# CanTp组帧和拆帧过程

**CAN诊断网络分层**

CAN诊断的网络分层也是参考OSI模型，该模型定义了网络互联的7层架构（物理层、数据链路层、网络层、传输层、会话层、表示层和应用层）。

CAN诊断通信包含了诊断应用（ISO 15765/ISO14229）、网络层（ISO15765-2）、数据链路层（ISO 11898-1）和物理层。

|  |  |  |
| --- | --- | --- |
| Table1—Enhanced and legislated OBD diagnostic specification applicable to the OSI Layers | | |
| Open System Interconnection(OSI) layers | Vehicle manufacturer enhanced diagnostics | Legislated on-board diagnostics(OBD) |
| Diagnostic application | User defined | ISO 15031-5 |
| Application layer | ISO 15765-3 | ISO 15031-5 |
| Presentation Layer | N/A | N/A |
| Session Layer | ISO 15765-3 | N/A |
| Transport Layer | N/A | N/A |
| Network Layer | ISO 15765-2 | ISO 15765-4 |
| Datalink Layer | ISO 11898-1 | ISO 15765-4 |
| Physical Layer | User defined | ISO 15765-4 |

当前随着统一诊断（UDS）服务发展，诊断应用层已经基本使用ISO14229标准。我们今天CANtp模块就属于网络传输层，就是使用ISO 15765-2的标准。

DCM模块指诊断控制管理模块，在应用层、表示层、会话层实现。传输层和网络层通过CanTp模块实现。

CAN诊断由发送端的请求与接收端的响应构成，诊断即为发送端与接收端数据往来。有的诊断一条消息完成，有的诊断需要多条消息完成，毕竟在诊断中，一条CAN消息只包含8个字节长度。对于一条CAN诊断消息的分段发送问题，即为网络层需要讨论的内容。

CanTP层传输的报文分为SF（单帧），FF（首帧），CF（连续帧），FC（流控帧）4类。

注：该文章只考虑正常地址模式

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 协议控制信息 | 服务ID | 消息字段 | | | | | | |
| 诊断仪请求帧 | PCI | SID | D1 | D2 | D3 | D4 | D5 | D6 | 单帧 |
| PCI | LEN | SID | D1 | D2 | D3 | D4 | D5 | 首帧 |
| PCI | D1 | D2 | D3 | D4 | D5 | D6 | D7 | 续帧 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 应答帧 | PCI | RSID | D1 | D2 | D3 | D4 | D5 | D6 | 单帧 |
| PCI | LEN | RSID | D1 | D2 | D3 | D4 | D5 | 首帧 |
| PCI | D1 | D2 | D3 | D4 | D5 | D6 | D7 | 续帧 |

PCI(Protocol Control Information)协议控制信息，包含了PDU单元类型和消息字节长度。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PCI数据结构 | | | | | | | | |
| 类型 | PCI类型 | | | | 附加信息 | | | |
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| SF(单帧) | 0 | 0 | 0 | 0 | Length | | | |
| FF(首帧) | 0 | 0 | 0 | 1 | Length高4位 | | | |
| CF(续帧) | 0 | 0 | 1 | 0 | 帧计数器 | | | |

单帧：诊断报文只需要1条就可以实现了。

当字节大于8字节时，就需要发送多帧报文。第一帧叫首帧，后续帧叫做连续帧。

CAN报文有8个字节，但是诊断报文不能把这8个byte数据当成payload来处理，

因为诊断的多帧或者单帧传输的时候，对端是知道传输的帧类型的。

PCI(Protocol Control Information)协议控制信息，包含了PDU单元类型和消息字节长度。

单帧传输的第一个byte叫做PCI。所以通过PCI可以知道帧类型。

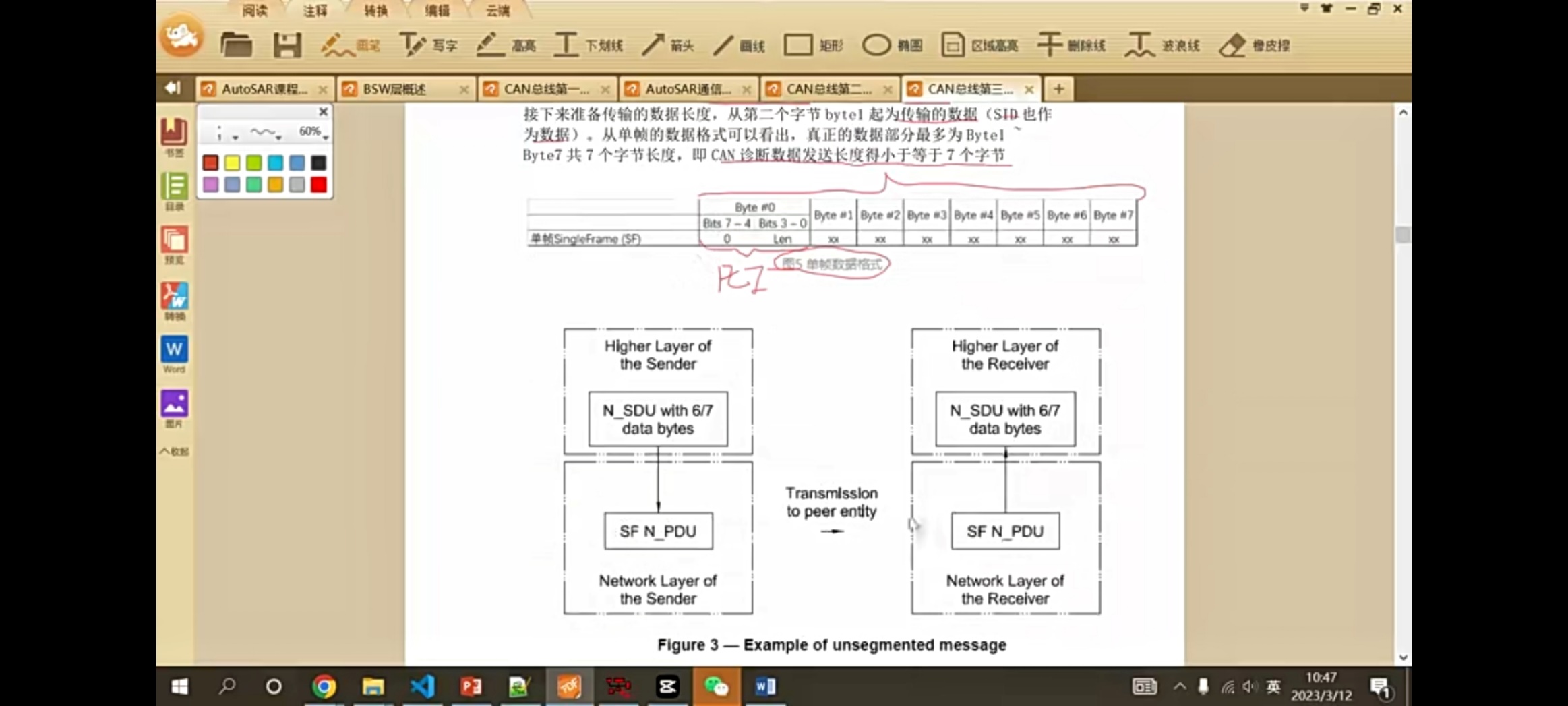
只传输一帧CAN报文(一帧N-PDU)，称为SF，传输7byte（在正常地址情况下）数据字节的消息，不需要NAD，byte0作为PCI使用。

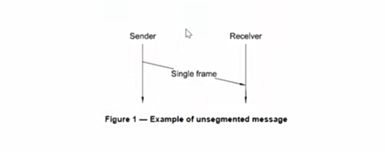
N-SDU指诊断所要发送的数据长度

SF(单帧)

在发送端到接收端的请求仅为一条消息时，即为单帧（SingleFrame），单帧数据格式如下表，单帧第一个字节byte0为PCI，PCI的高4位为0，PCI低四位为接下来准备传输的数据长度，从第二个字节byte1起为传输的数据（SID也作为数据）。从单帧的数据格式可以看出，真正的数据部分最多为Byte1~Byte7共7个字节长度，即CAN诊断数据发送长度得小于等于7个字节。

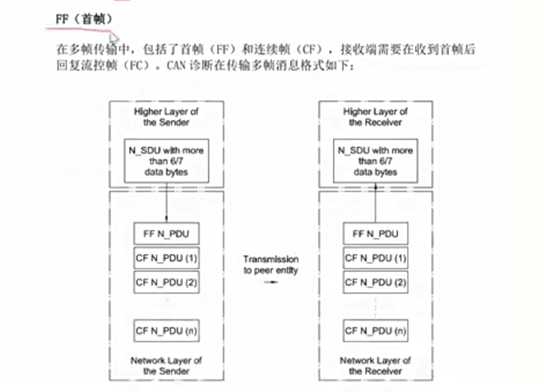
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Byte #0 | | Byte #1 | Byte #2 | Byte #3 | Byte #4 | Byte #5 | Byte #6 | Byte #7 |
| Bits7-4 | Bits3-0 |
| 单帧SingleFrame(SF) | 0 | len | xx | xx | xx | xx | xx | xx | xx |

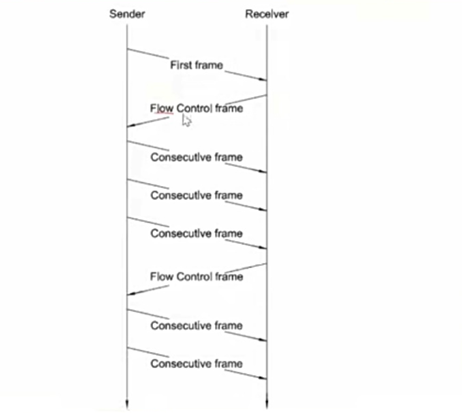




**多帧传输**会引入流控帧的概念。即需要传输的数据比较多，就需要多帧传输，发送过程中需要把N-SDU分割成多帧N-PDU来发送。接收过程中把多帧N-PDU重组发给上层。多帧传输分为首帧和续帧。

**FF（首帧）**，在多帧传输中，包括了首帧（FF）和连续帧（CF），接收端需要在收到首帧后回复流控帧（FC）。CAN诊断在传输多帧消息格式如下：





首帧格式如下表所示，首帧第一个字节PCI的高四位为1，PCI低四位与第二个字节LEN合起来为接下来传输的数据长度，从第三个字节byte2起为传输数据。最大多帧长度为4095个字节。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 首帧FirstFrame(FF) | Byte#0 | | Byte#1 | Byte#2 | Byte#3 | Byte#4 | Byte#5 | Byte#6 | Byte#7 |
| Bits7-4 | Bits3-0 |
| 1 | Len | | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 |

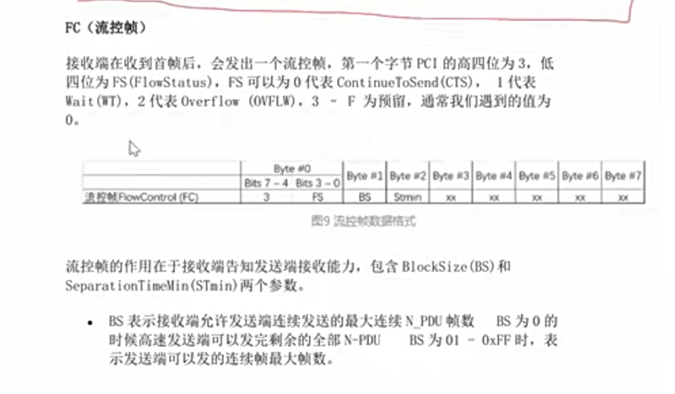
**CF(连续帧)，**连续帧格式如下表所示，连续帧第一个字节PCI的高四位为2，PCI的低四位为帧计数器，第二个字节byte1起为剩余传出的数据。例如首帧已经传输了Data0，Data1…至Data5，则第一个连续帧接着传输Data6，Data7…至Data12，第二个连续帧接着传输Data13，Data14…，以此类推直至所有数据传输结束。

续帧SN从0开始，到15后又返回0开始。（注意：FF发完后的第一个CF的SN为1）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 连续帧ConsecutiveFrame(CF) | Byte#0 | | Byte#1 | Byte#2 | Byte#3 | Byte#4 | Byte#5 | Byte#6 | Byte#7 |
| Bits7-4 | Bits3-0 |
| 2 | SN | Data6 | Data7 | xx | xx | xx | xx | xx |

**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4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时，表示发送端可以发的连续帧最大帧数。