

道路车辆——控制局域网诊断——第 2 部分：网络层服务

1、适用范围

ISO15765 中的这部分具体指明一个网络标定协议应该使之与基于 CAN 的车辆网络系统相适应，就像 ISO11898 中指明的一样。它已经被定义，与建立在 ISO14299-1 和 ISO15031-5 基础上的诊断服务相一致，但是它不仅仅局限于使用它们，同时和其他大多数汽车网络通信互相兼容。这个协议具体指明了一个不确定的通信。

2、参考标准

以下的参考文献对与这部分协议是不可缺少的。对于注明日期的参考，仅仅是这个应用的版本；对与没有注明日期的参考，有最近的版本（包括各种修改过的版本）

ISO 11898-1, 道路车辆—控制局域网—第一部分：数据链路层和物理信号
ISO/IEC 7498 (全文), 信息技术—开环系统的内部连接—基本参考模型

3、术语、定义和缩略术语

BS	块大小
CF	连续帧
confirm	确认服务的原语
ECU	电控单元
FC	流量控制
FF	第一帧
FF_DL	第一帧的数据长度
FS	流动状态
Indication	指示服务的原语
Mtype	信息种类
N_AE	网络地址扩展
N_AI	地址信息
N_Ar	网络层时序参数 Ar
N_As	网络层时序参数 As
N_Br	网络层时序参数 Br
N_Bs	网络层时序参数 Bs
N_ChangeParameter	网络层服务名称
N_Cr	网络层时序参数 Cr
N-Cs	网络层时序参数 Cs
N_Data	网络数据
N_PCIttype	网络协议控制信息种类
N_PDU	网络协议数据单元
N_SA	网络源地址
N_SDU	网络服务数据单元
N_TA	网络目标地址

N_TAtype	网络目标地址类型
N_USData	网络层未确认的分段数据传输服务的名称
NWL	网络层
Request	请求服务的原语
R	接收器
S	发送器
SF	单一帧
SF_DL	单一帧的数据长度
SN	序列号
STmin	最小分段时间

4、网络层概述

4.1 概述

本节介绍网络层的整体功能。ISO15765 的这部分具体指定一个不确定的网络层通信协议用来在各网络节点之间交换数据，比如从 ECU 到 ECU，或外部测试装置与 ECU 之间。如果要被传输的数据没有适合一个单一的 CAN 帧，那么会提供一种单独的方法。

4.2 网络层提供的向更高层的服务

由网络层提供的服务接口定义了所需要的访问功能服务集，例如，传送或接受数据和设置协议参数。

定义了一下两种类型的服务：

A) 通信服务

以下定义的服务，可以提供高达 4095 字节的数据传输

N_USdata.request: 这个服务用来请求数据传输。如果需要，网络层可以分割数据。

N_USData_FF,indication: 这个服务用来标识接收上层分段数据的开端。

N_USData.indication: 这个服务用来把接收的数据提供给更高一层。

N_USData.confirm: 这个服务用来确定更高层的请求服务是否成功。

B) 协议参数设置服务

以下定义的服务，可以动态设置协议参数。

N_ChangeParameter.request: 这个服务是用来请求对具体内部参数的动态设置。

N_ChangeParameter.confirm: 这个服务用来确定更高层的更改一个特定协议的请求是否成功

4.3 网络层的内部操作

这个网络层的内部操作提供了用流量控制的方法分割、传输、重组数据。网络层最主要的目的就是要把数据转换成能适应或者不能适应单一 CAN 帧。不能适应单一 CAN 帧的信息被分割成多个部分，这里可以每一个都可以转换成单一 CAN 帧。

图 1，显示的是一个没有分割过的信息传输

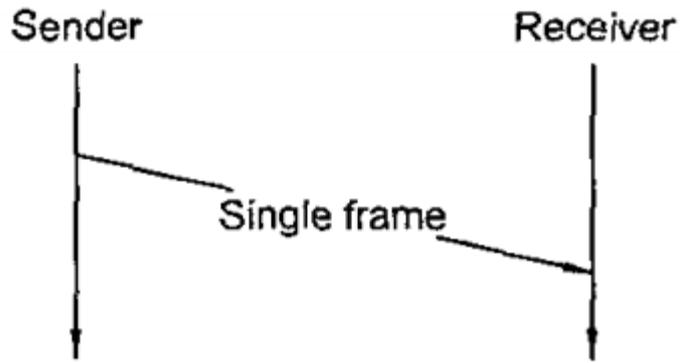


图 1

图 2 显示的是一个分割过的信息传输

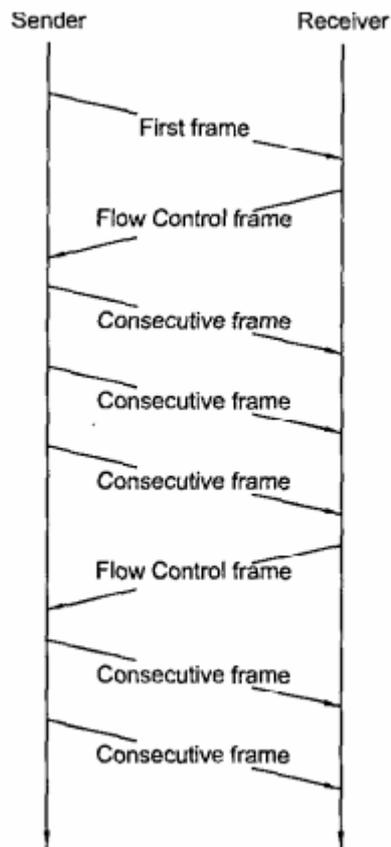


图 2

流量控制主要是用来调整发送器与网络层接收器相适应。这个流量控制方案允许使用诊断网关和子网络。

ISO15765 具体指定三种不同地址格式：标准的、扩展的和混合的。

5、网络层服务

5.1 概述

所有的网络层服务都具有相同的大体结构。为了定义这些服务，指定了三种基本的服务。

- A) 基本的请求服务，被更高的通信层或应用层用来传输控制信息和需要的数据到网络层。
- B) 基本的指示服务，被网络层用来传输状态信息和接收到的数据到上层的通信层或应用层。
- A) 基本的确认服务，被网络层用来传输状态信息到更高的通信层或应用层。

这项服务规范并没有指定一个应用程序编程接口，仅仅是指定一系列独立于任何设施的原始服务。

所有的网络层拥有同样的大体格式。原始的服务被写成以下的格式。

```

服务名.类型{
    参数 A,
    参数 B,
    参数 C,
    ...}

```

此处，“服务名”是服务的名称，例如，N_USData。“类型”指示基本的服务类型，“参数 A，参数 B...”，指网络数据单元被一个值列表被基本的服务所传递。这些基本的服务定义了一个服务用户（例如，诊断应用）怎么和一个服务提供者（例如网络层）进行合作。下列基本服务在这张国际标准表里做了具体的参数：请求，指示和确定。

- A) 使用基本的请求服务（service_name.request），服务用户向服务提供者请求一个服务。
- B) 使用基本的请示服务（service_name.indication），服务提供者通知服务用户关于网络层的内部事件。
- C) 基本的确定服务（service_name.confirm），服务提供者确认服务用户关于服务用户请求的结果。

5.2 网络层基本服务的规范

5.2.1 N_USData.request

这项基本服务请求传输带有“长度”字节的“信息数据”，从发送器到接收器，同时通过在 N_SA、N_TA、N_TAtype、和 N_AE 中地址信息进行硬件认证。每次只要 N_USData.request 发生了，网络层就要标识向服务用户传输信息成功（或失败），通过看 N_USData.confirm 这项服务是否被请求了。

```

N_USData.request{
    Mtype
    N_SA
    N_TA
    N_TAtype
    N_AE
    <MessageData>
    <Length>
}

```

5.2.2 N_USData.confirm

此服务是网络层发出。基本的服务确定 N_USData.request 是否已经完成，通过在 N_SA、N_TA、N_TAtype、和 N_AE 中地址信息进行认证。参数<N_Result>提

供服务请求的状态。

```
N_USData.confirm{
    Mtype
    N_SA
    N_TA
    N_TAtype
    N_AE
    <N_Result>
}
```

5.2.3 N_USData_FF.indication

此服务由网络层发出。基本的服务表示相邻上层接收到一个分段信息的第一帧，来自一个对等的协议硬件，通过在 N_SA、N_TA、N_TAtype、和 N_AE 中地址信息进行认证。

```
N_USData_FF.indication {
    Mtype
    N_SA
    N_TA
    N_TAtype
    N_AE
    <Length>
}
```

N_USData_FF.indication 通常由网络层请求，跟随在 N_USData.indication 后面，标识信息接收成功（或失败）。

此服务只有由网络层发出，并且接收到一个正确的第一帧分段信息。

如果网络层监测到第一帧的任何错误，然后这个信息会被网络层忽略，而且没有 N_USData_FF.indication 发出到相邻的上层。

如果网络层接收到一个数据长度值大于可接收缓冲区大小的第一帧，然后会被认为是一个错误的条件，同样没有 N_USData_FF.indication 发出到相邻的上层。

5.2.4 N_USData.indication

此服务也是由网络层发出。这项基本的服务指示<N_Result>事件和传递来自一个对等的协议硬件的带有“长度”字节的“信息数据”到相邻的上层，这些带有“长度”字节的“信息数据”通过在 N_SA、N_TA、N_TAtype、和 N_AE 中地址信息进行认证。

<MessageData>和<Length>参数仅仅在<N_Result>等于 N_OK 的情况下有效。

```
N_USData.indication{
    Mtype
    N_SA
    N_TA
    N_TAtype
    N_AE
    <MessageData>
    <Length>
```

<N_Result>

}

此项服务在接收到单一帧信息之后才发出，或者分段信息接收成功（或失败）的指示。

如果网络层监测到在单一帧中有任何错误，然后这些信息就会被网络层忽略，同时这项服务不会发出到相邻的上层。

5.2.5 N_ChangeParameters.request

此项服务是用来请求对局部协议硬件上的内部参数值进行修改。

参数修改总是可能的，除了接收到（N_USData_FF.indication）的第一帧之后和直到接收到（N_USData.indication）的通信信息的结束。

```
N_ChangeParameters.request {  
    Mtype  
    N_SA  
    N_TA  
    N_TAtype  
    N_AE  
    <Parameter>  
    <Parameter_Value>  
}
```

5.2.6 N_ChangeParameter.confirm

此项基本服务确认 N_ChangeParameter.Confirmation 服务申请信息的完成，通过在 N_SA、N_TA、N_TAtype、和 N_AE 中地址信息进行认证。

```
N_ChangeParameters.confirm {  
    Mtype  
    N_SA  
    N_TA  
    N_TAtype  
    N_AE  
    <Parameter>  
    <Result_ChangeParameter>  
}
```

5.3 服务数据单元规范

5.3.1 Mtype, 信息类型

类型： 枚举

范围： 诊断， 远程诊断

描述： 这个 Mtype 的参数用来确认地址信息参数的类型和范围，包括在一个服务请求。这部分的 ISO15765 为这个参数指定了两个数值范围。其目的是此文件的用户可以通过具体制定地址信息参数的组合和其他类型的地址信息参数来扩展上述的数值范围。对于每个新的地址信息的范围，一个新的 Mtype 参数的值会在被具体制定，并在新的地址信息中得到确认。

——如果 Mtype=诊断，地址信息 N_AI 将会包括 N_SA、N_TA 和 N_TAtype 这些参数。

——如果 Mtype=远程诊断，地址信息 N_AI 将会包括 N_SA、N_TA、N_TAtype 和 N_AE 这些参数。

5.3.2 N_AI 地址信息

5.3.2.1 N_AI 描述

这些参数用来做寻址信息。作为一个整体，N_AI 参数用来确认信息发送器和接收器的源地址 (N_SA)、目标地址(N_TA)，以及 N_TAtype 信息和可选地址扩展 (N_AE) 的通信模型。

5.3.2.2 N_SA ， 网络源地址

类型： 1 字节的无符号整型值

范围： 00—FF

描述： N_SA 参数用来编码发送的网络层协议硬件。

5.3.2.3 N_TA 网络目标地址

类型： 1 字节的无符号整型值

范围： 00—FF

描述： N_TA 参数用来编码接收的网络层协议硬件。

5.3.2.4 N_TAtype 网络目标地址类型

类型： 枚举

范围： 物理， 功能

描述： N_TAtype 参数是 N_TA 的延伸。N_TAtype 用来编码通信模型。这里指定了两种具体的通信模型：1 对 1 的通信，称为物理地址；1 对 n 的通信，称为功能地址。

——物理地址用来支持所有的网络层信息的类型。

——功能地址仅仅是用来支持通信的单一帧。

5.3.2.5 N_AE 网络地址延伸

类型： 1 字节的无符号整型值

范围： 00—FF

描述： N_AE 参数用来延伸大型网络的可用地址范围，并且可以用来编码子网络的网络层硬件的接收和发送，而不是局域网的，通信是在子网络上进行的。一旦 Mtype 设置成了远程诊断，N_AE 是寻址信息的唯一部分。

5.3.3 <Length>

类型： 12 位

范围： 1—4095

描述： 这个参数包括要传输和接收的数据的长度。

5.3.4 <MessageData>

类型： 字符串

范围：

描述： 这个参数包括所有更高层次硬件交换的数据。

5.3.5 <Parameter>

类型： 枚举

范围： STmin(separation time min)， BS(block size)

描述： 这个参数标识一个网络层参数。

5.3.6 <Parameter_Value>

类型： 1 字节的无符号整型值

范围： 0—255

描述： 这个参数分配到协议参数<Parameter>，此文件中服务部分有介绍的。

5.3.7 <N_Result>

类型： 枚举

范围： N_OK , N_TIMEOUT_A , N_TIMEOUT_Bs , N_TIMEOUT_Cr, N_WRONG_SN , N_INVALID_FS , N_UNEXP_PDU , N_WFT_OVRN, N_BUFFER_OVFLW, N_ERROR

描述： 这个参数包含有关服务执行结果的状态。一旦同时检测到两个或更多的错误，网络层硬件将会把错误列表中第一个发现的参数值给更高层。

——N_OK：这个值表示服务执行已经成功了，它可以发出给发送器和接收器两边的服务用户。

——N_TIMEOUT_A：这个值发出给协议用户，当计时器 N_Ar/N_As 超过了各自的限值，它可以发出给发送器和接收器两边的服务用户。

——N_TIMEOUT_Bs：这个值发出给协议用户，当计时器 N_Bs 超过了限值，它仅仅可以发出给发送器这边的服务用户。

——N_TIMEOUT_Cr：这个值发出给协议用户，当计时器 N_Cr 超过了限值，它仅仅可以发出给接收器这边的服务用户。

——N_WRONG_SN：只要有一个意外的系列号的值，这个参数就会发出给服务用户。它仅仅可以发出给接收器这边的服务用户。

——N_INVALID_FS：当在一个流动控制的 N_PDU 中接收到一个未知的或非法的流动状态值，这个参数就会发出给服务用户。它仅仅可以发出给发送器这边的服务用户。

——N_UNEXP_PDU：一旦接收到一个意外的协议数据单元，这个参数就会发出给服务用户。它仅仅可以发出给接收器这边的服务用户。

——N_WFT_OVRN：这个值会发出给服务用户，只要接收到的流量控制的 WAIT 帧超过其最大的计数值。

——N_BUFFER_OVFLW：这个值会发出给服务用户，只要接收到 FlowStatus=OVFLW 的流量控制 N_PDU。它仅仅可以发出给发送器这边的服务用户。

——N_ERROR：这是一个普通的错误值。当网络层检测到一个错误并且没有其他参数值能更好地描述这个错误，这个值会发出给服务用户。它可以发出给发送器和接收器两边的服务用户。

5.3.8 <Result_ChangeParameter>

类型： 枚举

范围： N_OK, N_RX_ON, N_WRONG_PARAMETER, N_WRONG_VALUE

描述： 这个参数包含有关服务执行结果的状态。

——N_OK：这个值代表服务执行已经成功完成了。它可以发送给发送器和接收器两边的服务用户。

——N_RX_ON：一旦接收到一个由<AI>发生来确定的信息，这个值发出给服务用户用来说明服务没有执行。它仅仅可以发送给在接收器这边的服务用户。

——N_WRONG_PARAMETER：由于未定义的<Parameter>，这个值会发出给服务用户用来确定服务没有执行。它可以发出给接收器和发送器两边的服务用户。

——N_WRONG_VALUE：由于超过范围的的<Parameter_Value>，这个值会发出给服务用户用来确定服务没有执行。它可以发出给接收器和发送器两边的服务用户。

6、网络层协议

6.1 协议功能

主要有以下两功能：一，传输和接收高达 4095 字节的数据信息；二，报告传输和接收成功（或失败）

6.2 单一帧传输

长达 6 或 7 字节数据的信息的传输，是通过传输一个独特的 N_PDU 来完成的。叫做 SF（单一帧），见图 3

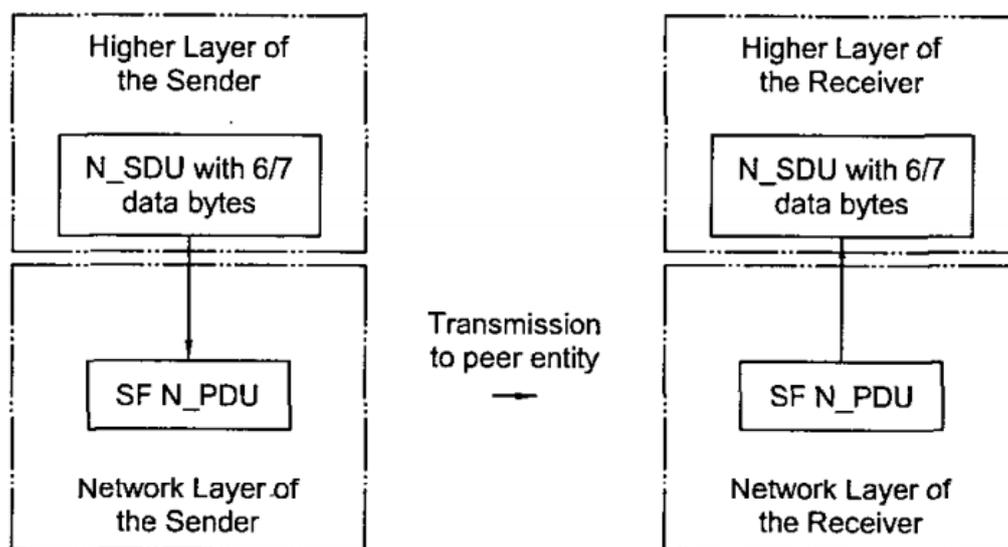


图 3

6.3 混合帧传输

传输更长字节的信息，是通过信息的分段和混合的 N_PDU 的传输来实现的。接收更长字节的信息，是通过混合的 N_PDU 的接收和接收数据字节的集合来实现的。混合的 N_PDU 被称为第一帧（相对信息的第一个 N_PDU）和连续帧（相对接下来的所有 N_PDU）。

接收混合 N_PDU 的接收器有能力去适应传输的吞吐量，通过使用流量控制协议数据单元（FC N_PDU）的流量控制机制。

超过 6 或 7 字节的数据信息被分割成：一，第一帧协议数据单元（FF N_PDU），包含 5 个字节或者 6 个字节；二，一个或多个连续帧协议数据单元，包含 6 个或 7 个字节的数据，CF N_PDU 仅仅包含剩余的字节数据，因此可能少于 6 个或 7 个字节。

图 4 描述的是，发送端的信息分割和接收端的信息整合

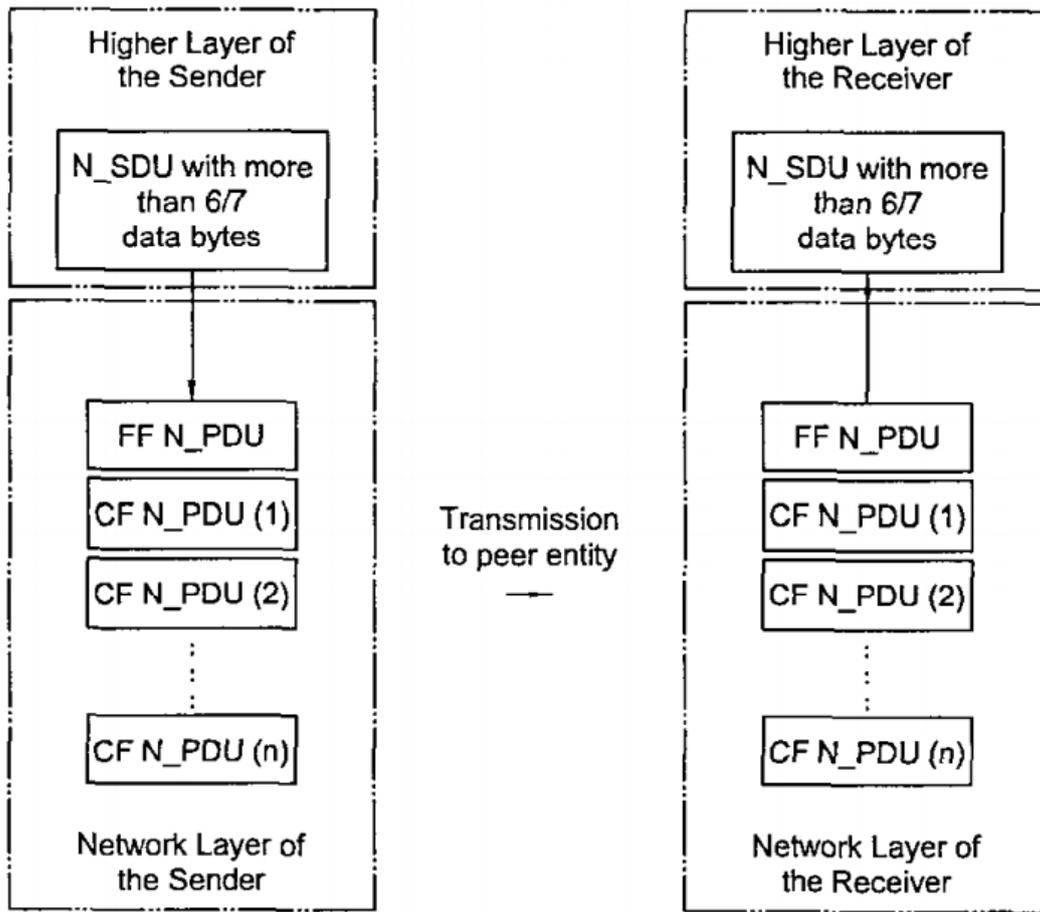


图 4 (注: 用来响应 FF N_PDU 接收的, 由接收器发出的 FC N_PDU 没有显示) 信息长度在 FF N_PDU 中传输。所有的 CF N_PDU 被发送器标号, 以便在接收器那边以相同的序列重新整合。

流量控制机制 (见图 5) 允许接收器向发送器通知其能力。不同的节点会有不同的能力, 流量控制由接收器向发送器通知其能力。发送器应该符合接收器的能力。这些能力定义如下: 一, 块大小 (BlockSize Bs), 接收器允许发送器发送的最大数字的 N_PDUs, 在等待授权继续传输接下来的 N_PDUs 之前。二, 最小分段时间 (SeparationTimeMin STmin), 发送器在两个 CF N_PDUs 之间传输的最小等待时间。

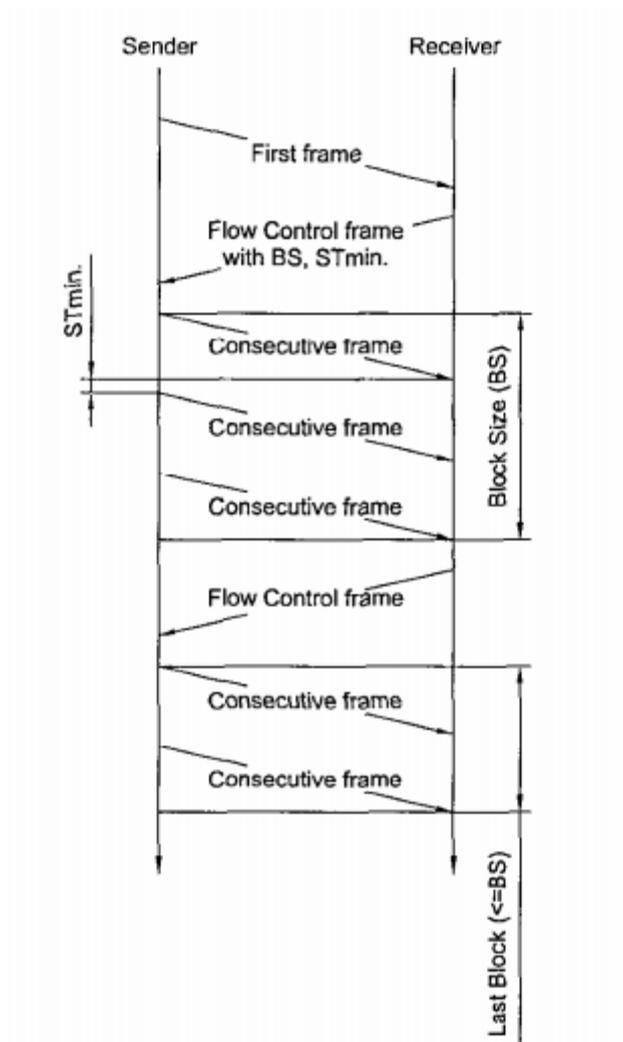


图 5 流体控制机制

所有的块，除了最后一个，都由 N_PDUS 的 BS 组成。最后一个包含剩余的 $N_PDUS(1$ 到 $BS)$ 。

每次发送器/接收器等待一个来自接收器/发送器的 N_PDU ，一个超时机制允许检测传输失败与否。

借助 $FC\ N_PDUs$ ，接收器有能力授权接下来的 $CF\ N_PDUs$ 的传输与否，如果即将被传输的字节数超过了接收器缓存区所能容纳的字节数，就延迟传输的授权或者否定接收分割信息的接收。

—— $FC.CTS$: continue to send, 授权连续发送

—— $FC.WAIT$: 请求继续等待

—— $FC.OVFLW$: 缓存溢出，标识在分割信息中第一帧中具体指定的字节数超过接收器硬件中缓存区所能储存的字节数。

$FC.WAIT$ 的数字有一个最大值的限制，接收器能够用列发送，这个最大值就是 N_WFTmax 。这个参数是一个系统常量，不在第一个 $FC\ N_PDU$ 中传输的。

6.4 网络层协议数据单元 (N_PDU)

6.4.1 协议数据单元类型

网络层的对等协议硬件不同节点间的通信是通过交换 N_PDUs 来实现的。

这部分 $ISO15765$ 指定了四种不同类型的网络层协议数据单元，单一帧 (SF

N_PDU)、第一帧(FF N_PDU)、连续帧(CF N_PDU)和流体控制帧(FC N_PDU),它们用来在网络层硬件中建立通信的路径,达到交换通信参数、传输用户数据和释放通信资源的功能。

6.4.2 SF N_PDU

SF N_PDU 被单一帧协议控制信息 (SF N_PCI) 识别。SF N_PDU 能够被网络硬件发出去,也能被一个或多个网络硬件所接收。它被发出去转换成一个服务数据单元,也能转换成一个没有无节的信息,服务数据单元通过一个单一信息请求传输到数据链路层。

6.4.3 FF N_PDU

FF N_PDU 能够被第一帧协议控制信息 (FF N_PCI) 所识别。FF N_PDU 能够被网络硬件发送出去,也能够被独特的网络硬件接收。FF N_PDU 识别分段信息的第一个 N_PDU,这些分段信息是被网络发送硬件传输和网络接收硬件所接收。

6.4.4 CF N_PDU

CF N_PDU 能够被连续帧协议控制信息 (CF N_PCI) 所识别。CF N_PDU 转换成服务数据单元的信息数据 (<MessageData>) 的分段信息。所有的 N_PDU 在 FF N_PDU 编码为 CF N_PDU 之后被发送硬件传输。在最后一个 CF N_PDU 已经接收到之后,接收硬件传递整合的信息到网络接收硬件的服务用户。CF N_PDU 能够被网络发送硬件发出去,也能够被一个独特的网络接收硬件所接收。

6.4.5 FC N_PDU

FC N_PDU 能够被流体控制协议信息所识别。FC N_PDU 指示网络发送硬件开始/停止/恢复 CF N_PDU 的传输。当准备好开始接收更多数据时,FC N_PDU 能够被网络层硬件发到网络层发送硬件,同时还需要满足一个条件:接收到正确的 FF N_PDU 和最后一个 CF N_PDU。

6.4.6 协议数据单元的描述

6.4.6.1 N_PDU 的格式

协议数据单元 (N_PDU),能够在处于一个节点的网络层与其他一个或多个节点的网络层之间传输。所有的 N_PDU 包含下面 3 块,地址信息 (N_AI)、协议控制信息 (N_PCI) 和数据段 (N_Data)。

6.4.6.2. 地址信息 (N_AI)

N_AI 用来识别网络层的通信硬件。N_AI 信息在 N_SDU、N_SA、N_TA、N_TAtype、

N_AE 中接收,N_AI 能够被复制和包含在 N_PDU 中。如果在 N_SDU 中接收到的信息数据 (<MessageData>和<Length>) 太长以至于需要网络层分段来进行传输这个完整的信息,那么 N_AI 就会被复制或重复在每一个需要传输的 N_PDU 中。

地址信息识别交换/接收/发送的数据类型。

6.4.6.3 协议控制信息 N_PCI

N_PCI 识别用来交换的 N_PDU 的类型。同样,它可以用来交换其他网络通信层硬件间的控制参数。

6.4.6.4 数据段 N_Data

N_PDU 中的 N_Data 用来发送服务用户数据,这些数据在 N_USData.request 服务中的<MessageData>中被接收。如果需要,<MessageData>会被分割成更小的部分以适应每一个 N_PDU 的数据段。

N_Data 的大小取决于 N_PDU 的类型和地址格式的选择。

6.5 协议控制信息规范

6.5.1 N_PCI

每一个 N_PDU 都能被一个 N_PCI 识别。

N_PDU name	N_PCI bytes			
	Byte #1		Byte #2	Byte #3
	Bits 7 - 4	Bits 3 - 0		
SingleFrame (SF)	N_PCType = 0	SF_DL	N/A	N/A
FirstFrame (FF)	N_PCType = 1	FF_DL		N/A
ConsecutiveFrame (CF)	N_PCType = 2	SN	N/A	N/A
FlowControl (FC)	N_PCType = 3	FS	BS	STmin

图 6 N_PCI 字节摘要

Hex value	Description
0	SingleFrame For unsegmented messages, the network layer protocol provides an optimized implementation of the network protocol with the message length embedded in the PCI byte only. SingleFrame (SF) shall be used to support the transmission of messages that can fit in a single CAN frame.
1	FirstFrame A FirstFrame (FF) shall only be used to support the transmission of messages that cannot fit in a single CAN frame, i.e. segmented messages. The first frame of a segmented message is encoded as an FF. On receipt of an FF, the receiving network layer entity shall start assembling the segmented message.
2	ConsecutiveFrame When sending segmented data, all consecutive frames following the FF are encoded as ConsecutiveFrame (CF). On receipt of a CF, the receiving network layer entity shall assemble the received data bytes until the whole message is received. The receiving entity shall pass the assembled message to the adjacent upper protocol layer after the last frame of the message has been received without error.
3	FlowControl The purpose of FlowControl (FC) is to regulate the rate at which CF N_PDUs are sent to the receiver. Three distinct types of FC protocol control information are specified to support this function. The type is indicated by a field of the protocol control information called FlowStatus (FS), as defined hereafter.
4 - F	Reserved This range of values is reserved by this part of ISO 15765.

图 7 N_PCItyp e 值的定义

6.5.2 单一帧 N_PCI 参数的定义

6.5.2.1 SF N_PCI byte

如下图所示

N_PDU name	SF N_PCI byte							
	Byte #1							
	7	6	5	4	3	2	1	0
SingleFrame	0	0	0	0	SF_DL			

参数单一帧数据长度 (SF_DL) 用在 SF N_PDU 中以指明服务用户数据字节数,

如下图所示

Hex value	Description
0	Reserved This value is reserved by part of ISO 15765.
1-6	SingleFrame DataLength (SF_DL) The SF_DL is encoded in the low nibble of N_PCI byte #1 value. It shall be assigned the value of the service parameter <Length>.
7	SingleFrame DataLength (SF_DL) with normal addressing SF_DL = 7 is only allowed with normal addressing.
8-F	Invalid This range of values is invalid.

6.5.2.2 SF_DL 错误处理

如果网络层接收到一个带有 SF_DL=0 的 SF，那么网络层就会忽视掉这个接收到的 SF N_PDU。

如果网络层接收到一个使用标准地址时的 SF_DL>7 的 SF，或者使用扩展/混合地址是 SF_DL>6 的 SF，那么网络层就会忽视掉这个接收到的 SF N_PDU。

6.5.3 第一帧 N_PCI 参数定义

6.5.3.1 FF N_PCI bytes

N_PDU name	FF N_PCI bytes								
	Byte #1							Byte #2	
	7	6	5	4	3	2	1	0	7-0
FirstFrame	0	0	0	1					FF_DL

6.5.3.2 第一帧数据长度 (FF_DL) 参数定义

FF_DL 用在 FF N_PDU 中用来指明服务用户数据字节数。

Hex value	Description
0-6	Invalid This range of values is invalid.
7	FirstFrame DataLength (FF_DL) with extended addressing or mixed addressing FF_DL = 7 is only allowed with extended or mixed addressing.
8-FFF	FirstFrame DataLength (FF_DL) The encoding of the segmented message length results in a twelve (12) bit length value (FF_DL) where the least significant bit (LSB) is specified to be bit "0" of the N_PCI byte #2 and the most significant bit (MSB) is bit three (3) of the N_PCI byte #1. The maximum segmented message length supported is equal to 4 095 bytes of user data. It shall be assigned the value of the service parameter <Length>.

6.5.3.3 FF_DL 错误处理

如果网络层接收到一个带有 FF_DL 大于接收器缓存区可容纳的大小，那么就会被认为是一个错误的条件。网络层就会退出信息的接收，并且发送一个带有参数 FlowStatus=Overflow 的 FC N_PDU。

如果网络层接收到一个使用标准地址时的 FF_DL<8 的 FF，或者使用扩展/混合

地址是 FF_DL<7 的 FF，那么网络层就会忽视掉这个接收到的 FF N_PDU。

6.5.4 连续帧 N_PCI 参数定义

6.5.4.1 CF N_PCI byte

N_PDU name	CF N_PCI byte							
	Byte #1							
	7	6	5	4	3	2	1	0
ConsecutiveFrame	0	0	1	0	SN			

6.5.4.2 序列号 (SN) 参数定义

这个参数 SN 是用在 CF N_PDU 中，用来指明连续帧的顺序。

——SN 对所有分段信息都是从 0 开始。FF 被分配这个 0 值。它在 N_PCI 中不包括一个明确的序列号，但是被认为是一个分段的数值 0。

——第一个 CF 立即接着一个 FF 的 SN，应该把 SN 设置为 1。

——每一个新的 CF 在一个分段信息传输期间被发送了，SN 自动+1。

——SN 的值不会被任何 FC 帧所影响。

——SN 的值达到 15 以后，置 0 为下个 CF 做准备。

N_PDU	FF	CF						
SN (hex)	0	1		E	F	0	1	

Hex value	Description
0-F	SequenceNumber (SN) The SequenceNumber (SN) shall be encoded in the lower nibble bits of N_PCI byte #1. The SN shall be set to a value within the range of zero (0) to fifteen (15).

6.5.4.3 SN 错误处理

如果一个 CF N_PDU 信息接收到一个错误的序列号，将会启动正常的错误处理。退出接收信息的过程，网络层发送一个带有参数 <N_Result>=N_WRONG 的 N_USData.indication 服务申请到相邻的上层。

6.5.5 流体控帧 N_PCI 参数定义

6.5.5.1 FlowControl N_PCI bytes

N_PDU name	FC N_PCI bytes									
	Byte #1								Byte #2	Byte #3
	7	6	5	4	3	2	1	0		
FlowControl	0	0	1	1	FS			BS	STmin	

6.5.5.2 流动状态 (FS) 参数定义

FS 参数说明网络发送硬件是否能继续传输信息。网络发送硬件应该能支持所有的 FS 参数的指定（不是保留的）值。

Hex value	Description
0	ContinueToSend (CTS) The FlowControl ContinueToSend parameter shall be encoded by setting the lower nibble of the N_PCI byte #1 to "0". It shall cause the sender to resume the sending of Consecutive frames. The meaning of this value is that the receiver is ready to receive a maximum of BS number of Consecutive frames.
1	Wait (WT) The FlowControl Wait parameter shall be encoded by setting the lower nibble of the N_PCI byte #1 to "1". It shall cause the sender to continue to wait for a new FlowControl N_PDU and to restart its N_BS timer.
2	Overflow (OVFLW) The FlowControl Overflow parameter shall be encoded by setting the lower nibble of the N_PCI byte #1 to "2". It shall cause the sender to abort the transmission of a segmented message and make an N_USData.confirm service call with the parameter <N_Result>=N_BUFFER_OVFLW. This N_PCI FlowStatus parameter value is only allowed to be transmitted in the FlowControl N_PDU that follows the FirstFrame N_PDU and shall only be used in case the message length FF_DL of the received FirstFrame N_PDU exceeds the buffer size of the receiving entity.
3 - F	Reserved This range of values is reserved by this part of ISO 15765.

6.5.5.3. FS 错误处理

如果接收到一个带有无效（或保留）的 FS 参数的 FC N_PDU 信息，网络层将会启动正常的错误处理。退出传输信息的过程，网络层发送一个带有参数 <N_Result>=N_INVALID_FS 的 N_USData.confirm 服务申请到相邻的上层。

6.5.5.4 块大小（BS）参数定义

BS 参数用 FC N_PCI 中第二字节进行编码。

BS 单元是每一块 CF N_PDUs 中的绝对数量，例如，如果 BS 等于十进制的 20，那么这块中将包含 20 个（十进制）CF N_PDUs。

在分段数据传输中，仅仅是连续帧的最后一块可能含有少于 BS 数量的帧。

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) This 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.

6.5.5.5 最小分段时间（STmin）参数定义

STmin 参数 FC N_PCI 中第三字节进行编码。

时间是被接收硬件指定的，能够被发送网络硬件所保存，为了保证分度信息的持续发送。

STmin 参数值指明连续帧网络协议数据单元之间传输的最小时间。

Hex value	Description
00 – 7F	SeparationTime (STmin) range: 0 ms – 127 ms 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: 100 μs – 900 μs The units of STmin in the range F1 hex – F9 hex are even 100 microseconds (μ s), where parameter value F1 hex represents 100 μ s and parameter value F9 hex represents 900 μ s.
FA – FF	Reserved This range of values is reserved by this part of ISO 15765.

STmin 的计算，是从传输一个 CF 的结束开始，到请求下一个 CF 的时候结束。

6.5.5.6 ST 错误处理

如果接收到一个带有保留的 ST 参数值的 FC N_PDU 信息，那么网络发送硬件就会使用在 ISO15765 中指明的最大的 ST 值 (7Fhex-127ms) 来代替网络接收硬件为保证连续进行分段信息传输的接收到的值。

6.6 FC.Wait 帧传输的最大数值 (N_FWTmax)

这个变量的目的是避免通信发送节点因为一个错误的条件阻止了，那样下一个能够继续等待。这个参数放在局部通信中，不会被传输，因此不是 FC 协议数据单元的一部分。

——N_FWTmax 参数用来说明一行中被接收器传输的 FC N_PDU WTs 有多少个。

——N_FWTmax 参数的上限应该在系统生效时间内由用户自行定义。

——N_FWTmax 参数在信息传输过程中，仅仅能被网络接收硬件应用。

——如果 N_FWTmax 参数被置 0，流量控制应该依靠流量控制继续 (flow control continue) 发送 FC N_PDU CTS，流量控制等待 (Flow control wait FC N_PDU WT) 不会再网络层中应用。

6.7 网络层时序

6.7.1 时序参数

图 8 表明网络层时序参数，表 1 定义了网络层时序参数值和它们基于数据链路层服务的响应的开始和结束的位置。

性能要求值必须具有通信需求的约束，来和每一个通信地方匹配，为了和这个规范兼容。某个应用应该定义具体的性能要求，只要在表 1 定义的范围内。

超时数值定义的比这个值要高，因为性能要求值为了保证系统正常工作和克服性能要求绝对不能达到的通信条件。具体的超时数值对任何给定的条件，都当做是一个下限。实际的超时发生在不迟于具体超时值的 50% 以上。

网络层应该发布一个正确的基本服务，可以是网络层服务用户检测到错误的条件。

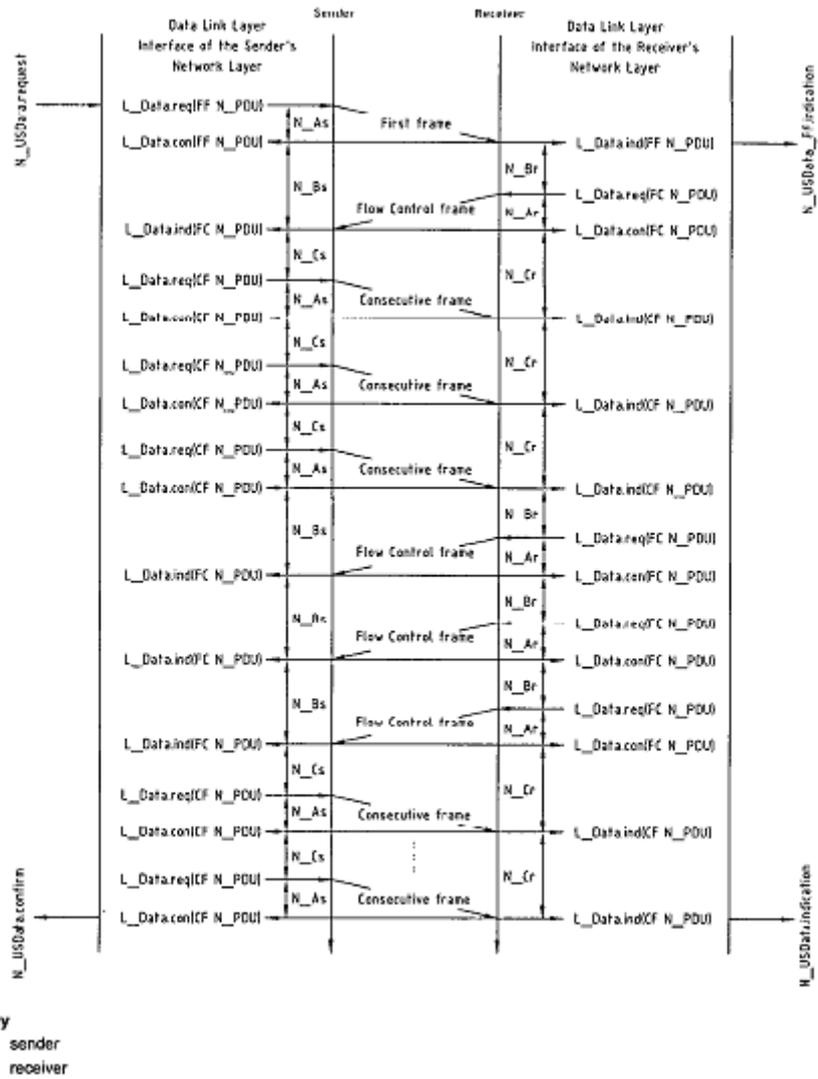


图 8 网络层时序参数的分布

Timing Parameter	Description	Data Link Layer service		Timeout (ms)	Performance requirement (ms)
		Start	End		
N_As	Time for transmission of the CAN frame (any N_PDU) on the sender side	L_Data.request	L_Data.confirm	1 000	N/A
N_Ar	Time for transmission of the CAN frame (any N_PDU) on the receiver side	L_Data.request	L_Data.confirm	1 000	N/A
N_Bs	Time until reception of the next FlowControl N_PDU.	L_Data.confirm (FF), L_Data.confirm (CF), L_Data.indication (FC)	L_Data.indication (FC)	1 000	N/A
N_Br	Time until transmission of the next FlowControl N_PDU	L_Data.indication (FF), L_Data.confirm (FC)	L_Data.request (FC)	N/A	$(N_{Br} + N_{Ar}) < (0.9 * N_{Bs} \text{ timeout})$
N-Cs	Time until transmission of the next ConsecutiveFrame N_PDU	L_Data.indication (FC), L_Data.confirm (CF)	L_Data.request (CF)	N/A	$(N_{Cs} + N_{As}) < (0.9 * N_{Cr} \text{ timeout})$
N_Cr	Time until reception of the next ConsecutiveFrame N_PDU	L_Data.confirm (FC), L_Data.indication (CF)	L_Data.indication (CF)	1 000	—

s is the sender of the message.
r is the receiver of the message.

表 1 网络层时序参数的数值

6.7.2 网络层超时

以下定义的网络超时的原因和采取的操作

Timeout	Cause	Action
N_As	Any N_PDU not transmitted in time on the sender side.	Abort message transmission and issue N_USData.confirm with <N_Result> = N_TIMEOUT_A.
N_Ar	Any N_PDU not transmitted in time on the receiver side.	Abort message reception and issue N_USData.indication with <N_Result> = N_TIMEOUT_A.
N_Bs	FlowControl N_PDU not received (lost, overwritten) on the sender side or preceding FirstFrame N_PDU or ConsecutiveFrame N_PDU not received (lost, overwritten) on the receiver side.	Abort message transmission and issue N_USData.confirm with <N_Result> = N_TIMEOUT_Bs.
N_Cr	ConsecutiveFrame N_PDU not received (lost, overwritten) on the receiver side or preceding FC N_PDU not received (lost, overwritten) on the sender side.	Abort message reception and issue N_USData.indication with <N_Result> = N_TIMEOUT_Cr.

6.7.3 意外到来的 N_PDU (Unexpected arrival of N_PDU)

意外到来的 N_PDU 是指一个节点在预期顺序 N_PDU 之外接收到的 N_PDU。它是 ISO15765 定义的 N_PDU (SF N_PDU、FF N_PDU、CF N_PDU 或 FC N_PDU)，不是标准期待顺序接收到的，或者它是一个未知的 N_PDU，这样它就不能够用本文件中的定义来解析。

根据网络层设计通信中的全双工和半双工，这个不能解析的“未知”，可以分成两类：在半双工中，在两个节点中的点对点的通信，只能往一个方向；在全双工中，两个节点中的点对点的通信，可以往同时往两个方向。

网络层的设计，必须考虑到一个从一个节点接收或传输带有相同地址信息 (N_AI)，也要考虑到接收到一个意外到来的 N_PDU。

一个大致的原则，不论从任何一个节点来的意外到来 N_PDU 将会被忽略，除了

SF N_PDUs 和物理地址的 FF N_PDUs; 功能上, 地址第一帧也会被忽略。当对摸个意外到来的 N_PDU 采取具体操作时, 意味这网络层不会通知上层它的到来。下表定义了网络层应对意外到来的 N_PDU 的具体操作, 需要考虑实际网络层状态 (NWL status) 和设计中的半双工与全双工通信。

NWL status	Reception of				
	SF N_PDU	FF N_PDU	CF N_PDU	FC N_PDU	Unknown N_PDU
Segmented Transmit in progress	Full-duplex: If a reception is in progress, see corresponding cell below in this table, otherwise process the SF N_PDU as the start of a new reception.	Full-duplex: If a reception is in progress, see corresponding cell below in this table, otherwise process the FF N_PDU as the start of a new reception.	Full-duplex: If a reception is in progress, see corresponding cell below in this table.	If awaited, process the FC N_PDU, otherwise ignore it.	Ignore
	Half-duplex: ignore	Half-duplex: ignore	Half-duplex: ignore		
Segmented Receive in progress	Terminate the current reception, report an N_USData.indication, with <N_Result> set to N_UNEXP_PDU, to the upper layer, and process the SF N_PDU as the start of a new reception.	Terminate the current reception, report an N_USData.indication, with <N_Result> set to N_UNEXP_PDU, to the upper layer, and process the FF N_PDU as the start of a new reception.	If awaited, process the CF N_PDU in the on-going reception and perform the required checks (e.g. SN in right order), otherwise ignore it.	Full-duplex: If a transmission is in progress, see corresponding cell above in this table.	Ignore
				Half-duplex: ignore	
Idle ^a	Process the SF N_PDU as the start of a new reception.	Process the FF N_PDU as the start of a new reception.	Ignore	Ignore	Ignore

^a Neither a segmented transmission nor segmented reception is in progress.

6.7.4 等待帧错误处理

当接收器传输 N_WFTmax 流量控制等待网络协议数据单元 (FC N_PDU WT) 在一列中, 同时它不能达到传输流量控制清除网络协议数据单元 (FC N_PDU CTS) 的性能要求, 接收器这边退出信息接收同时向更高层发出一个<N_Result> 设置为 N_WFT_OVRN 的 N_USData.indication。

发送器通知退出信息接收, 通过一个<N_Result>设置为 N_TIMEOUT_Bs 的 N_USData.confirm。

6.8 信息的交错 (interleaving of messages)

网络层协议可以达到不同的信息地址的不同信息的并行传输。需要确定接收器是否能够采用一直的方式来集合接收到的网络协议数据单元。

7、数据链路层的应用

7.1 数据链路层接口服务

7.1.1 L_Data.request

此项基本服务请求传输<Data>, 这些<Data>映射到没有具体属性的数据链路协议数据单元, 通过<Identifier>来选择。

<Identifier>可以提供具体的地址格式来传输<Data>:

```

L_Data.request{
    <Identifier>
    <DLC>
    <Data>
}

```

7.1.2 L_Data.confirm

此项基本服务用来确认一个用来请求具体<Identifier>的 L_Data.request 是否完成。

<TransferStatus>参数提供请求服务的状态。

```

L_Data.confirm{
    <Identifier>
    <TransferStatus>
}

```

7.1.3 L_Data.indication

此项基本服务标识数据链路层事件到相邻上层和提供经<Identifier>鉴定的<Data>。

```

L_Data.indication {
    <Identifier>
    <DLC>
    <Data>
}

```

7.2 数据链路层服务参数

ISO 11898-1 中定义了以下数据链路层服务参数。

<Identifier>	CAN identifier	CAN 标识符
<DLC>	Data Length Code	数据长度代码
<Data>	CAN frame data	CAN 帧数据
<TransferStatus>	Status of a transmission	传输状态

7.3 Mapping of the N_PDU fields

7.3.1 地址格式

网络层数据交换支持三种地址格式：标准、扩展和混合的。每一种地址格式需要不同的 CAN 帧数据字节来封装地址信息。因此，带有单一 CAN 帧的数据字节数由选择的地址格式来决定。

7.3.2 标准地址

对每一个 N_SA、N_TA、N_TAtype 和 Mtype 的组合，都会分配一个单独的 CAN 标识符。N_PCI 和 N_Data 放置在 CAN 帧数据领域。

N_PDU type	CAN Identifier	CAN frame data field							
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
SingleFrame (SF)	N_AI	N_PCI	N_Data						
FirstFrame (FF)	N_AI	N_PCI		N_Data					
ConsecutiveFrame (CF)	N_AI	N_PCI	N_Data						
FlowControl (FC)	N_AI	N_PCI			N/A				

7.3.3 标准固定的地址

标准固定的地址是标准地址的一种子格式，更好地定义了地址信息映射到 CAN 标识符。上述的大多数标准地址，在 N_AI 和 CAN 标识符之间的响应是开放的。对于标准固定的地址，仅可以使用 29 位 CAN 标识符。以下两表格定义了 N_AI 和 CAN 标识符之间的映射，由目标地址（N_TAtype）决定的。N_PCI 和 N_Data 放置在 CAN 标识符中。

N_PDU type	29 bit CAN Identifier bit position							CAN frame data field byte position							
	28 ... 26	25	24	23 ... 16	15	8	7 ... 0	1	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin)	0	0	218 (dec)	N_TA	N_SA	N_PCI	N_Data							
FirstFrame (FF)	110 (bin)	0	0	218 (dec)	N_TA	N_SA	N_PCI		N_Data						
ConsecutiveFrame (CF)	110 (bin)	0	0	218 (dec)	N_TA	N_SA	N_PCI		N_Data						
FlowControl (FC)	110 (bin)	0	0	218 (dec)	N_TA	N_SA	N_PCI			N/A					

N_PDU type	29 bit CAN Identifier bit position							CAN frame data field byte position							
	28 ... 26	25	24	23 ... 16	15	8	7 ... 0	1	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin)	0	0	219 (dec)	N_TA	N_SA	N_PCI	N_Data							
FirstFrame (FF)	110 (bin)	0	0	219 (dec)	N_TA	N_SA	N_PCI		N_Data						
ConsecutiveFrame (CF)	110 (bin)	0	0	219 (dec)	N_TA	N_SA	N_PCI		N_Data						
FlowControl (FC)	110 (bin)	0	0	219 (dec)	N_TA	N_SA	N_PCI			N/A					

7.3.4 扩展地址

对每一个 N_SA、N_TA、N_TAtype 和 Mtype 的组合，都会分配一个单独的 CAN 标识符。N_TA 放置在 CAN 标识符的第一个数据字节。N_PCI 和 N_Data 放置在 CAN 标识符剩余的字节中。

N_PDU type	CAN Identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
SingleFrame (SF)	N_AI, except N_TA	N_TA	N_PCI	N_Data					
FirstFrame (FF)	N_AI, except N_TA	N_TA	N_PCI		N_Data				
ConsecutiveFrame (CF)	N_AI, except N_TA	N_TA	N_PCI	N_Data					
FlowControl (FC)	N_AI, except N_TA	N_TA	N_PCI			N/A			

7.3.5 混合地址

7.3.5.1 29 位 CAN 标识符

如果 Mtype 设置为远程诊断，混合地址格式才会被使用。

以下两表格定义了地址信息(N_AI)映射到 29 位 CAN 标识符列表和第一个 CAN 帧的数据字节，由目标地址类型决定。N_PCI 和 N_Data 放置在剩余的 CAN 帧数据区域。

N_PDU type	29 bit CAN Identifier bit position							CAN frame data field byte position							
	28 ... 26	25	24	23 ... 16	15	8	7 ... 0	1	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin)	0	0	206 (dec)	N_TA	N_SA	N_AE	N_PCI	N_Data						
FirstFrame (FF)	110 (bin)	0	0	206 (dec)	N_TA	N_SA	N_AE	N_PCI		N_Data					
ConsecutiveFrame (CF)	110 (bin)	0	0	206 (dec)	N_TA	N_SA	N_AE	N_PCI		N_Data					
FlowControl (FC)	110 (bin)	0	0	206 (dec)	N_TA	N_SA	N_AE	N_PCI			N/A				

N_PDU type	29 bit CAN Identifier bit position							CAN frame data field byte position							
	28 ... 26	25	24	23 ... 16	15	8	7 ... 0	1	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin)	0	0	205 (dec)	N_TA	N_SA	N_AE	N_PCI	N_Data						
FirstFrame (FF)	110 (bin)	0	0	205 (dec)	N_TA	N_SA	N_AE	N_PCI	N_Data						
ConsecutiveFrame (CF)	110 (bin)	0	0	205 (dec)	N_TA	N_SA	N_AE	N_PCI	N_Data						
FlowControl (FC)	110 (bin)	0	0	205 (dec)	N_TA	N_SA	N_AE	N_PCI	N/A						

7.3.5.2 11 位 CAN 标识符

当 Mtype 设置为远程诊断，混合地址才是会使用的地址格式。

以下的表格定义了地址信息 (N_AI) 和 11 位 CAN 标识符列表之间的映射。对每一个 N_SA、N_TA、N_TAtype 和 Mtype 的组合，都会分配一个单独的 CAN 标识符。N_AE 放置在 CAN 帧数据区域的第一个数据字节。N_PCI 和 N_Data 放置在剩余的 CAN 帧数据区域中。

N_PDU type	CAN Identifier	CAN frame data field byte position							
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
SingleFrame (SF)	N_AI	N_AE	N_PCI	N_Data					
FirstFrame (FF)	N_AI	N_AE	N_PCI	N_Data					
ConsecutiveFrame (CF)	N_AI	N_AE	N_PCI	N_Data					
FlowControl (FC)	N_AI	N_AE	N_PCI				N/A		

7.4 CAN 帧数据长度代码 (DLC)

7.4.1 DLC 参数

DLC 参数具体制定 CAN 帧中传输的数据字节数。ISO 15765 中这部分没有具体指定任何关于一个 CAN 帧中数据长度的需求，而是那些在网络层协议数据单元中提到过的大小。

本文件中针对网络层的应用布什让 CAN 帧用它们的全部长度填充或优化 DLC 使之达到与网络层协议数据单元适合的长度。

7.4.2 CAN 帧数据填充 (CAN frame data padding)

DLC 通常设为 8。如果要被传输的 N_PDU 少于 8 字节，那么发送器不得不设置 DLC 到最大值 8 (用没有使用的数据字节填充)。特别地，这个在 SF、FC 帧或分段信息中的最后一个 CF 中发生。

CAN 帧的 DLC 参数被发送器设置，被接收器接收后决定发送到网络层的数据字节数。DLC 参数不能用来决定信息长度，它可以从一个信息中的开端 N_PCI 信息中提取出来。

7.4.3 CAN 帧数据操作

DLC 不是总需要是 8。如果将要被传输的 N_PDU 短于 8 个字节，发送器可以操作 CAN 总线负载，通过缩短 CAN 帧数据为仅包含被 N_PDU 占据的字节数。

CAN 帧数据操作仅仅能被应用在 SF、FC 帧或分段信息中的最后一个 CF。

CAN 帧的 DLC 参数被发送器设置，被接收器接收后决定发送到网络层的数据字节数。DLC 参数不能用来决定信息长度，它可以从一个信息中的开端 N_PCI 信息中提取出来。

7.4.4 DLC 错误处理

在一个接收到的 CAN 帧中，网络层能够计算最小的 CAN DLC 参数期望值，由 N_PCI 的值决定。

如果没有进一步的操作，接收到 DLC 值小于期望的 CAN 帧会被网络层所忽略。