

Bit No.	1	2	3	4	5	6	7	8
signal	EA	C/R	0	0	1	0	1	0

Length 字节包含值 2，它有两个值字节。

表 14：DLCI 字节

Bit No.	1	2	3	4	5	6	7	8
signal	EA	1	D L C I					

DLCI 字节的第 2 位没用，固定为 1。EA 位，依据 3.2.1.2 的描述。

表 15：RLS 字节

Value Octet	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8
1	L1	L2	L3	L4	res	res	res	res

一端向另外一端发送一条 RLS 命令，它的各个域，都设置了想要的值的。当另外一端接收到 RLS 命令后，它必须用包含它接收到的值的 RLS 回应帧回应。

L1-L4 位指明了线路状态。 L1 置 0，表示没出现错误。 L1 置 1，指明了如下的错误：
L2-L4:

- 100 越界错误（ Overrun Error ） - 接收到的字符有不可读的字符
- 010 校验错误（ Parity Error ） - 接收到的字符的校验位不正确
- 001 装配错误（ Framing Error ） - 一个字符不是以停止符结束的
- Res 位，由发送方置 0，被接收方忽略。

3.4.6.3.11. 服务协商命令 SNC（ Service Negotiation Command ）

这个命令用来查询和设置特殊 DLC 的特殊服务的。它以特殊的数字视频类型为例子。

在一些情况下， AT 命令和原始数据在同一个 DLC 上混合， 是不太合适的。 这些情况下，可以建立一些特殊的 DLC，传输一些特殊的数据类型。这种情况的例子，尤其对视频传输更有用，当 AT 命令控制了链接，在 DLC 上传输时，音频数据可以由另一个 DLC 运载。在数据流里，用 AT 命令发送 ESC 序列，这种机制是可供选择的。假如这个命令没被用，DLC 默认设置为正常的 AT 命令模式。假如这个命令使用了，在 DLCI 字节指明的 DLC，修改为用来运载特殊字节类型。这个命令的发起方，也可以查询每个 DLCI 的特殊服务。

SNC 的格式如下：

表 16：SNC 格式

Byte No.	1	2	3	4	5
Meaning	Type field code	Length	DLCI	Service Value octet (optional)	Voice Codec Value octet (optional)

Type 域字节格式：

表 17：type 域字节

Bit No.	1	2	3	4	5	6	7	8
signal	EA	C/R	0	0	1	0	1	1

Length 字节包含值 1 或值 3，它有一个或三个值字节。

表 18：DLCI 字节

Bit No.	1	2	3	4	5	6	7	8
---------	---	---	---	---	---	---	---	---

signal	EA	1	DLCI
--------	----	---	------

DLCI 字节的第 2 位没用，固定为 1。EA 位，依据 3.2.1.2 的描述。

表 19：服务值字节

Value Octet	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8
1	EA	S1	S2	S3	S4	S5	S6	S7

EA 位，依据 3.2.1.2 的描述。

表 20：音频编码值字节

Value Octet	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8
1	EA	V1	V2	V3	V4	V5	V6	V7

EA 位，依据 3.2.1.2 的描述。

表 21：服务值的各个比特位

Value Bit	Service
S1	Data
S2	Voice
S3	Reserved
S4	Reserved
S5	Reserved
S6	Reserved
S7	Reserved

表 22：音频编码值的各个比特位

Value Bit	Service
V1	Voice (coded –GSM 06.21)
V2	Voice (coded - PCM 64 kbit/s U-law)
V3	Voice (coded ADPCM 32kbit/s) ITU-T G.726
V4	Voice (coded halfrate)
V5	Voice (coded - PCM 64kbit/s A-law)
V6	Voice (coded PCM 128kbit/s)
V7	Reserved

发送方传输一个 SNC 命令，它的各个域，都设置成该特殊 DLC 可以使用的所有可能的服务的值。接收方用它选择的值的集合，以一个 SN 回应帧回复它。接收方构建回应帧的时候，必须遵守如下的规则：

DLCI 值不能变。

接收方可以而选择发送方提议的服务的子集，但不能是超集。接收方所选择的，才是有效的。假如接收方不支持发送方提议的任何服务，它把服务字节置回复。

即使选择了数据服务，音频编码值字节也总要出现的（ present）。这种情况下，该字节的 V1-V7 比特位，都置 0。

0 值，意味着标准 AT 命令模式。

假如命令里没有包含值字节，它被解释位一个请求，并且接收方用所有可能的服务回应。

假如发送方认为回应是可以接收，也就是说，服务别特务都匹配（ match），发送方根据服务，开始使用 DLC。假如回应是不可接受的，发送方将发起另外一个修改了参数的 PNC 命令，直到最终协商好，或者向更高层传递失败的信息。

假如 DLC 上没有执行过服务协商，它在标准 AT 模式下运行。这种情况下，一个来电的指示，在另一个（ that） DLC 上。

发送方把保留位置 0，接收方忽略这些保留位。

3.4.7. 电源控制与唤醒机制

对各种类型的 MS 和一些 TE 来说，设备耗电非常少，这是非常重要的。比如，在非激活的条件下，进入各种省电状态，就能达到这个目标。假如 TE 和 MS 都希望进入省电状态，MUX 系统必须能够完全关闭。通过使用 MUX 的控制通道，来达到这个功能。

假如 TE 和 MS 都希望进入省电状态，一个 PSC 命令在 MUX 的控制通道上，发送到另一端。接收端接收到这个消息，将完全传输当前进行中的所有的帧，向上层报告一个忙或者电源关闭的情况，冻结所有的定时器，发送回一个 PSC 回应帧。当 PSC 命令的发起方，收到确认后，它就可以进入省电状态了。

每一端都可以发送 PSC 命令，请求另外一端进入省电状态。接收方必须用 PSC 回应帧，确认 PSC 命令帧，但不一定要遵从命令而进入省电状态。如果没有收到回应帧，发送方重发命令帧，但是，必须先执行唤醒流程。

每一端都可以从省电状态，通过发送由连续的标识字符组成的唤醒信号，发起唤醒流程。当另一端收到标识字符时，如果必要它将醒来 (wake-up)，并且发送标识字符。当发起端接收到这些字符后，它停止发送标识字符，并且开始传输第一个数据帧。当接收端检测到有效帧后，它将停止发送标识字符。两端都解冻它们的定时器，像以前一样继续运行。

假如 T3 时间内，唤醒流程收不到回应帧 (连续的标识字符)，将向上层发出一个警告，并且停止标识字符的传输。

3.4.8. 流控

3.4.8.1. RTR 流控

图 12 显示了 DTE 到 DCE 的链接。本段定义的流控方案，同样适用于 DTE-DTE 的链接。07.10 的两端都用 RTR (RFR/CTS) 流控配置过。对本地应用程序来说，流控信号就是其他三个流控信号和对方设备的 RTR 信号的联合。标签为 FCS1-FCS3 的流控信号定义如下：

FCS1 MSC 命令或者 type2 的集成层的控制信号字节的第 2 位。每个 DLCI 都流控。

FCS2 07.10 通过控制通道命令 Fcon 和 Fcoff (basic 模式) 聚合的流控。

FCS3 07.10 内部的缓冲区管理 (implementation specific)

流控信号 FCS1-FCS3 跟对端的 RTR 信号组合在一起，产生了本地的 RTR 输入信号。比如，为了模拟 DTE 串口的 CTS 信号的表达式是：

$$\text{DTE.CTS} = \text{DCE.RTR AND FCS1 AND FCS2 AND FCS3}$$

在 moden 状态命令 (当使用集成层 1, 3 和 4) 的第 2 位 (FC) 和第 4 位 (RTR)，或控制信号字节 (当使用集成层 2) 的第 2 位 (FC) 和第 4 位 (RTR)，流控模拟器复制了发送的 RTR 信号。

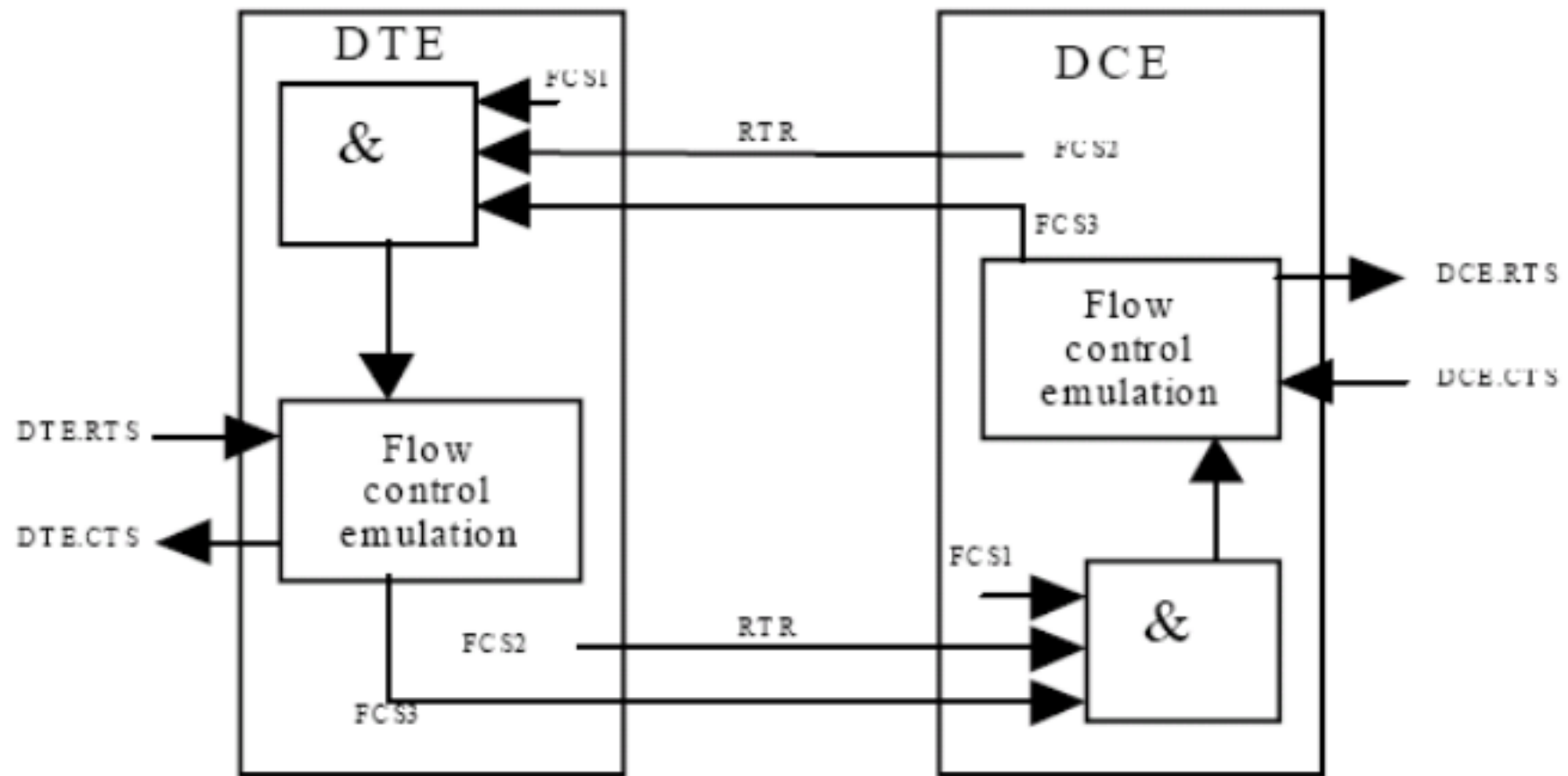


Figure 12: RTR Flow Control

3.4.8.2. XON/XOFF 流控

当嵌入了 XON/XOFF 字符，一些 07.10 的实例可以检测来自本地应用程序的 XON/XOFF 字符。这种情况下，字符遵照 07.10，但不发送给对端的 07.10 实例，比如 XON/XOFF 字符被过滤出来，流控信号作为 07.10 流信号传送，见图 13。

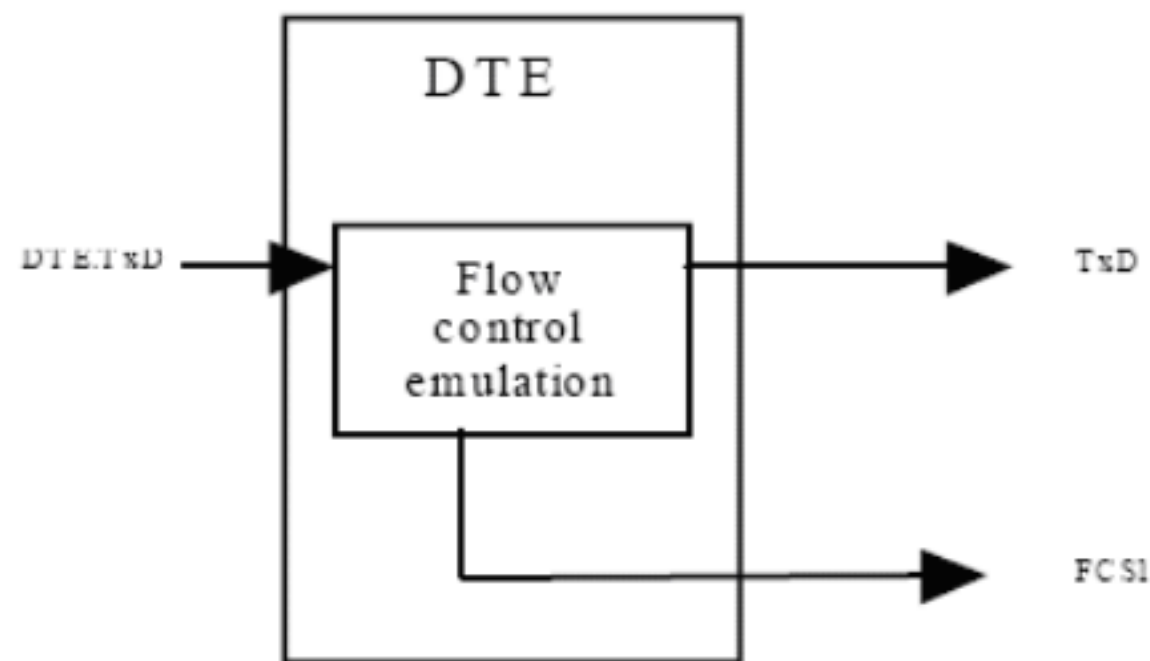


Figure 13: XON/XOFF Flow Control on input

如果输出 (output) 使用了 XON/XOFF 流控，07.10 将用 XON/XOFF 字符实现本地应用程序的流控，比如，XON/XOFF 字符根据 07.10 流控信号 (参见图 14)，被插进数据流。

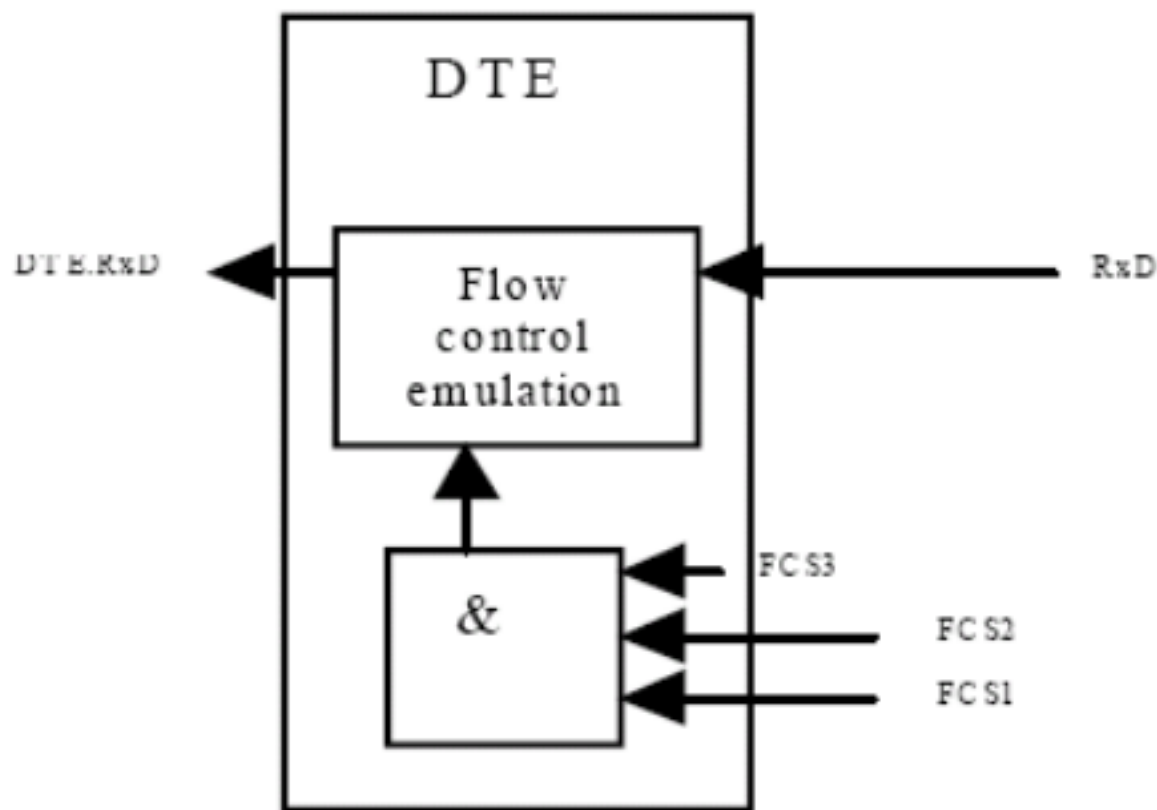


Figure 14: XON/XOFF Flow Control on input

3.5. 集成层 Convergence Layer

集成层的定义，允许带结构的数据，通过 MUX 传输，不丢失结构和其他与数据流相关的参数。集成层通常用来通过一个 DLC，运载 V.24 的状态，或者用来确保一个声音编码帧的边界受到保护。

集成层决定数据是否被 ERM 模式或者 non-ERM 模式的流程所运载。
特殊的集成层用于连接被使用的 DLCI，但是可能通过 MUX 控制信道的使用，跟这些默认的 DLCI 进行协商。

3.5.1. 类型 1-未结构化的字节流

未结构化的字节流数据，比如 64kbit/s 的未编码的声音或者不带 V.24 控制信号的常规异步数据，被直接插入到 I 域。这种情况下，可以说，集成层是空的。
Type1 是每个 DLC 的默认集成层。

3.5.2. 类型 2-带参数的未结构化的字节流

传输 V.24 控制信号相关的一个数据流，它每个 I 域的第一个字节包括信号的状态，见下图。扩展位的使用，允许添加其他字节来满足其他的情况。目前，可选的第二个字节，被定义为处理中断信号的传输。

Bit No	1	2	3	4	5	6	7	8
Signal	EA	FC	RTC	RTR	Reserved 0	Reserved 0	IC	DV

图 15：控制信号的字节格式

控制信号字节的描述如下：
Bit1。EA 为 1 表示本字节是该域的最后一个字节，其他字节 EA 位为 0。
Bit2。DLC 的流控（Flow Control）。Non-ERM 模式下，设备不能接收帧时，置 1。ERM

模式下，发送方，永远置 0，接收方忽略它。

Bit3。准备好发送 (Ready To Communicate)。设备准备好发送的时候，置 1。

Bit4。准备好接收 (Ready To Receive)。设备准备好接收数据的时候，置 1。

Bit5/6。保留。发送方要置 0，接收方忽略它。

Bit7。来电 (Incoming Call) 标识。该位置 1 指明有个来电。

Bit8。数据有效 (Data Valid)。该位置 1 指明有效的数据正被传送。

接收方和发送方的 V.24 控制信号字节的各 BIT 位，分别在表 23 和表 24 中。

Table 23: Mapping from the control signal octet by a receiving entity

Control Signal Byte	DTE receiving		DCE receiving	
bit number, name	signal	V.24 circuit	signal	V.24 circuit
2, FC	DLC flow control	-	DLC flow control	-
3, RTC	DSR	107	DTR	108/2
4, RTR	CTS	106	RFR (note)	133
7, IC	RI	125	ignored	-
8, DV	DCD	109	ignored	-
NOTE Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.				

Table 24: Mapping to the control signal octet by a sending entity

Control Signal Byte	DTE sending		DCE sending	
bit number, name	signal	V.24 circuit	signal	V.24 circuit
2,FC	frame flow control	-	frame flow control	-
3,RTC	DTR	108/2	DSR	107
4,RTR	RFR (note)	133	CTS	106
7,IC	always 0	-	RI	125
8,DV	always 1	-	DCD	109
NOTE. Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.				

Non-ERM 模式下，FC 位为 DLC 提供帧的流控。如果一端不能接收数据帧了，比如，由于缓冲区管理的问题，它将置 FC 位为 1。当它又能接收数据帧了，它将置 FC 位为 0。如果一端接收一个 FC 位置 1 的数据帧，它将停止传输数据帧，直到收到一个 FC 位置 0 的数据帧。

如果一端由于流控的原因，不能传输数据帧了，但它想自己停止接收数据帧，或者想在控制信号中发送一个改变信号或者想发送一个中止状态，它要继续发送不包含用户数据（比如只包括控制信号的字节和可选的中止信号的字节）的帧。

ERM 模式下，FC 位没有用，将被发送方置 0，被接收方忽略。如果一端已经不再（被防止）发送 I 帧，例如，正接收一个 RNR 帧，如果它想在控制信号中发送一个改变信号，或者想发送一个中止状态，它要继续发生一个不包含用户数据的 UI 或者 UIH 帧。

Bit No	1	2	3	4	5	6	7	8
Signal	EA	B1	B2	B3	L1	L2	L3	L4

图 16：中止信号的字节格式

中止信号字节携带能让 DLC 检测出来的中止信息。其意义如表 25。

表 25：中止信号的字节含义

Bit	Value	Meaning
B1	1	Octet encodes a break signal
	0	Octet does not encode a break signal
B2	1	Discard data in buffers
	0	Do not discard data in buffers
B3	1	Transmit break signal onwards as soon as possible
	0	Transmit break signal onwards in sequence
L1-L4	4-bit value	Length of break in units of 200ms

L1 是 break length 的最低位， L4 是最高位。

当一端接收到中止字节（ break octet ），它将处理这个信息，并用适当的方法传递它。这不在本文档论述范围之内。

I 域的其他字节包括该 DLC 所需的数据。

3.5.3. 类型 3-不可中断的帧数据

一个不可中断的帧数据的例子就是由连续的音频数据帧组成的音频编码数据。音频编码数据数据帧，带有完整的帧结构，在尽可能短的时延下，到达音频解码器，是非常重要的。最简单的把办法是确保把一个完整的音频数据帧放进一个 I 域中。在传输过程中，即使高优先级的数据在等待，这个数据帧也不能变短（shortened 压缩？）。

在传输器（transmitter）那，数据的每个帧，都被插进一个 I 帧，UI 帧或 UIH 帧的 I 域中。接收器作为完整帧，处理数据，传输它。

音频编码数据应该使用 UI 帧或 UIH 帧传输，因为重新传输的延时，通常是不能接受的。

3.5.4. 类型 4-可中断的帧数据

当时延不像 type3 那么重要时，这种类型的集成层，被用来传输有结构的数据。结构化的数据可以被分割成多帧，并且可以在另外一端重新装配。按 PPP 封装的 IP 数据，就是一个例子，它的数据可以运载在 Type4 的集成层上。

每个 type4 的数据帧的第一个字节是一个控制字节，如下图 17。

Bit	No	1	2	3	4	5	6	7
Signal	EA	-	-	-	-	-	B	F

Figure 17: Format of Type 4 octet

EA 位以后扩展，如果需要用到多个字节时。这里设置为 1。

B 和 F 位用来指示该数据帧是一个序列的第一个数据帧，中间的数据帧或者最后一个数据帧。它的含义如下面表 26 所示。

表 26：B 位和 F 位的含义

B 位	F 位	含义
1	0	一个序列的第一帧
0	0	一个序列的中间帧
0	1	一个序列的最后一帧
1	1	数据完全被包含在一个数据帧中

注意 1：假如其他的帧结构，比如第二层协议，已经包含在数据流中，按 PPP 封装的 IP 数据可以使用 type1 的集成层运载。

注意 2：如果一个数据帧作为最后一个数据帧进行编码，或者所有的数据包含在一个数据帧中，数据帧不能被压缩，因为，压缩会导致帧头 (header) 含义的错误。构造不包含任何数据的数据帧作为最后一个数据帧，或者当数据帧需要压缩时避免使用 11 编码(11 code)，是明智的。

3.6. DLCI 值

默认情况下，所有 DLC 使用了 type1 的集成层。其他层的使用，可以通过 MUX 的控制通道进行协商。

表 27：DLCI 分配表

用法 Usage	DLCI 编号 (10 进制)	Priority 优先级
Multiplexer control channel	0	0
AT commands (07.07 and 07.05)	1-7	7
AT commands (07.07 and 07.05)	8-15	15
AT commands (07.07 and 07.05)	16-23	23
AT commands (07.07 and 07.05)	24-31	31
AT commands (07.07 and 07.05)	32-39	39
AT commands (07.07 and 07.05)	40-47	47
AT commands (07.07 and 07.05)	48-55	55
AT commands (07.07 and 07.05)	56-61	61
Reserved	62-63	

DLCI 0 为控制通道保留。
DLCI 62 为 basic 模式下 BOFC 和 EOFC 的 ETSI 目的保留，也只能这么用。
DLCI 63 为 HDLC 模式下的 ETSI 目的保留（由于特别的含义），也只能这么用。
在 DLC 指定优先级的信息的缺省情况下，使用这个表。

3.7. 系统参数

MUX 定义了下面这些系统的参数。T1，N1，N2 和 k 可以通过控制通道进行协商，或者用这里给的默认值。T2 和 T3 通过 AT+CMUX 命令进行设置。

3.7.1. 确认时间 T1

T1 就是通道一端在采取其他动作（比如重发）之前，等待确认信息的时间。通道两端可以有不同的 T1 值。

单位是毫秒。 Times of up to 2.55 seconds may be used.
默认值是 100ms，最小值是 10ms。

3.7.2. 帧的最大长度 N1

N1 定义了帧的信息域，能容纳的字节的最大数量。它不包括为了透明而增加的字节。

默认值是 64 字节 (advanced) 或者 31 字节 (basic)，范围是 1-32768。

注意：如果使用了 a Type 2 convergence layer，N1 要小心选择了。帧必须足够大，以包含完整的协议帧。

3.7.3. 最大重发次数 N2

N2 定义了重发的次数。通道两端可以有各自不同的 N2 值。

默认值是 3，范围是 0-255。

3.7.4. 窗口大小 k

窗口大小参数 k 定义了一个 DLC 能理解的最大的 I 帧的数目。不同的方向，需要用同一个值。窗口大小不能大于 7。

这个参数只使用于错误恢复模式激活的情况下。

默认值是 2，范围是 1-7。

3.7.5. 控制通道的响应时间 T2

T2 就是 MUX 的控制通道重发一个命令前的等待时间。

T2 必须大于 T1。单位也是毫秒。 Times of up to 2.55 seconds may be used.

默认值是 300ms，最小值是 20ms。

3.7.6. 唤醒流程的响应时间 T3

T3 就是发送端在无响应报警之前，对省电唤醒命令所等待的时间。

单位是秒。 Times of up to 2.55 seconds may be used.

默认值是 10s，最小值是 1s。

3.8. 启动和关闭 MUX

启动 MUX 使用 +CMUX 命令。这个命令指明了启动 MUX 使用的 ERM 还是 non-ERM。发送 SABM，初始化 DLCI 0。然后，使用同样的 3.4.1 的流程，建立其他的 DLC。MUX 可以跟已经建立的 DLC，协商参数。

关闭 MUX，可能有 TE 或 MS 端发起。先要用 3.4.2 的流程，释放除了 DLC 0 的所有 DLC。然后，MUX 通过控制通道，发送一个关闭的消息。这个消息被双方确认后，就退出了多串口模式。如果在 T2 时间内收不到对方对关闭命令的回应，发起方必须重发关闭命令，如果在 T3 的时间内还收不到，必须强行关闭。

4. Error Recovery Mode

错误覆盖模式，用的不多，但是比没有覆盖模式安全，并且只能在高级模式中使用。帧模式是 1 帧。在这里不做介绍。