# CANTP组帧拆帧

流控帧是会在多帧传输中使用到。

**FC(流控帧)**

接收端在收到首帧后，会发出一个流控帧，第一个字节PCI的高四位为3，低四位为FS（FlowStatus），FS可以为0代表ContinueToSend（CTS），1代表Wait（WT），2代表Overflow（OVFLW），3-F为预留，通常我们遇到的值为0。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 流控帧FlowControl（FC） | Byte#0 | | Byte#1 | Byte#2 | Byte#3 | Byte#4 | Byte#5 | Byte#6 | Byte#7 |
| Bits7-4 | Bits3-0 |
| 3 | FS | BS | Stmin | xx | xx | xx | xx | xx |

流控帧数据格式

流控帧的作用在于接收端告知发送端接收能力，包含BlockSize（BS）和SeparationTimeMin（STmin）两个参数。

BS表示接收端允许发送端连续发送的最大连续N\_PDU帧数

BS为0的时候高速发送端可以发完剩余的全部N-PDU

BS为01-0xFF的时候，表示发送端可以发的连续帧最大帧数

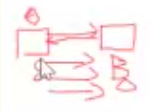


比如A、B通信，B会把流控帧发给A。

|  |  |
| --- | --- |
| Hex value | Description |
| 00 | BlockSize(BS) The BS parameter value zero(0) shall be used to indicate to the sender that no more FC frames shall be sent during the transmission of the segmented message.The sending network layer entity shall send all remaining consecutive frames without any stop for further FC frames from the receiving network layer entity. |
| 01-FF | BlockSize(BS) The range of BS parameter values shall be used to indicate to the sender the maximum number of consecutive frames that can be received without an intermediate FC frame from the receiving network entity. |

Definition of BS values

STmin表示发送端发送连续帧的最小间隔时间。

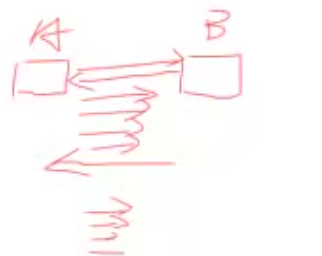


STmin也就是指A发送多帧给B的时候的两帧之间的最小的间隔时间。

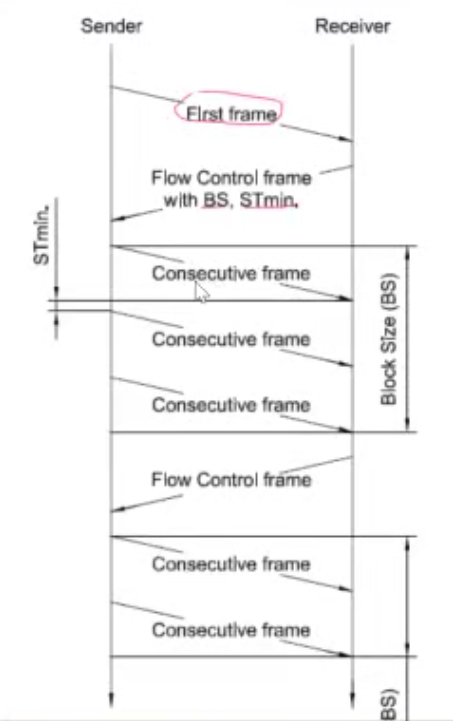
|  |  |
| --- | --- |
| **Hex value** | **Description** |
| 00-7F | **SeparationTime(STmin)range:0ms-127ms**  The units of STmin in the range 00 hex - 7F hex are absolute milliseconds(ms) |
| 80-F0 | **Reserved**  This range of values is reserved by this part of ISO 15765. |
| F1-F9 | **SeparationTime(STmin)range:100us-900us**  The units of STmin in the range F1 hex - F9 hex are even 100 microseconds(us),where parameter value F1 hex represents 100us and parameter value F9 hex represents 900us. |
| FA-FF | **Reserved**  This range of values is reserved by this part of ISO 15765. |

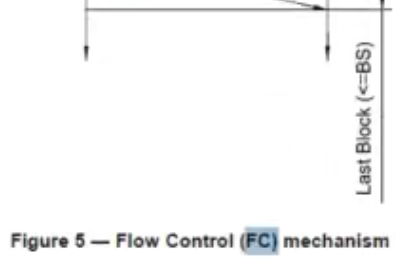
Definition of STmin values

发送端收到接收端反馈的流控帧后，发送端会按照流控帧给出的接收能力进行发送连续帧，即连续传输的连续帧不超过BS，连续帧的发送间隔时间大于等于STmin。若接收端已经收满BS数量的数据，且与首帧中包含的数据长度对比发现数据还没接收满，则会继续发送流控帧，发送端会接着发送连续帧，以此类推，直至所有数据发送完成。



比如A打算发10帧报文给B，但是先发了4帧后，b的数据以及接收满了，b则会继续发一个流控帧给A，要求A将剩下的6帧报文发送完毕。流控帧的本质作用用于接收端告知发送端自己的接受能力。





**举例**

例1：用诊断仪读取某节点ECU的DID 0xF280，诊断仪需要发送0x22 0xF2 0x80 3个字节数据，多余数据用0xAA填充，写出诊断仪的请求数据

**分析：**该例子只需要发送3个字节数据，小于7个字节，所以单帧发送即可，单帧byte0高4位为0，第4位len = 3，后面跟着发送的数据0x22 0xF2 0x80，其他数据使用0xAA填充

(PCI) (SID)

**答：**诊断仪：0x30 0x22 0xF2 0x80 0xAA 0xAA 0xAA 0xAA

**例2：**用诊断仪写入某节点的ECU的DID 0xF190，诊断仪需要发送0x2E 0xF1 0x90 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10共20个字节长度的数据，多余字节用0xAA填充，请写出诊断仪请求过程中与ECU之间的交互数据。

分析：该例子需要发送20个字节长度数据，大于7个字节，所以需要利用多帧格式，首帧byte0的高4位为1，低4位与byte1为数据长度20，即0x014，后面再跟6个字节0x2E 0xF1 0x90 0x00 0x01 0x02；诊断仪发完首帧后，ECU端需要返回流控帧，byte0的高4位为3；第一个连续帧为byte0的高4位为2，低4位为1，后面跟7个字节0x03 0x04 0x05 0x06 0x07 0x08 0x09；第二个连续帧为byte0的高4位为2，低4位为2，后面跟7个字节0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10；

**答：**

诊断仪：0x10 0x14 **0x2E 0xF1 0x90 0x00 0x01 0x02**

ECU：0x30 BS STmin 0xAA 0xAA 0xAA 0xAA 0xAA

诊断仪：0x21 **0x03 0x04 0x05 0x06 0x07 0x08 0x09**

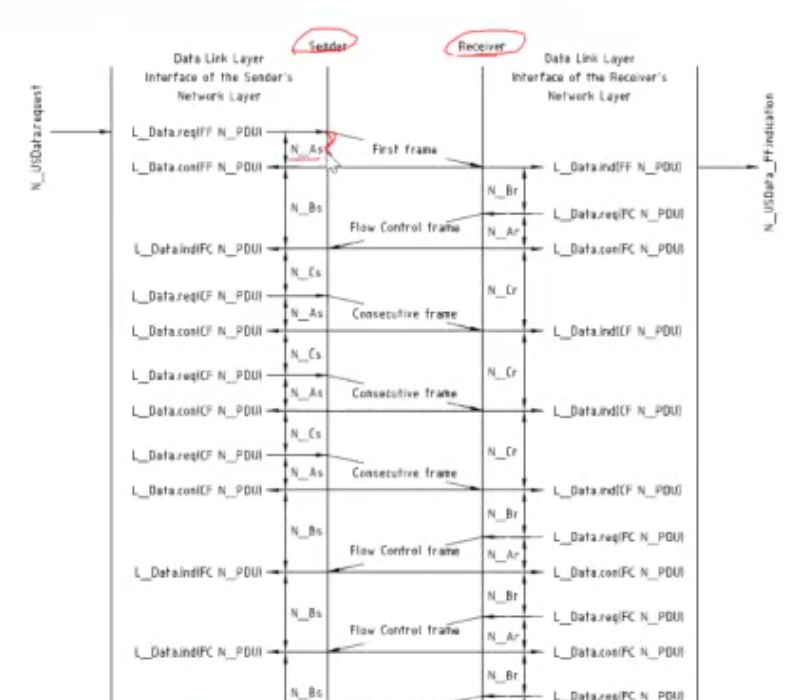
诊断仪：0x22 **0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10**

# CANTP时间参数

**网络层时间参数要求**

CAN诊断的数据传输不仅需要遵从上述格式要求，还有严格的时间要求。网络层涉及的时间参数有N\_As,N\_Ar,N\_Bs,N\_Br,N\_Cs和N\_Cr。这些时间参数在传输过程中的对应关系如图所示：

多帧传输过程中，诊断发送报文的时候，首帧发送到接受端接受到这条报文的时间叫做N\_As。N\_As有一个超时的要求，就是发送端发送报文之后，接收端不可能一直接受不到。如果是这样，那么会超时。若ACK拉高了，说明总线上报文接收到发送端的报文了。CANtp层，发送端调用CANif开始计时到驱动检测到了总线的ACK，会给CANtp回复一个tx Confirmation。从发送到收到Confirmation这段时间称为N\_As，发送端的时间参数，此参数可以通过达芬奇工具配置CANtp需要的参数从而进行设置。



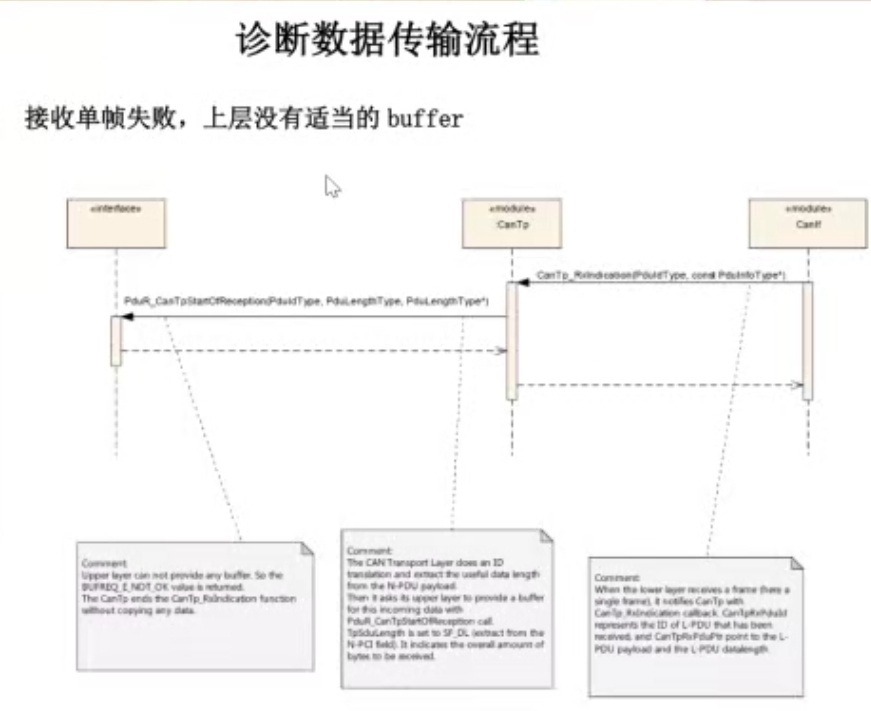
N\_BS是发送端发送首帧出去后，收到接收端的流控帧为止。也可以进行配置，超时则会进行一些异常处理，终止当前的诊断请求。

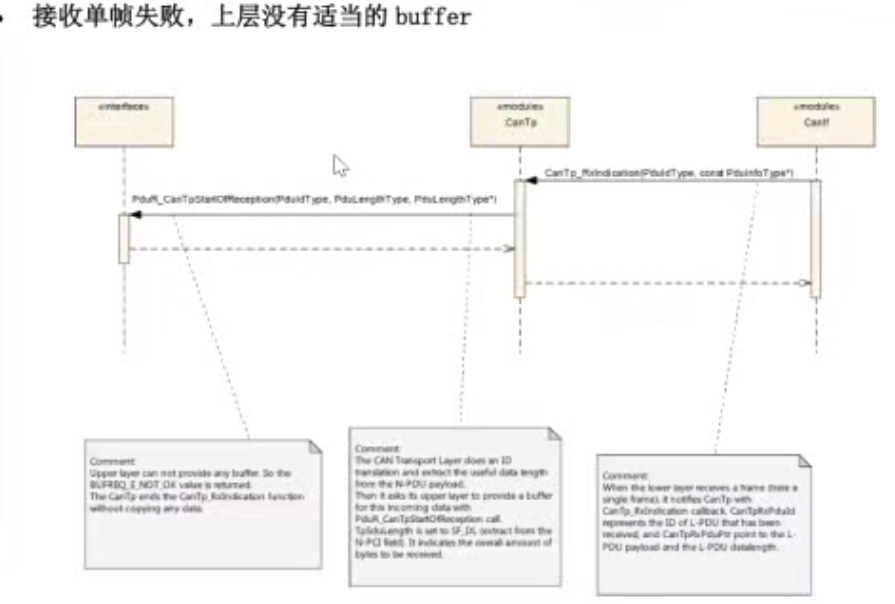
N\_Cs是收到的流控帧之后间隔多久发连续帧，续帧发完之后，间隔多久再发续帧，N\_CS即是STmin，续帧和续帧之间的时间间隔。STmin是续帧确认发出之后，确认收到了，到下一个续帧发送的时间。接收端也可以设置超时，收到首帧之后，过多久才发流控帧，如果一直没有发流控帧，这是有一个超时的。比如CANTp的程序跑飞了，或者是有一些程序异常，导致迟迟未发流控帧，但是计数器还在一直计数，计数到超时之后，可能程序就报错了，程序就终止了诊断。

N\_Ar就是CANtp调的CANif的发送接口之后，开始计时到can if回一个t3 Confirmation之后，通过ack确认接收端已经收到流控帧之后，这段时间是有超时的。设置超时的目的是为了避免发送流控帧之后接收端一直没有回复ack，保证安全性。N\_cr是发送流控帧之后，收到了接收端发的续帧之间的时间间隔。如果发了流控帧之后一直收不到连续帧，N\_Cr会产生超时。

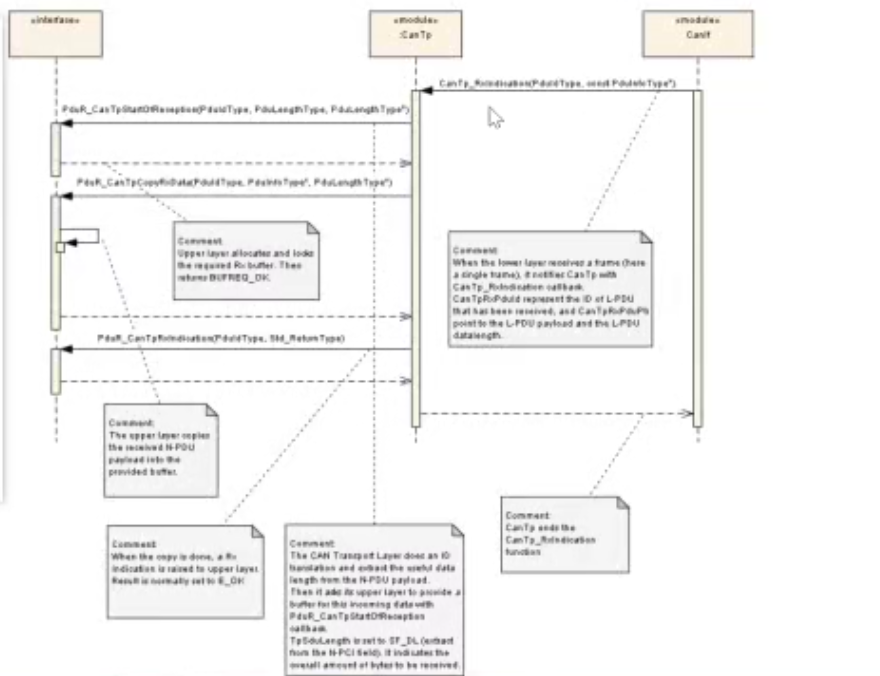
这计时网络层是时间参数，发送端有As、bs、cs接受端有ar、br、cr，它们各自超时监控的含义是不一样的。

|  |  |
| --- | --- |
| 时间参数 | 描述 |
| N\_As | 发送端发送一条CAN帧的时间 |
| N\_Ar | 接收端接收一条CAN帧的时间 |
| N\_Bs | 发送端发送完首帧或者发送完续帧或者接收到流控帧到接收到下一帧流控帧的时间 |
| N\_Br | 接收端收到首帧或者发送完流控帧或者接收到续帧到请求发送下一帧流控帧的时间 |
| N\_Cs | 发送端收到流控帧或者发送完续帧到请求发送下一帧连续帧的时间 |
| N\_Cr | 接收端发送完流控帧或者接收到续帧到接收到下一帧连续帧的时间 |



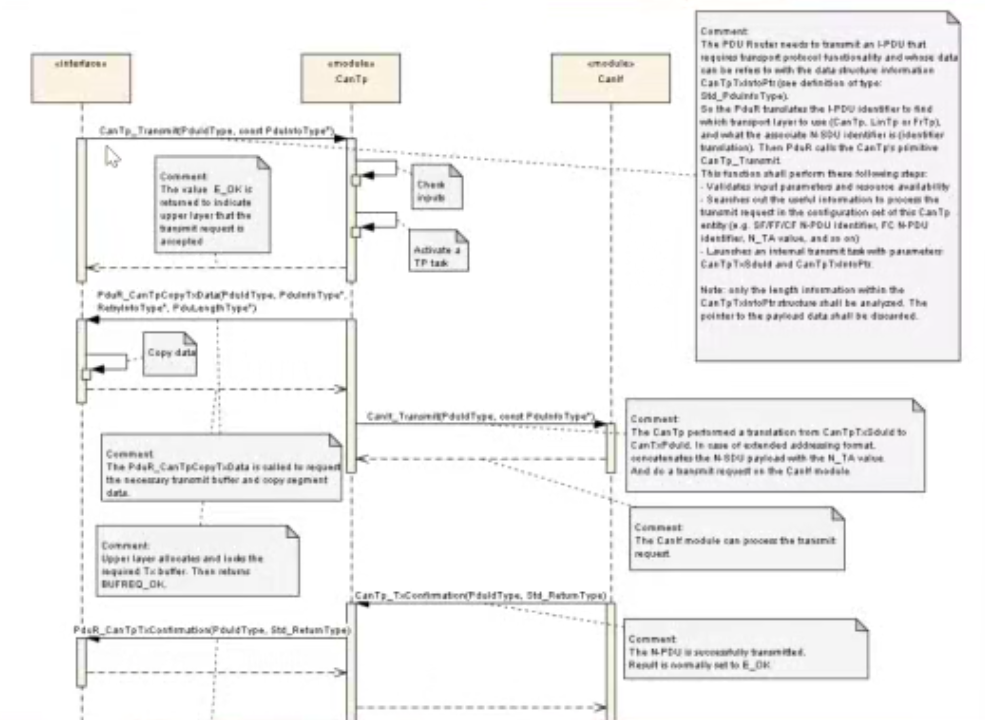


成功接收单帧



上图是代码中的函数调用关系。

成发送单帧



成功发送多帧

