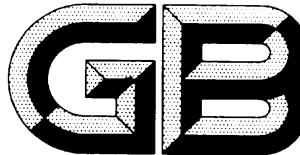


ICS

点击此处添加中国标准文献分类号



中华人民共和国国家标准

GB/T XXXXX—XXXX/ISO 14229

道路车辆 统一诊断服务(UDS)

Road vehicles - Unified diagnostic services

(ISO 14229, MOD)

(征求意见稿)

(本稿完成日期: 2020. 01. 15)

XXXX-XX-XX 发布

XXXX-XX-XX 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前言.....	II
引言.....	III
1 范围.....	1
2 规范性引用文件.....	1
3 术语、定义和缩略语.....	1
4 约定.....	1
5 概述.....	1
6 诊断网络架构.....	4
7 DoCAN 用例概述和准则.....	4
8 DoCAN 用例定义.....	4
参考文献.....	6

前　　言

本部分按照GB/T 1.1—2009给出的规则起草。

本部分使用翻译法修改采用ISO 14229《道路车辆 统一诊断服务(UDS)》系列标准，包含ISO 14229的下列部分：

- 第1部分：规范和要求；
- 第2部分：会话层服务；
- 第3部分：CAN的统一诊断设备(UDSonCAN)；
- 第4部分：FlexRay实施的统一诊断服务(UDSonFR)；
- 第5部分：IP实施的统一诊断服务(UDSonIP)；
- 第6部分：K-Line实施的统一诊断服务(UDSonK-Line)；
- 第7部分：局域互连网络(UDSonLIN)上的UDS。

本部分与ISO 14229的技术性差异如下：

- 修改了本部分的适用范围；
- 关于标准中的引用文件，本部分做了具有技术性差异的调整：
 - 用等同采用国际标准的GB/T XXXX代替ISO 14229系列标准；

本部分由中华人民共和国工业和信息化部提出。

本部分由全国汽车标准化技术委员会(SAC/TC114)归口。

本部分起草单位：。

本部分主要起草人：。

道路车辆 统一诊断服务(UDS)

1 范围

本部分规定了道路车辆统一诊断服务(UDS)的基本要求、会话层服务、启用控制区区域网络(CAN)的统一诊断设备、FlexRay 实施的统一诊断服务(UDSSonFR)、因特网协议实施的统一诊断服务(UDSSonIP)、K-Line 实施的统一诊断服务(UDSSonK-Line)、局域互连网络(UDSSonLIN)上的 UDS。

本标准适用于道路车辆统一诊断服务。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

ISO 10681-2 道路车辆 FlexRay通信 第2部分：通信层服务(Road vehicles—Communication on FlexRay—Part2:Communication layer services)

ISO 13400（所有部分）道路车辆 互联网协议的诊断通信（DoIP）（Road vehicles—Diagnostic communication over Internet Protocol (DoIP)）

ISO 14230-1 道路车辆 K-Line(DoK-Line)诊断系统 第1部分:物理层（Road vehicles—Diagnostic communication over K-Line(DoK-Line)—Part1:Physical layer）

ISO 14230-2 道路车辆 K-Line(DoK-Line)诊断系统 第2部分:数据链路层（Road vehicles—Diagnostic communication over K-Line(DoK-Line)—Part2>Data link layer）

ISO 14230-4 道路车辆 K-Line(DoK-Line)诊断系统 第4部分:排放相关系统的需求（Road vehicles—Diagnostic systems—Keyword Protocol 2000—Part4:Requirements for emission-related systems）

ISO 15765-1 道路车辆 基于控制器局域网的诊断通信 第1部分：综述（Road vehicles—Diagnostic communication over Controller Area Network(DoCAN)—Part1:General information and use case definition）

ISO 15765-2 道路车辆 基于控制器局域网的诊断通信 第2部分：传输层协议和网络层服务（Road vehicles—Diagnostic communication over Controller Area Network (DoCAN) —Part2:Transport protocol and network layer services）

ISO 15765-4 道路车辆 基于控制器局域网的诊断通信 第3部分：排放相关系统的需求（Road vehicles—Diagnostic communication over Controller Area Network(DoCAN)—Part4:Requirements for emissions-related systems）

ISO 17458-2 道路车辆 FlexRay通信系统 第2部分：数据链路层规范(Road vehicles—FlexRay communications system—Part 2>Data link layer specification)

ISO 17458-4 道路车辆 FlexRay通信系统 第4部分：电气物理层规范(Road vehicles—FlexRay communications system—Part 4: Electrical physical layer specification)

ISO 17987-3 道路车辆 本地互连网络(LIN) 第3部分：协议规范(Road vehicles—Local Interconnect Network (LIN) —Part3:Protocol specification)

3 术语、定义和缩略语

3.1 术语和定义

ISO/IEC 7498-1、ISO 14230-1、ISO 14230-2、ISO 10681-2、ISO 15765-1、ISO 15765-2以及ISO 15765-4、ISO 17987（所有部分）中界定的及下列术语和定义适用于本文件。

3.1.1

网关 gateway

用于在OSI各层级传输PDU的网络设备。

示例：允许使用不同通信协议和不同通信速率的控制模块网络相互进行通信的网络设备，包括但不限于网桥、交换机、路由器或应用层路由等网关装置。

3.1.2

路由器 router

在OSI第3层和第4层传输PDU的网络设备。

3.1.3

交换机 switch

在OSI第2层传输PDU的网络设备。

引导管理器 boot manager

引导软件的一部分，在ECU电源接通或复位后立即执行，其主要目的是检查执行重编程软件的有效应用程序是否可控。

注：引导管理器还可以考虑将控制转换到重新编程软件的其他条件。

3.1.4

引导内存分区 boot memory partition

引导软件所在的服务端内存的区域。

3.1.5

引导软件 boot software

在服务端内存的特殊部分中执行的软件，主要用于启动ECU并执行服务端刷新。

注1：刷新过程中，这个内存区域不会被擦除，并且当服务端应用程序丢失或被认为无效时必须执行，以确保其重编程服务端。

注2：参照术语3.1和3.17。

3.1.6

客户端 client

诊断仪的一个使用诊断服务的功能。（Tester）

注：诊断仪通常使用其他功能，如数据库管理，数据解析，人机接口。

3.1.7

诊断数据 diagnostic data

电子控制单元的存储器中的数据，且该数据可能被诊断仪检查和/或可能被修改。

注1：诊断数据包括模拟输入输出，数字输入输出，中间值和各种状态信息。

注2：诊断数据的示例如车速，节气门开度，外后视镜位置，系统状态等。诊断数据定义了三种类型的值：

——当前值：电子控制单元正常运行当前使用的值（或由此产生）；

——存储值：在特定时刻（例如当发生故障或周期性地）产生的当前值的内部备份；该备份在电子控制单元的控制下进行；

——静态值：如VIN。

服务端没有义务基于诊断的目的保留其数据的内部备份，在这种情况下，诊断仪只能请求当前值。

注 3：基于售后或开发的应用场景会选择不同的服务端功能（例如，只在开发的应用场景下，允许访问所有存储器位置）。

3.1.8

诊断例程 diagnostic routine

嵌入在电子控制单元中并且可以根据客户端的请求由服务端启动的例程。

注：它可以运行非正常的操作程序，或者可以在此模式下启用并使用正常的操作程序执行。在第一种情况下，不适用于服务端的正常操作。在第二种情况下，当电子控制单元的所有其它部分正常工作时，可以运行多个诊断程序。

3.1.9

诊断服务 diagnostic service

客户端发起的信息交换，以便请求服务端的诊断信息或/和基于诊断目的修改 ECU 行为。

3.1.10

诊断会话 diagnostic session

服务端启用特定的诊断服务和功能集的内部状态。

3.1.11

诊断故障码 diagnostic trouble code

DTC

由车载诊断系统定义的用于标识故障状态的数字通用标识符。

3.1.12

电子控制单元 ECU

至少包括一个服务端的电子控制单元。

注：被认为是电子控制单元的系统包括防抱死制动系统（ABS）和发动机管理系统。

3.1.13

功能单元 functional unit

功能密切相关或互补的诊断服务集。

3.1.14

整型 integer type

具有不同值的简单类型，包括正整数、负整数和零。

3.1.15

本地客户端 local client

与服务端连接到同一本地网络的客户端，并且是与服务端同一地址空间。

3.1.16

本地服务端 local server

与客户端连接到同一本地网络的服务端，并且是与客户端同一地址空间。

3. 1. 17

开放系统互联 open systems interconnection

OSI

3. 1. 18

永久 DTC permanent DTC

即使在清除 DTC 请求之后，仍然保持在非易失性存储器中的诊断故障码（DTC），直到满足其他**要求**才能进行清除（例如，每个 DTC 的适合的监视器已成功通过）。

3. 1. 19

记录 record

一个或多个通过唯一标识方式引用的诊断数据元素。

注：记录的示例是包括各种输入/输出数据和故障代码的快照。

3. 1. 20

远程服务端 remote server

不直接连接到诊断主网络的服务端。

注 1：通过远程地址识别远程服务端。远程地址表示其地址独立于主网络的地址空间。

注 2：通过主网络上的本地服务端访问远程服务端。主网络上的每个本地服务端都可以作为一组独立的远程服务端的网关。因此，必须有一对地址始终用于识别远程服务端：一个本地地址，用于标识远程网络的网关，以及一个标识远程服务端本身的远程地址。

3. 1. 21

远程客户端 remote client

不直接连接到诊断主网络的客户端。

注 1：通过远程地址识别远程客户端。

注 2：远程地址表示其地址独立于主网络的地址空间。

3. 1. 22

刷新软件 reprogramming software

引导软件的一部分，用于电子控制单元的刷新。

3. 1. 23

安全 security

保护车辆模块不受来自车辆诊断数据链路“未授权”入侵的机制。

3. 1. 24

服务端 server

电子控制单元提供诊断服务的部分功能。

3. 1. 25

支持的诊断故障码（DTC） supported DTC

当前被配置/标定并且能够在预定义的车辆条件下执行的诊断故障代码。,

3. 1. 26

诊断设备 tester

控制诸如车载电子控制单元的测试，检查，监控或诊断等功能的系统，可专用于特定类型的操作员（例如专用于车库机械的非车载扫描工具，专用于装配厂的非车载测试工具或车载诊断仪）

注：诊断设备也被称为客户端。

3. 2 缩略语

下列缩略语适用于本文件。

.con: 服务原语.确认 (service primitive.confirmation)

.ind: 服务原语.指示 (service primitive.indication)

.req: 服务原语.请求 (service primitive.request)

A_PCI: 应用层协议控制信息 (application layer protocol control information)

AE: 扩展地址 (address extension)

CF: 连续帧 (consecutive frame)

CDD: 通信数据词典 (common data dictionary)

CMD: 通信消息词典 (common message dictionary)

DSC: 诊断会话控制 (diagnostic session control)

DA: 目标地址 (destination address)

DLC: 数据长度码 (data length code)

DoFR: 基于 FlexRay 的诊断通信 (diagnostic communication over FlexRay)

DID: 数据标识符 (data identifier)

DoIP: 互联网协议的诊断通信 (diagnostic communication over Internet Protocol)

DoIP_AI: DoIP 地址信息 (DoIP address information)

ECU: 电子控制单元 (electronic control unit)

EDR: 事件数据记录仪 (event data recorder)

FF: 首帧 (first frame)

FB: 首字节 (first byte)

FC: 流控帧 (flow control)

FF: 第一帧 (first frame)

FR: 总线 (FlexRay)

ID: 标识符 (identifier)

IP: 互联网协议 (Internet Protocol)

ISO: 国际标准化组织 (International Standardization Organization)

LF: 结束帧 (last frame)

Mtype: 报文类型 (message type)

NA: 网络地址 (network address)

NAD: 节点地址 (node address)

NCF: 节点配置文件 (node configuration file)

N/A: 不适用(not applicable)

NR_SI: 否定响应服务标识符 (negative response service identifier)

NRC: 否定响应代码 (negative response code)

OSI: 开放系统互联 (open systems interconnection)

P2: 服务器响应时间 (server response time)
pDID: 周期数据标识符 (periodic data identifier)
Rx: 接收 (reception)
SA: 源地址 (source address)
SF: 单帧 (single frame)
SFID: 子功能标识符 (subfunction identifier)
SID: 服务标识符 (service identifier)
SM: 子网掩码 (subnet mask)
SOM: 消息起始 (start of message)
STF: 起始帧 (start frame)
S_{Tmin}: 间隔时间 (separation time)
STRT: 待服务端响应 (serviceToRespondTo)
S_AE: 会话层地址扩展 (session layer address extension)
S_SA: 会话层源地址 (session layer source address)
S_Data: 会话层数据传输服务名称 (session layer data transfer service name)
SI: 服务标识符 (service identifier)
SOM: 消息起始 (start of message)
S_Mtype: 会话层消息类型 (session layer message type)
S_PDU: 会话层协议数据单元 (session layer protocol data unit)
S_TA: 会话层目标地址 (session layer target address)
S_TAtype: 会话层目标地址类型 (session layer target address type)
TA: 目标地址 (target address)
Tx: 发送 (transmission)
UDS: 统一诊断服务 (unified diagnostic services)
UDSonFR: 基于 FlexRay 的统一诊断服务 (Unified Diagnostic Services(UDS)on FlexRay)
UART: 通用异步收发两用机 (universal asynchronous receiver transmitter)
USDT: 未确认分段数据传输 (unacknowledged segmented data transfer)
UUDT: 未确认不分段数据传输 (unacknowledged unsegmented data transfer)
VM: 车辆制造商 (vehicle manufacturer)

4 约定

本标准遵循适用于诊断服务的OSI服务公约 (ISO/IEC 10731) 中的约定。

5 规范和要求

5.1 一般规定

本标准是基于OSI服务约定 (ISO / IEC 10731: 1994) 中的惯例，因为它们适用于诊断服务。

这些约定规定了服务用户与服务提供者之间的交互。信息通过服务原语在服务用户和服务提供者之间传递参数。

服务和协议之间的区别如图1所示。

本标准定义了确认和未确认的服务。

确认的服务使用六个服务原语请求，请求确认，指示，响应，响应确认和确认。

未确认的服务仅使用请求，请求确认和指示服务原语。

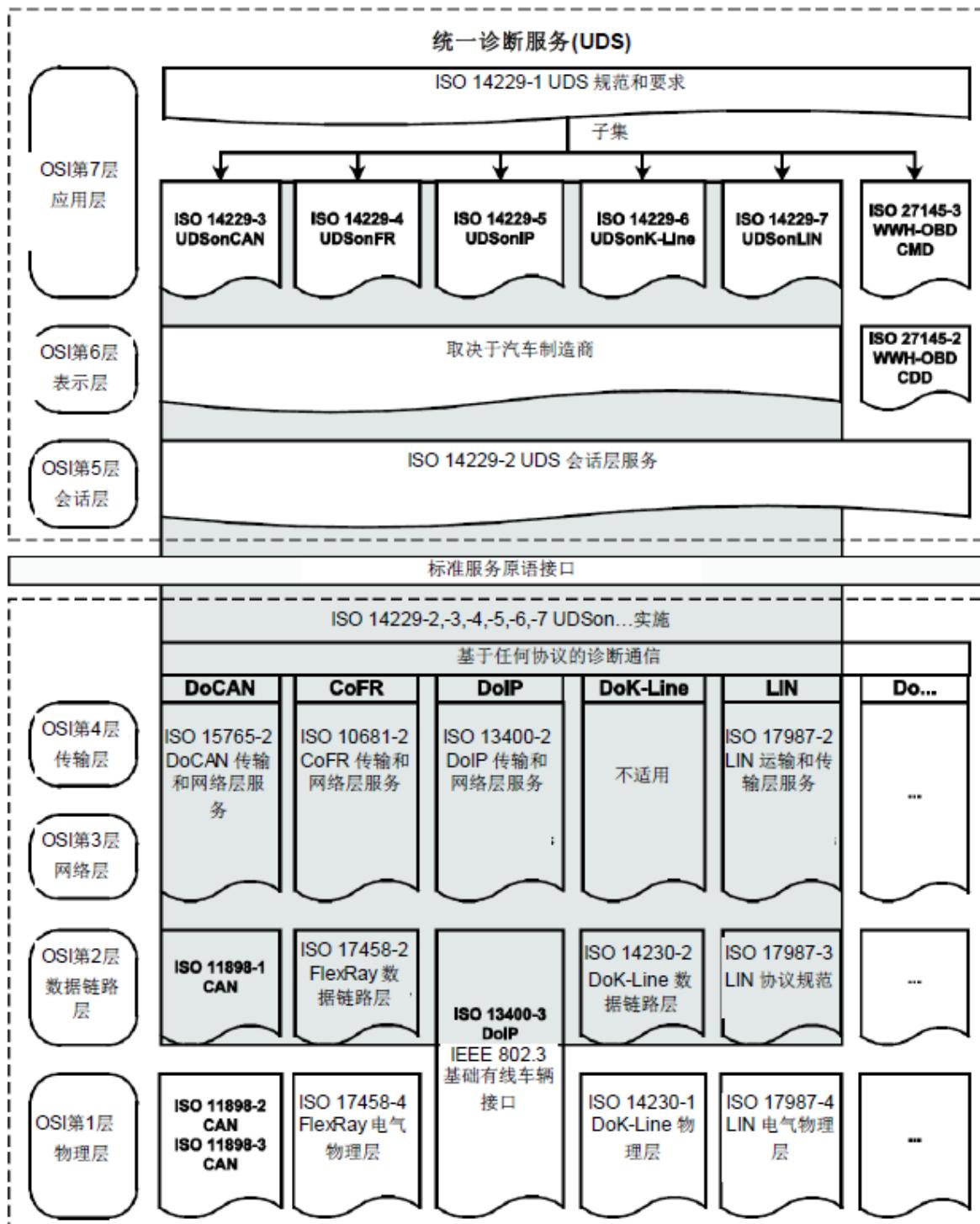


图1 依据 OSI 模型的 UDS 参考文件实施

本标准定义的所有服务，请求和指示服务原语都具有相同的格式和参数。因此，对于所有服务，响应和确认服务原语（除了请求确认和响应确认）总是具有相同的格式和参数。

本标准在定义服务原语时，仅列出请求和响应服务原语。

5.2 应用层服务

5.2.1 概述

应用层服务通常被称为诊断服务。应用层服务用于基于客户端 - 服务端的系统，以执行诸如车载车辆服务端的测试，检查，监控或诊断等功能。客户端（通常称为诊断设备）使用应用层服务来请求在一个或多个服务端中执行诊断功能。服务端（通常作为 ECU 的部分功能）使用应用层服务将由请求的诊断服务提供的响应数据发送回客户端。客户端通常是非车载诊断仪，但在某些系统中也可以是车载诊断仪。应用层服务的使用独立于客户端，无论该客户端是车载还是非车载诊断仪。同一车辆系统中允许有多个客户端。

诊断应用层的服务接口提供了大量具有相同通用结构的服务。对于每个服务，规定了六项服务原语：由诊断仪应用中的客户端功能使用的服务请求原语，将所请求的诊断服务的数据传递到诊断应用层：

——服务请求原语：在诊断仪应用中由客户端使用，将所请求的诊断服务的数据传递到诊断应用层；

——服务请求-确认原语：在诊断仪应用中由客户端使用，指示在服务请求原语中传递的数据在诊断仪所连接得总线上成功发送；

——服务指示原语：由诊断应用层使用，将数据发送到 ECU 诊断应用的服务端；

——服务响应原语：由 ECU 诊断应用的服务端使用，将由所请求的诊断服务提供的响应数据发送到诊断应用层；

——服务响应-确认原语：由 ECU 诊断应用中的服务端使用，指示在 ECU 接收到诊断请求的总线上成功发送服务响应原语中传递的数据；

——服务确认原语：由诊断应用层使用，将数据发送给诊断仪应用中的客户端。

图 3 中是应用层服务原语-确认服务。

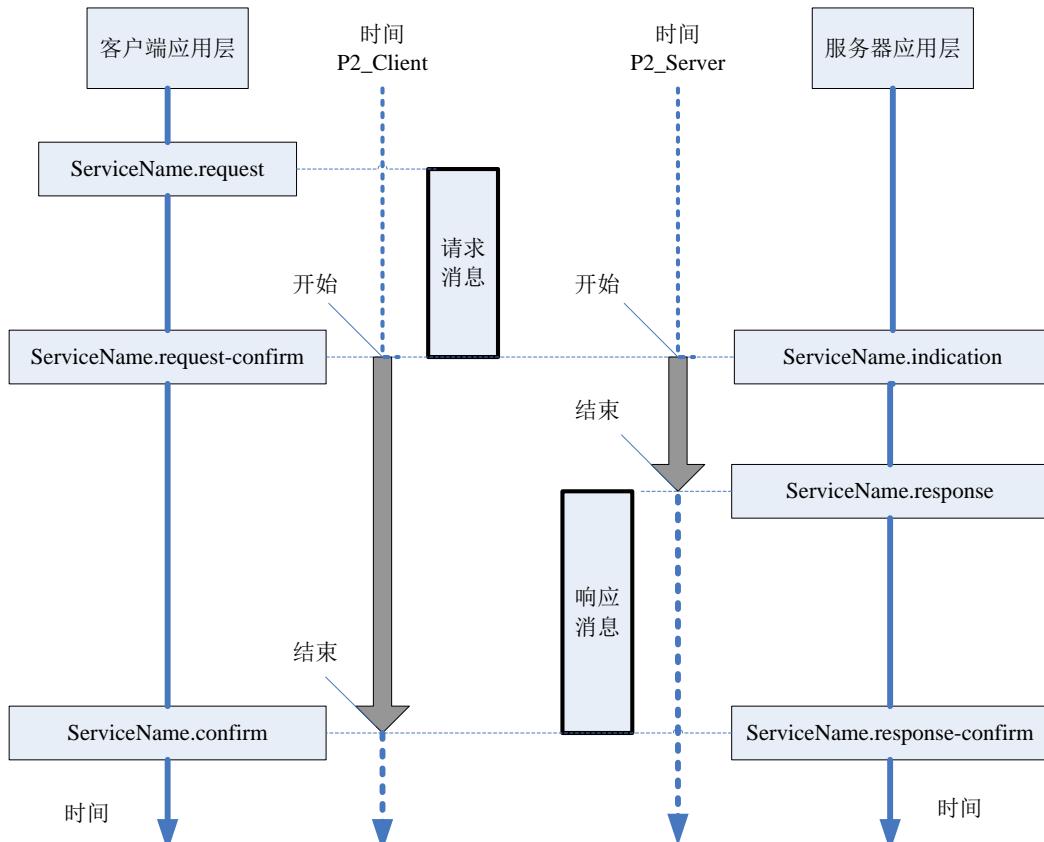


图 3 应用层服务原语-确认服务

图 4 描述了应用层服务原语-非确认服务

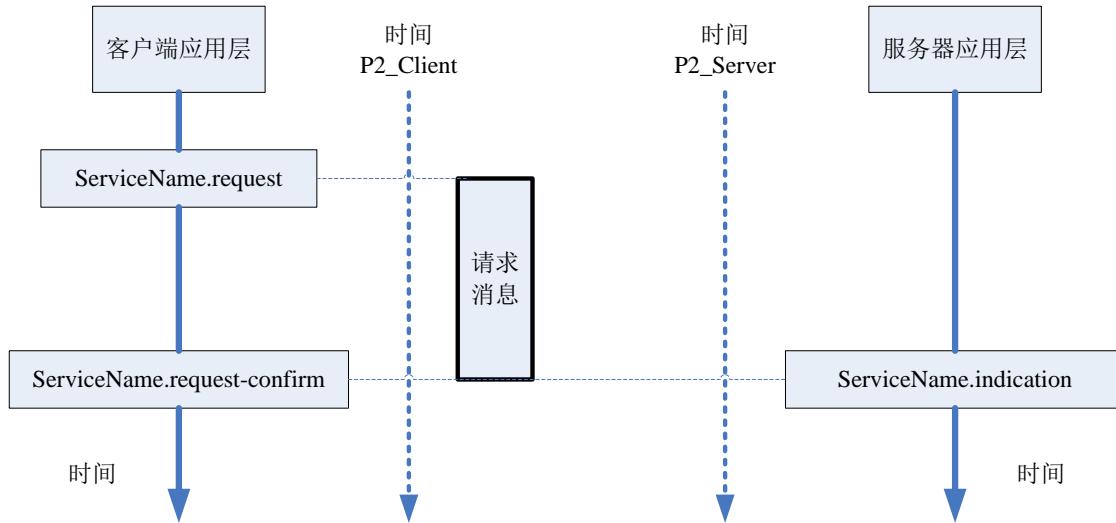


图 4 应用层服务原语-非确认服务

对于给定的服务，请求-确认原语和响应-确认原语通常具有相同的服务数据单元。这些服务原语的目的是指示先前的请求或响应服务原语调用已经完成。本标准中描述的服务将不会使用这些服务原语，但是数据链路特定的实施文档可能会使用这些描述来定义诸如**服务执行参考点**。(当响应完全传送到客户端时，ECU 复位服务将调用复位，客户端通过服务端响应确认原语在服务端中指示)

5.2.2 应用层服务格式描述

应用层服务可以具有两种不同的格式，具体取决于车辆诊断系统的配置方式。应用层服务的格式由参数 **A_Mtype** 控制。

如果把车辆系统配置到客户端可以使用 **A_SA** 和 **A_TA** 地址参数对所有服务端进行寻址，则应使用应用层服务的默认格式。即 **A_Mtype = diagnostics**。

如果把车辆系统配置到客户端需要使用除了 **A_SA** 和 **A_TA** 地址参数之外的地址信息对特定服务端进行寻址，则应使用应用层服务的远程格式。即 **A_Mtype = remote diagnostics**。

应用层服务的不同格式在 6.3 中规定。

5.2.3 服务原语的格式描述

5.2.3.1 通用定义

所有应用层服务都具有相同的通用格式。服务原语是按如下格式进行编写：

```
service_name.type (
    parameter A, parameter B, parameter C
    [, parameter 1, ...]
)
```

注：

- “service_name” 是诊断服务的名称（例如：诊断会话控制），
- “type” 指示服务原语的类型（例如：请求），
- “parameter A, ...” 是 A_SDU（应用层服务数据单元）作为服务原语传递的值的列表（寻址信息），
- “parameter A, parameter B, parameter C” 是必须包含在所有服务调用中的强制参数。
- “[, parameter 1, ...]” 是取决于特定服务的参数（例如：parameter 1 可以是诊断会话控制服务的诊断会话）。[] 表示参数列表的这一部分可能为空。

5.2.3.2 服务请求和服务指示原语

对于每个应用层服务，根据以下通用格式指定服务请求和服务指示原语：

```
service_name.request (
    A_MType,
    A_SA,
    A_TA,
    A_TA_type,
    [A_AE],
    A_Length,
    A_Data[, parameter 1, ...],
```

)

诊断设备应用程序中的客户端功能使用请求原语启动服务并将所请求的诊断服务的数据传递给应用层。

```
service_name.indication (
    A_MType,
    A_SA,
    A_TA,
    A_TA_type,
    [A_AE],
    A_Length,
    A_Data[, parameter 1, ...],
)
```

应用层使用指示原语来指示对 ECU 诊断应用程序有重要意义的内部事件，并将关于所请求诊断服务的数据传递给 ECU 诊断应用程序的服务端功能。这意味着当数据从客户端发送到服务端时，各个参数的值不应由应用层的对等通信协议实体改变。由客户端应用程序中的客户端功能传递到服务请求呼叫的应用层的相同值将由对等应用层的服务指示由诊断应用程序的服务端功能接收。

5.2.3.3 服务响应和服务确认原语

对于每个应用层服务，根据以下通用格式指定服务响应和服务确认原语：

```
service_name.response (
    A_MType,
    A_SA,
    A_TA,
    A_TA_type,
    [A_AE],
    A_Length,
    A_Data[, parameter 1, ...],
)
```

ECU 诊断应用程序中的服务端功能使用响应原语来启动服务并将所请求的诊断服务提供的响应数据发送到应用层。

```
service_name.confirm (
    A_MType,
    A_SA,
    A_TA,
    A_TA_type,
    [A_AE],
    A_Length,
    A_Data[, parameter 1, ...],
)
```

应用层使用确认原语来指示对客户端应用有重要意义的内部事件，并将关联的之前的服务所请求的结果传递给诊断设备应用程序中的客户端功能。它不一定指示远程对等接口的任意活动，例如，如果服务端不支持所请求的服务或者通信被破坏。

特定应用层的响应和确认原语通常具有相同的参数和参数值。这意味着当数据从服务端发送到客户端时，各个参数的值不应被应用层的对等通信协议实体改变。由 ECU 诊断应用程序的服务端功能传递到服务响应呼叫中的应用层的相同值将由客户端功能在诊断测试应用程序中从对等应用层的服务确认接收。

对于每个响应和确认原语将规定两种不同的服务数据单元（两组参数）。

——如果请求的诊断服务可以通过 ECU 中的服务端功能成功执行，则应与第一个服务数据单元一起使用肯定响应和肯定确认原语。

——如果请求的诊断服务失败或不能通过 ECU 中的服务端功能及时完成，则应将第二服务数据单元用于否定响应和否定确认原语。

5.2.3.4 服务请求-确认和服务响应-确认原语

对于每个应用层服务，根据以下通用格式规定服务请求-响应和服务响应-确认原语：

```
service_name.req_confirm (
    A_MType,
    A_SA,
```

```
A_TA,  
A_TA_type,  
[A_AE],  
A_Result  
)
```

应用层使用请求-确认原语来指示对客户端应用程序有重要意义的内部事件，并将关联的之前的服务所请求的结果传递给诊断设备应用程序中的客户端功能。

```
service_name.rsp_confirm (   
    A_MType,  
    A_SA,  
    A_TA,  
    A_TA_type,  
    [A_AE],  
    A_Result  
)
```

应用层使用响应-确认原语来指示对服务端应用程序有重要意义的内部事件，并将关联的之前的服务所响应的结果传递给诊断设备应用程序中的服务端功能。

5.2.4 服务数据单元定义

5.2.4.1 强制参数

5.2.4.1.1 通用定义

应用层服务包含三个强制参数。以下参数定义适用于本标准中规定的应用层服务（标准和远程格式）。

5.2.4.1.2 A_Mtype, 应用层消息类型

类型：枚举

范围：诊断，远程诊断

描述：

参数 Mtype 应用于识别 6.2 中规定的车辆诊断系统的格式。本标准为此参数指定了两个值的范围：

如果 A_Mtype = 诊断，service_name 原语应由参数 A_SA, A_TA 和 A_TAtype 组成。

如果 A_Mtype = 远程诊断，service_name 原语应由参数 A_SA, A_TA、A_TAtype 和 A_AE 组成。

5.2.4.1.3 A_SA, 应用层源地址

类型：2 字节无符号整型数值

范围：0x0000 – 0xFFFF

描述：

参数 SA 应用于对标识客户端和服务端地址的标识符进行编码。

对于服务请求（和服务指示），A_SA 表示已请求诊断服务的客户端地址。请求诊断服务的每个客户端功能都应用一个 A_SA 值表示。如果在同一个诊断设备实现多个客户端，那么每个客户端应具有自己的客户端标识符和相应的 A_SA 值。

对于服务响应（和服务确认），A_SA 表示已执行请求的诊断服务的服务端的地址。服务端可以在一个 ECU 中实现，也可以在多个 ECU 中分布式实现。如果服务端仅在一个 ECU 中实现，则只能使用一个 A_SA 值进行编码。如果服务端在多个 ECU 中分布式实现，则相应的服务端功能地址应为每个服务端编码一个 A_SA 值。

如果消息来源于远程客户端或服务端，则 A_SA 代表本地服务端，该服务端为远程网络到主网络的网关。

注：如果使用物理寻址请求消息，响应消息中的 A_SA 值将与相应请求消息中的 A_TA 值相同。

5.2.4.1.4 A_TA, 应用层目标地址

类型：2 字节无符号整型数值

范围：0x0000 – 0xFFFF

描述：

参数 A_TA 应用于对标识客户端和服务端地址的标识符进行编码。

有两种寻址方式：

——物理寻址；

——功能寻址。

因此，可以为车辆系统定义两个独立的目标地址集（每个寻址方式一个）。

物理寻址始终是在一个 ECU 中实现的服务端的专用消息。当使用物理寻址时，通信是客户端和服务端之间

的点对点通信。

如果客户端不知道要响应诊断服务请求的服务端的物理地址，或者服务端是否在多个 ECU 中分布实现，则客户端将使用功能寻址。当使用功能寻址时，通信是从客户端到在一个或多个 ECU 中实现的服务端的广播通信。

对于服务请求（和服务指示），**A_TA** 表示将执行请求的诊断服务的服务端的标识符。如果正在寻址远程服务端，则 **A_TA** 代表本地服务端，该服务端为远程网络到主网络的网关。

对于服务响应（和服务确认），**A_TA** 表示最初请求诊断服务并应接收请求的数据（即请求的 **A_SA**）的客户端地址。服务响应（和服务确认）应始终使用物理寻址。如果远程客户端被寻址，则 **A_TA** 代表本地服务端，该服务端为远程网络到主网络的网关。

注：响应消息的 **A_TA** 值始终与相应请求消息的 **A_SA** 值相同。

5.2.4.1.5 **A_TA_Type**, 应用层目标地址类型

类型：枚举

范围：物理，功能

描述：

参数 **A_TA_type** 是 **A_TA** 参数的扩展。它用于表示传输消息的寻址方式。

5.2.4.1.6 **A_Result**

类型：枚举

范围：ok, error

描述：

req_confirm 和 **rsp_confirm** 原语使用参数“**A_Result**”来指示消息是否已正确发送（ok）或消息传输是否不成功（错误）。

5.2.4.1.7 **A_Length**

类型：4 字节无符号整型数值

范围：0 – (2³²-1)

描述：

该参数包括发送/接收的数据的长度。

5.2.4.1.8 **A_Data**

类型：字节流

范围：N/A

描述：

该参数包括要由较高层实体交换的所有数据。

5.2.4.2 车辆系统要求

车辆制造商应确保系统中的每个服务端都具有唯一的标识服务端地址的标识符。车辆制造商应确保系统中的每个客户端都具有唯一的标识客户端地址的标识符。

车辆系统诊断网络的所有客户端和服务端地址都应编码在相同的源地址范围内。这表示客户端和服务端不能在同一给定的车辆系统中用相同的 **A_SA** 值表示。

服务端的物理目标地址应始终与服务端的源地址相同。

远程服务端标识符的分配可以独立于主网络上的标识客户端和服务端地址的标识符。

一般来说，只有所寻址的服务端应响应客户端请求的消息。

5.2.4.3 可选参数 - **A_AE**, 应用层远程地址

类型：2 字节无符号整型数值

范围：0x0000 – 0xFFFF

描述：

A_AE 用于扩展可用的地址范围以对客户端和服务端标识符进行编码。**A_AE** 只能用于区分本地服务端和远程服务端概念的车辆。远程地址表示其自身的地址范围，与主网络上的地址无关。

参数 **A_AE** 应用于编码远程客户端和服务端标识符。**A_AE** 可以根据具有 **A_AE** 的消息的方向表示远程目标地址或远程源地址。

对于由主网络上的客户端发送的服务请求（和服务指示），**A_AE** 表示将执行请求的诊断服务的远程服务端标识符（远程目标地址）。

A_AE 可以用作物理地址和功能地址。对于 **A_AE** 的每个值，系统构建器应指定该值表示的是物理地址还是功能地址。

注：没有特殊参数用以表示物理或功能的远程地址，以 A_TA_type 指定 A_TA 的寻址方法的方式。物理和功能远程地址共享相同的 1 字节范围的值，每个值的含义应由系统架构师定义。

对于由远程服务端发送的服务响应（和服务确认），A_AE 表示已执行请求的诊断服务的远程服务端的物理位置（远程源地址）。

远程服务端可以仅在一个 ECU 中实现，或者在多个 ECU 中分布实现。如果远程服务端仅在一个 ECU 中实现，则只能使用一个 A_AE 值进行编码。如果远程服务端在多个 ECU 中分布实现，则应使用一个 A_AE 值对远程客户端标识符进行编码，且远程服务端的每个物理地址都有一个 A_AE 值。

5.3 应用层协议

5.3.1 一般定义

应用层协议应始终是确认的消息传输，这意味着对于从客户端发送的每个服务请求，服务端应发送一个或多个相应的响应。

该规则的唯一例外是在使用功能寻址时的一些情况，或者请求/指示规定不应产生响应/确认。为了不给系统带来许多不必要的消息，在几个情况下，即使服务端无法完成请求的诊断服务，也不会发送否定的响应消息。这些例外情况在本标准的相关小节中进行了描述（例如，见 7.5）。

应用层协议应与会话层协议并行处理。这意味着即使客户端正在等待之前发出的请求的响应，也应符合适当的会话层定时要求（例如，根据需要发送“诊断设备在线服务”请求以保持其他服务端的诊断会话，其实现取决于所使用的数据链路层）。

5.3.2 协议数据单元规定

A_PDU（应用层协议数据单元）由 A_SDU（应用层服务数据单元）和层特定控制信息 A_PCI（应用层协议控制信息）直接构成。A_PDU 应具有以下通用格式：

```
A_PDU (
    Mtype,
    SA,
    TA,
    TA_type,
    [RA, ]
    A_Data = A_PCI + [parameter 1, ...],
    Length
)
```

其中：

- “Mtype, SA, TA, TA_type, RA, Length” 与 A_SDU 中使用的参数相同；
- “A_Data” 是为每个单独的应用层服务定义的一串字节数据。A_Data 字符串应以 A_PCI 开头，后面是每个服务指定的 A_SDU 的所有服务特定参数。括号表示参数列表的这部分可能为空；
- “Length” 决定 A_Data 的字节数。

5.3.3 应用层协议控制信息

5.3.3.1 PCI, 协议控制信息

A_PCI 有两种格式。A_PCI 参数的第一个字节的值代表格式。

对于所有第一个字节不等于 0x7F 的服务请求和服务响应，应使用以下定义：

```
A_PCI (
    SI)
```

其中：

- “SI” 是参数服务标识符。

对于第一个字节值等于 0x7F 的服务响应，应使用以下定义：

```
A_PCI (
    NR_SI,
    SI
)
```

其中：

- “NR_SI” 是识别否定服务响应/确认的特殊参数；
- “SI” 是服务标识符的参数。

注：对于服务通过周期性标识符读取数据(0x2A, 见 10.5)中定义的周期性数据响应消息的传输，应用层协议数据单元(A_PDU)中不存在 A_PCI。

5.3.3.2 SI, 服务标识符

类型: 1 字节无符号整型数值

范围: 0x00 - 0xFF 根据表 2 的定义。

表 2 服务标识符定义

服务标识符	服务类型 (第6位)	定义出处
0x10 - 0x3E	本标准定义的服务请求	本标准
0x3F	不适用	预留
0x50 - 0x7E	本标准定义的肯定响应	本标准
0x7F	否定响应服务标识符	本标准
0x80 - 0x82	不适用	本标准预留
0x83 - 0x88	本标准定义的服务请求	本标准
0x89 - 0xB9	不适用	本标准预留
0xBA - 0xBE	服务请求	系统供应商定义
0xBF - 0xC2	不适用	本标准预留
0xC3 - 0xC8	本标准定义的肯定响应	本标准
0xC9 - 0xF9	不适用	本标准预留
0xFA - 0xFE	肯定响应	系统供应商定义
0xFF	不适用	预留

注: 请求消息的服务标识符和肯定响应消息的服务标识符之间存在一一对应的关系, SI 字节值的第 7 位 (bit6) 表示服务类型。所有请求消息的 SI 第 7 位 (bit 6) = 0。除了通过“周期性标识符读取数据”(0x2A, 参见 10.5) 服务的定期数据响应消息之外, 所有肯定响应消息的 SI 第 7 位 (bit6) = 1。

描述:

SI 将用于对服务原语中已被调用的特定服务进行编码。每个请求服务应分配一个唯一的 SI 值。每个肯定响应服务也应分配相应的唯一 SI 值。

服务标识符用于表示从应用层传递到较低层 (并从较低层返回) 的 A_Data 数据字符串中的服务。

5.3.3.3 NR_SI, 否定响应服务标识符

类型: 1 字节无符号整型值

固定值: 0x7F

描述:

参数 NR_SI 是识别否定服务响应/确认的特殊参数。应是 A_PCI 的一部分, 用于否定响应/确认消息。

注: NR_SI 值与 SI 值协调。为了使 A_Data 编码和解码更容易, NR_SI 值不同于 SI 值。

5.3.4 否定响应/确认服务原语

每个诊断服务都具有否定响应/否定确认消息, 根据表 3 中的 A_Data 字节值定义。A_Data 第一个字节 (A_PCI.NR_SI) 通常是特定的否定响应服务标识符。A_Data 第二个字节 (A_PCI.SI) 应为否定响应消息对应的服务请求/指示消息的服务标识符值的副本。

表 3 A_PDU 否定响应

A_PDU 参数	参数名称	约定值	字节值	助记符
SA	源地址	M	0xXXXX	SA
TA	目标地址	M	0xXXXX	TA
TAtype	目标地址类型	M	0xXX	TAT
RA	远程地址	C	0xXXXX	RA
A_Data.A_PCI.NR_SI	否定响应 SID	M	0x7F	SIDNR
A_Data.A_PCI.SI	<服务名称>请求 SID	M	0xXX	SIDRQ
A_Data.Parameter 1	响应码	M	0xXX	NRC_
M(强制): 在发出否定响应 A_PDU 的情况下, 这些 A_PDU 参数应该存在。				
C (有条件的): RA (远程地址) PDU 参数仅在远程寻址的情况下才存在。				

注: A_Data 代表否定响应消息的消息数据字节。

在否定响应消息中使用参数 responseCode 来指示诊断服务失败或无法及时完成的原因。这些值在附录 A.1

中定义。

5.3.5 服务端响应实施规则

5.3.5.1 一般定义

以下小节规定了执行服务时的服务端的行为。服务端和客户端应遵循这些实施规则。

5.3.5.2 一般的服务端响应行为

本节中规定的一般的服务端响应行为对所有请求消息都是必需的。验证步骤从接收请求消息开始。下图展示了三种实现要求：

——强制：每个请求消息都要进行检查

——可选：可以由每个请求消息来选择性地进行检查

——制造商/供应商自定义：该步骤可以通过附加的制造商/供应商的具体检查来扩展。

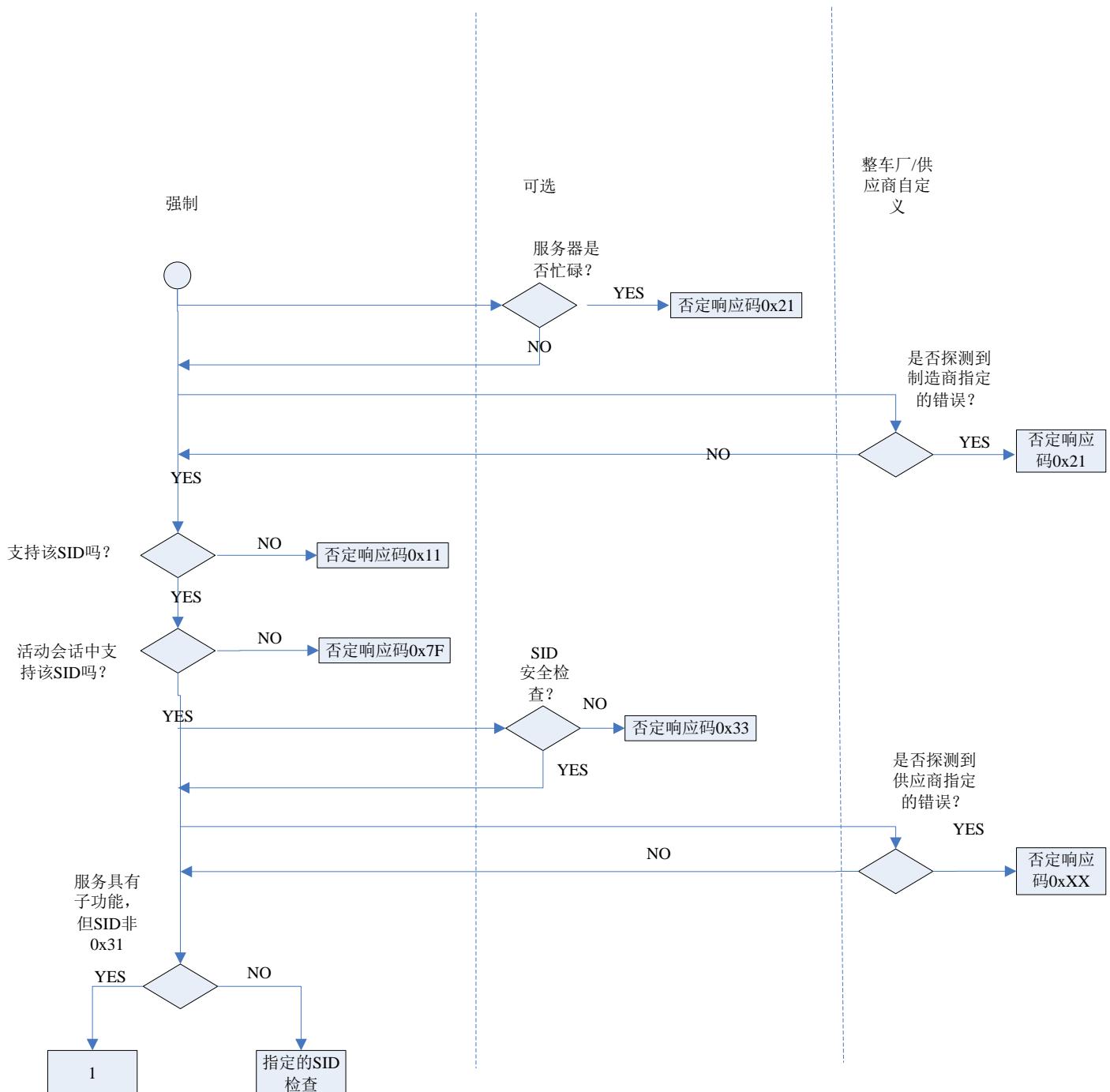


图 5 描述了一般的服务端响应行为。

说明：

1. 因为另一个诊断任务已被不同的客户端请求并正在进行中，所以诊断请求不能被接受。

2. 请参阅每个服务的响应行为（支持的否定响应码）

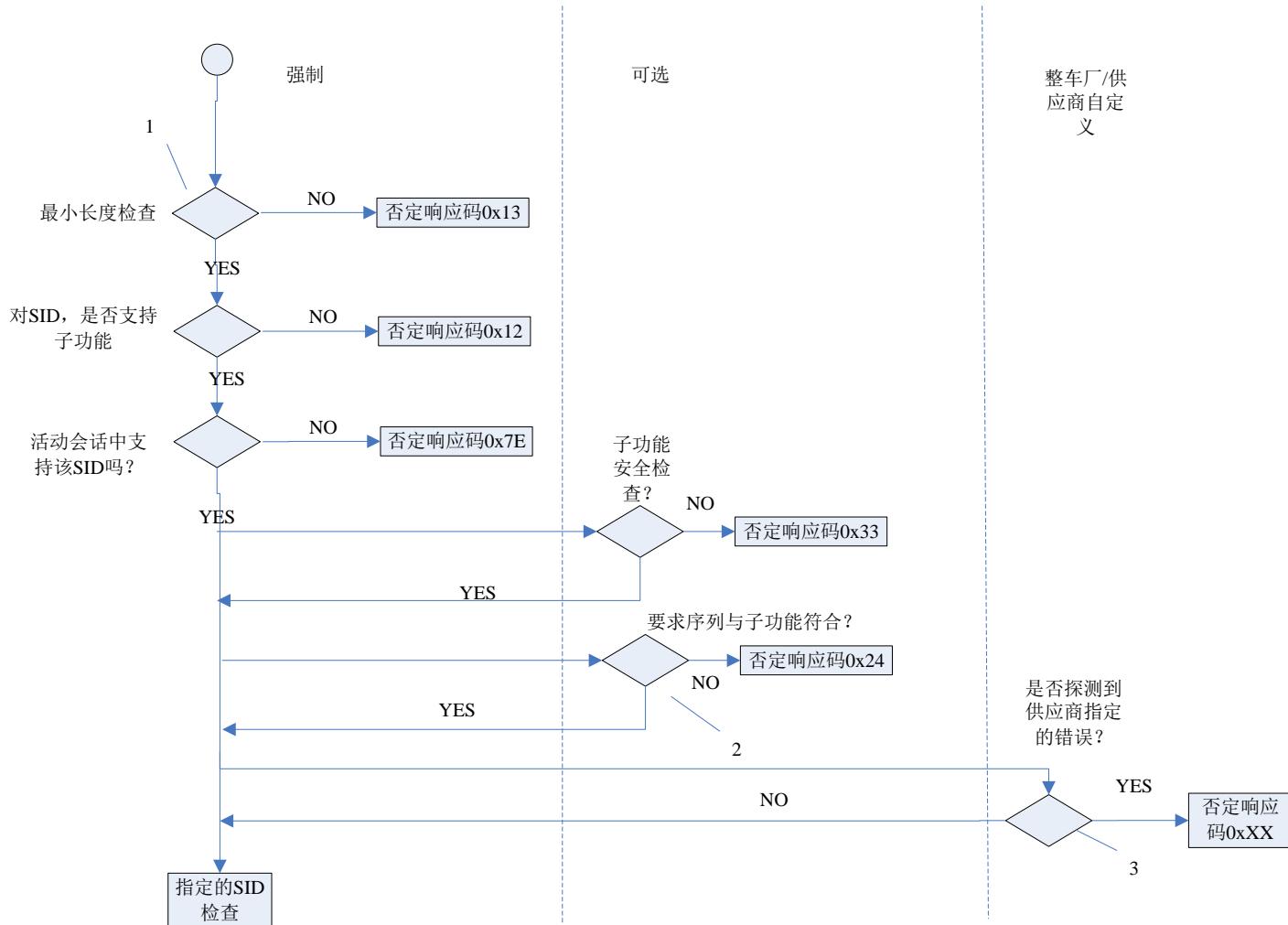
图 5 一般的服务端响应行为

5.3.5.3 具有子功能参数的请求消息和服务端响应行为

5.3.5.3.1 对具有子功能参数的请求消息的一般的服务端响应行为

对于具有子功能参数的所有请求消息，此子条款中指定的一般服务端响应行为是必需的。在本节上下文中的请求消息被定义为遵守本标准定义的格式化要求的服务请求消息。

图 6 描述了对于请求的具有子功能参数的请求消息的一般服务端响应行为。



说明：

1. 至少 2 (SID + 子功能参数)。
2. 如果子功能进行序列检查，例如 链路控制或加密访问。
3. 参考每个服务的响应行为（支持的否定响应代码）。

图 6 对于请求的具有子功能参数的请求消息的一般的服务端响应行为

5.3.5.3.2 物理寻址的客户端请求消息

在对每个服务的描述中引用本条款中指定的服务端响应行为，该服务支持从客户端接收的物理寻址请求消息中的子功能参数。

表 4 显示物理寻址的通信方案。

表 4 具有子功能参数的物理寻址请求消息和服务端响应行为

服务端用例#	客户端请求消息		服务端能力			服务端响应		服务端响应的说明
	寻址方式	子功能(抑制肯定响应指示位)	支持的SI	支持的子功能	支持的数据参数(只有当适用时)	消息	否定响应码	
a)	物理寻址	否(该位=0)	是	是	至少一个	肯定响应	-	服务端发送肯定响应
b)					至少一个	否定响应	0xXX	服务端发送否定响应,因为当读取请求消息的数据参数时发生了错误
c)					无		ROOR	否定响应码为0x31的否定响应
d)			否	-	-		SNS/SNSIAS	否定响应码为0x11或0x7F的否定响应
e)			是	否	-		SNS/SNSIAS	否定响应码为0x12或0x7E的否定响应
f)		是(该位=1)	是	是	至少一个	无响应	-	服务端未发送响应
g)					至少一个	否定响应	0xXX	服务端发送否定响应,因为当读取请求消息的数据参数时发生了错误
h)					无		ROOR	否定响应码为0x31的否定响应
i)			否	-	-		SNS	否定响应码为0x11的否定响应
j)			是	否	-		SNS	否定响应码为0x12的否定响应

5.3.5.3.3 功能寻址客户端请求消息

本条款中指定的服务端响应行为在每个服务的服务描述中被引用,该服务描述支持从客户机接收到的功能寻址请求消息中的子功能参数。

表5显示了具有功能寻址的可能的通信方案。

表5功能寻址的具有子功能参数的请求消息和服务端响应行为

服务端用例#	客户端请求消息		服务端能力			服务端响应		服务端响应的说明
	寻址方式	子功能(抑制肯定响应指示位)	支持的SI	支持的子功能	支持的数据参数(只有当适用时)	消息	否定响应码	
a)	功能寻址	否(该位=0)	是	是	至少一个	肯定响应	-	服务端发送肯定响应
b)					至少一个	否定响应	0xXX	服务端发送否定响应,因为当读取请求消息的数据参数时发生了错误
c)					无	无响应	-	服务端不发送响应
d)			否	-	-		-	服务端不发送响应
e)			是	否	-		-	服务端不发送响应
f)		是(该位=1)	是	是	至少一个	无响应	-	服务端不发送响应
g)					至少一个	否定响应	0xXX	服务端发送否定响应,因为当读取请求消息的数据参数时发生了错误
h)					无	无响应	-	服务端不发送响应
i)			否	-	-		-	服务端不发送响应
j)			是	否	-		-	服务端不发送响应

5.3.5.4 不具子功能参数的请求消息和服务端响应行为

5.3.5.4.1 对不具子功能参数的请求消息和服务端响应行为

对于没有子功能参数的请求消息没有通用的服务端响应行为可用。在本节上下文中的请求消息为符合本标准格式化要求的服务请求消息。

5.3.5.4.2 物理寻址客户端请求消息

在每个服务的描述中引用本条款中指定的服务端响应行为，该服务描述不支持子功能参数，而是支持从客户端接收的物理寻址请求消息中的数据参数。

表 6 显示了物理寻址通信方案。

表 6 物理寻址的不具有子功能参数的请求消息和服务端响应行为

服务端用例#	客户端请求消息	服务端能力			服务端响应		服务端响应的说明
		支持的SI	支持的子功能	支持的数据参数(只有当适用时)	消息	否定响应码	
a)	物理寻址	是	所有	肯定响应	-		服务端发送肯定响应
b)			至少一个		-		服务端发送肯定响应
c)			至少一个	否定响应	0xXX		服务端发送否定响应，因为当读取请求消息的数据参数时发生了错误
d)		-	无		ROOR		否定响应码为 0x31 的否定响应码
e)		否	否		SNS /SNSIAS		否定响应码为 0x11 或 0x7F 的否定响应码

5.3.5.4.3 功能寻址的客户端请求消息

在每个服务的描述中引用本条款中指定的服务端响应行为，该服务描述不支持从客户端接收到的功能寻址请求消息中的子功能参数，而支持数据参数。

表 7 显示了功能寻址通信方案。

表 7 功能寻址的不具有子功能参数的请求消息和服务端响应行为

服务端用例#	客户端请求消息	服务端能力			服务端响应		服务端响应的说明
		支持的SI	支持的子功能	支持的数据参数(只有当适用时)	消息	否定响应码	
a)	功能寻址	是	所有	肯定响应	-		服务端发送肯定响应
b)			至少一个		-		服务端发送肯定响应
c)			至少一个	否定响应	0xXX		服务端发送否定响应，因为当读取请求消息的数据参数时发生了错误
d)		-	无		无响应	-	服务端不发送响应
e)		否	否			-	服务端不发送响应

5.4 服务描述约定

5.4.1 服务描述

本节定义了本标准中每个诊断服务的描述。它定义了每个诊断服务的通用服务描述格式。

本节简要介绍了该服务的功能。每个诊断服务规范以描述客户端和服务端执行的操作开始，每个服务都

是特定的。每个服务的描述都包括一个表，其中列出了其原语的参数：请求/指示，肯定或否定结果的响应/确认。都具有相同的结构。

对于给定的请求/指示和响应/确认 A_PDU 定义，每个参数由以下约定值（Cvt）之一描述。

表 8 定义了 A_PDU 参数的约定定义。

表 8 A_PDU 参数约定定义

类型	名称	描述
M	强制的	A_PDU 中必须包含该参数。
C	有条件的	基于某些标准（例如 A_PDU 内的子功能/参数），A_PDU 中可以包含和使用该参数。
S	可选的	表示该参数是必须的（除非另有说明），并且可以从参数列表中选择。
U	用户自定义的	取决于用户的动态使用情况，该参数可能存在也可能不存在。

注：标记为“M”（必选）的“<服务名称>请求 SID”并不表示服务端必须支持该服务。'M' 仅表示在服务端支持该服务的情况下，请求 A_PDU 中必须包含该参数。

5.4.2 请求消息

5.4.2.1 请求消息定义

本节包括一个或多个表，分别定义了用于服务请求/指示的 A_PDU（应用层协议数据单元，参见 7）参数。如果不同子功能参数（\$ Level）的请求消息在 A_Data 参数的结构上有所不同，并且无法在单个表中清楚地标识，则每个子功能参数（\$ Level）可能会有单独的表格。

表 9 定义了具有子功能的请求 A_PDU 定义。

表 9 具有子功能的请求 A_PDU 定义

A_PDU 参数	参数名称	约定值	字节值	助记符
MType	消息类型	M	XX	MT
SA	源地址	M	XXXX	SA
TA	目标地址	M	XXXX	TA
TAtype	目标地址类型	M	XX	TAT
RA	远程地址	C	XXXX	RA
A_Data.A_PCI.SI	<服务名称>请求识别码 SID	M	XX	SIDRQ
A_Data.Parameter 1	子功能 = [参数]	S	XX	LEV_PARAM
A_Data.Parameter 2	数据参数#1	U	XX	DP_...#1
.
A_Data.Parameter k	数据参数#k-1	U	XX	DP_...#k-1
Length	A_Data 长度	M	XXXXXXXX	LGT

C: RA（远程地址）PDU 参数仅在远程寻址的情况下才存在。

表 10 定义了不具有子功能的请求 A_PDU 定义。

表 10 不具有子功能的请求 A_PDU 定义

A_PDU 参数	参数名称	约定值	字节值	助记符
MType	消息类型	M	XX	MT
SA	源地址	M	XXXX	SA
TA	目标地址	M	XXXX	TA
TAtype	目标地址类型	M	XX	TAT
RA	远程地址	C	XXXX	RA
A_Data.A_PCI.SI	<服务名称>请求识别码 SID	M	XX	SIDRQ
A_Data.Parameter 1	数据参数#1	U	XX	DP_...#1
.
A_Data.Parameter k	数据参数#k	U	XX	DP_...#k-1

Length	A_Data 长度	M	XXXXXXXX	LGT
C: RA (远程地址) PDU 参数仅在远程寻址的情况下才存在。				

在所有请求/指示中，寻址信息 MType, TA, SA, TAtype 和 Length 是强制性的。寻址信息 RA 是可选的。

注：寻址信息如表 10 所示。服务请求/指示定义仅指定 A_Data A_PDU 参数，因为 A_Data A_PDU 参数代表了请求/指示服务的消息数据字节。

5.4.2.2 请求消息子功能参数\$Level (LEV_) 定义

本节定义了为服务<服务名称>请求/服务指示定义的子功能参数\$Level (LEV_)

如果所描述的服务不使用子功能参数值并且不使用抑制肯定响应指示位（隐含地表示需要响应），则该子节不包含任何定义。

子功能参数字节分为两个部分（按位），见表 11。

表 11 子功能参数结构

位的位置	描述
7	抑制肯定响应指示位 该位表示服务端是否应该抑制肯定响应消息。 '0' = FALSE, 不要抑制肯定响应消息（即需要肯定响应消息）。 '1' = TRUE, 禁止肯定响应消息（不应发送肯定响应消息；服务器不应发出肯定响应信息）。 与抑制肯定响应指示位无关，否定响应消息由服务端根据7.5中的规定发送。 即使不需要肯定响应（即SPRMIB = true），也必须执行服务，以保持实现一致性，无论SPRMIB值如何。 对于任何给定服务，服务端支持的所有子功能参数值（即子功能结构的位6-0）都应支持抑制肯定响应指示位，其值为'0' 和 '1'。
6-0	子功能参数值 子功能参数的第1-7位包含服务的子功能参数值 (0x00 – 0x7F)。

子功能参数值包含七个位（即子功能参数字节的位 6 到 0），该子功能参数可以有多个值来进一步指定服务行为。

除了抑制肯定响应指示位之外，支持子功能参数值的服务应支持子功能参数值表中定义的子功能参数值。每个服务包含一个表，用于定义子功能参数值的值，仅考虑第 1-7 位。

重要：包含大量数据需要使用分页缓冲区的响应，如果 SPRMIB 设置为 TRUE，第一批数据的传输仍然可以在响应时间窗口内启动，但是服务执行的终止会超出响应时间窗口的限制。如果在这种情况下响应被抑制，则无法将延迟信息通知给客户端，但是服务端仍然很忙，还没有准备好接收另一个请求。对于客户端，本标准建议不要在请求大量数据的情况下设置 SPRMIB（例如，SID 0x19 SF 0x0A），因为这将打破 SPRMIB 的目的。对于服务端实现，建议发送 NRC 0x78 (RCRRP)，并随后发送肯定响应，以防在 SPRMIB 为 TRUE 时使用分页缓冲区处理。

表 12 定义了请求消息的子功能参数定义。

表 12 请求消息子功能参数定义

位 6-0	描述	约定值	助记符
XX	子功能#1 子功能参数#1的描述	M/U	SUBFUNC1
.			
XX	子功能#m 子功能参数#m的描述	M/U	SUBFUNCm

表 12 中 Cvt 列的定义如表 13 所示。

表 13 子功能参数约定值

类型	名称	描述
M	强制	如果支持该服务，则服务端必须支持子功能参数。
U	用户自定义	根据服务的使用情况，服务端可能支持或不支持子功能参数。

完整的子功能参数字节值是根据抑制肯定响应指示位的值和选择的子功能参数值计算的。

表 14 定义了子功能字节值的计算。

表 14 子功能字节值的计算

第 7 位	第 6 位	第 5 位	第 4 位	第 3 位	第 2 位	第 1 位	第 0 位
抑制肯定响应标志位	子功能参数值在服务的子功能参数值表中指定						
产生子功能参数字节值 (位 7 - 0)							

5.4.2.3 请求消息数据参数定义

该节定义了服务<服务名称>请求/指示的数据参数\$ DataParam (DP_)。如果所描述的服务不使用任何数据参数，则该节中不包含任何定义。数据参数部分可以包含多个字节。该节包含数据参数的通用描述。详细定义参见本标准的附录。具体附录取决于服务。附录还指示了数据参数是否应被支持或在服务端支持该服务的情况下为用户可选。

表 15 定义了请求消息数据参数。

表 15 请求消息数据参数定义

定义
数据参数#1 数据参数#1 的描述
.
数据参数#n 数据参数#n 的描述
.

5.4.3 肯定响应消息

5.4.3.1 肯定响应消息定义

本节定义了响应/确认服务的 A_PDU 参数（详见 7.2 中应用层协议数据单元 A_PDU 的描述）。当不同子功能参数\$ Level 的响应消息在 AData 参数的结构上存在差异时，每个子功能参数\$ Level 都可能有一个单独的表。

注：诊断服务的肯定响应消息（如果需要）应在执行诊断服务后发送。如果诊断服务需要不同的处理（例如，ECUReset 服务），则可以在诊断服务的服务描述中找到何时发送肯定响应消息的适当描述。

表 16 定义了 A_PDU 的肯定响应。

表 16 A_PDU 肯定响应

A_PDU 参数	参数名称	约定值	字节值	助记符
SA	源地址	M	XXXX	SA
TA	目标地址	M	XXXX	TA
TAtype	目标地址类型	M	XX	TAT
RA	远程地址	C	XXXX	RA
A_Data.A_PCI.SI	<服务名称>请求识别码 SID	M	XX	SIDRQ
A_Data.Parameter 1	数据参数#1	U	XX	DP_...#1
.
A_Data.Parameter k	数据参数#k	U	XX	DP_...#k
C: RA (远程地址) PDU 参数仅在远程寻址的情况下才存在。				

在所有响应/确认中，寻址信息 TA, SA 和 TAtype 是强制性的。寻址信息 RA 是可选的。

注：寻址信息如表 16 所示。因为 A_Data A_PDU 参数表示服务响应/确认的消息数据字节，所以进一步的响应/确认服务定义仅指定 A_Data A_PDU 参数。

5.4.3.2 肯定响应消息数据参数定义

该节定义了服务<服务名称>响应/确认的数据参数。

如果所描述的服务不使用任何数据参数，则该节不包含任何定义。数据参数部分可以包含多个字节。

该节包含数据参数的通用描述。详细定义参见本标准的附录。具体附录取决于服务。附录还指示了数据参数是否应被支持或在服务端支持该服务的情况下为用户可选。

表 17 定义了响应数据参数。

表 17 响应数据参数

定义
数据参数#1 数据参数#1 的描述。如果请求支持子功能参数字节，则此参数与请求消息的子功能参数字节中包含的第 7 位子功能参数值相同，且位 7 设置为零。

.
数据参数#n 数据参数#n 的描述。

5.4.4 支持的否定响应码 (NRC_)

本节定义了针对服务的否定响应码。每个响应代码将发生的情况记录在表 18 中。否定响应消息的定义参见 7.4。服务端应使用否定响应 A_PDU 来指示识别的错误条件。

除了服务描述中指定的否定响应码（如果适用）外，还应使用附录 A.1 中列出的否定响应码。具体参见附录 A.1。

表 18 定义了支持的否定响应码。

表 18 支持的否定响应码

否定响应码	描述	助记符
0xXX	否定响应码#1 1. 条件#1 m. 条件#m	NRC_
.	.	NRC_
0xXX	否定响应码#n 1. 条件#1 k. 条件#k	NRC_

5.4.5 消息流示例

该节包含服务<服务名称>的消息流示例。所有示例中都只显示了诊断消息（无寻址信息）。

表 19 定义了请求消息流示例。

表 19 请求消息流示例

消息方向		客户端→服务端	
消息类型		请求	
A_Data	描述（所有值都是十六进制）	字节值	助记符
#1 (A_PCI)	<服务名称>响应代码 SID	0xXX	SIDPR
#2	数据参数#1	0xXX	LEV_/DP_
#n	数据参数#n-1	0xXX	DP_

表 20 定义了肯定响应消息流程示例。

表 20 肯定响应消息流示例

消息方向		服务端→客户端	
消息类型		响应	
A_Data	描述（所有值都是十六进制）	字节值	助记符
#1 (A_PCI)	<服务名称>响应 SID	0xXX	SIDRQ
#2	数据参数#1	0xXX	DP_
#n	数据参数#m	0xXX	DP_

如果适用于服务<服务名称>（例如每个子功能参数\$ Level），会给出多个示例。

表 21 显示了否定响应消息的消息流示例。

表 21 否定响应消息的消息流示例

消息方向		服务端→客户端	
消息类型		响应	
A_Data	描述（所有值都是十六进制）	字节值	助记符

#1 (A_PCI.NR_SI)	否定响应 SID	0x7F	SIDRSIDNRQ
#2 (A_PCI_SI)	<服务名称>请求 SID	0xXX	SIDRQ
#3	响应码	0xXX	NRC_

5.5 诊断和通信管理功能单元

5.5.1 概述

表 22 定义了诊断和通信管理功能单元。

表 22. 诊断和通信管理功能单元

服务	描述
诊断会话控制	客户端请求控制与服务端的诊断会话。
电子控制器单元(ECU)复位	客户端强制服务端执行重置。
安全访问	客户端向服务端请求安全解锁。
通信控制	客户端控制服务端中通信参数的设置（例如，通信波特率）。
诊断设备在线	客户端向服务端指示诊断设备仍然连接在网络上。
访问计时参数	客户端使用该服务来读取或修改活动通信的计时参数。
安全数据传输	客户端使用该服务以扩展数据链路安全性来执行数据传输。
诊断故障码(DTC)控制设置	客户端控制服务端中的诊断故障码(DTC)设置。
事件响应	客户端请求在服务端中设置和/或控制事件机制。
链路控制	客户端请求控制通信波特率。

5.5.2 诊断会话控制 (0x10) 服务

5.5.2.1 服务描述

诊断会话控制服务用于在服务端中启用不同的诊断会话。

诊断会话在服务端中启用一组特定的诊断服务和/或功能集。该服务为服务端提供这样一种能力，即报告对当前诊断会话有效的数据链路层特定参数值（例如，计时参数值）的能力。本标准的用户应在每个诊断会话中明确地定义可执行的服务和/或功能。

在服务端中应该总是有一个诊断会话处于活动状态。服务端在上电时应进入默认诊断会话。如果没有启动其他诊断会话，那么服务端应保持默认诊断会话。

服务端应该能够在正常运行条件和车辆制造商定义的其他操作条件下（例如，跛行模式）提供诊断功能。

如果客户端请求了一个正在运行的诊断会话，那么服务端应该发送一个肯定响应，并按照图 7 执行。图 7 描述了服务端在会话之间转换时的内部行为。

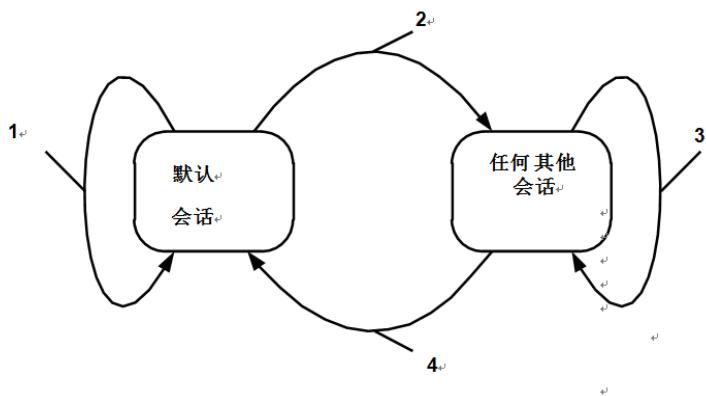
每当客户端请求新的诊断会话时，服务端应在服务器中的新会话计时激活之前发送诊断会话控制肯定响应消息。有些情况可能需要在发送肯定响应之前进入新会话，同时维持用于旧的协议发送响应的时间。如果服务端无法启动请求的新诊断会话，那么它应发送诊断会话控制否定响应消息，并且当前会话将继续（参见诊断会话参数定义以获取服务端和客户端的更多信息）。非默认诊断会话（不包括编程会话）中的诊断服务和诊断功能集是默认会话中支持的功能的超集，这意味着切换到任何非默认诊断会话时，在默认会话支持的诊断功能也是可用的。某个会话可以使能车辆制造商特定的服务和功能，但是这不属于本标准的内容。

要启动新的诊断会话，服务端可能会要求满足某些条件。所有这些条件都是用户定义的。举例如下：

- 服务端可能只允许具有特定标识符（客户端诊断地址）的客户端启动特定的新诊断会话（例如，服务端可能要求只有标识符为 0xF4 的客户端才能启动扩展诊断会话）。
- 可能需要满足某些安全条件（例如，车辆不应移动或发动机不能运转）。

在一些系统中，当新的诊断会话开始时，希望改变通信计时参数。诊断会话控制服务实体可以使用适当的服务原语改变底层指定的计时参数，以改变本地节点中的以及客户端想要与之通信的潜在节点中的通信计时。

图 7 提供了有关诊断会话转换的概述以及服务端在转换到其他会话时应执行的操作。



图例

- 1 默认会话：当服务端在默认会话中且客户端请求启动默认会话时，服务端应该完全重新初始化默认会话。在激活的会话期间，服务端应重置所有的激活/启动/更改的设置/控制。这不包括被编程到非易失性存储器中的长期变化。
- 2 其他会话：当服务端从默认会话转换到除默认会话之外的任何其他会话时，服务端只应通过在默认会话期间的响应事件（0x86）服务停止已在服务端中配置的事件（类似于停止响应事件）。
- 3 相同或其他会话：当服务端从除默认会话以外的任何诊断会话转换到默认会话以外的其他会话（包括当前活动的诊断会话）时，服务端应（重新）初始化诊断会话，这意味着：
 - i) 通过响应事件（0x86'）服务在服务端中配置的每个事件都应停止。
 - ii) 应重新锁定安全性。请注意，安全访问的锁定应将任何依赖于安全访问的活动诊断功能重置为未锁定状态（例如，数据标识符 DID 的活动输入输出控制）。
 - iii) 新会话中支持且不依赖安全访问的所有其他活动诊断功能应予以保留。例如，任何已配置的周期性调度程序在从一个非默认会话切换到另一个或同一个非默认会话时应保持活动状态，并且不会影响通信控制和控制诊断故障码(DTC)设置服务的状态，这意味着正常通信应在会话切换的时间点被禁用时保持禁用状态。
- 4 默认会话：当服务端从默认会话以外的任何诊断会话转换到默认会话时，服务端应通过响应事件（0x86'）服务停止已在服务端中配置的每个事件，并且应启用安全性。默认会话中不支持的任何其他活动诊断功能均应终止。例如，任何已配置的周期调度程序或输出控制都应被禁用，并且通信控制和控制诊断故障码(DTC)设置服务的状态应重置，这意味着当正常通信在会话切换到默认会话的时间点被禁用时应该被重新启用。在激活的会话期间，服务端应重置所有激活/启动/更改的设置/控制。这不包括被编程到非易失性存储器中的长期变化。

图 7 — 服务端诊断会话状态图

表 23 定义了在默认会话和非默认会话期间（定时服务）允许的服务。任何非默认会话都与诊断会话计时器相关联，客户端应将计时器保持活动状态。

表 23 — 在默认和非默认诊断会话期间允许的服务

服务	默认会话	非默认会话
诊断会话控制—0x10	x	x
ECU 复位 — 0x11	x	x
安全访问—0x27	不适用	x
通信控制—0x28	不适用	x
诊断设备在线—0x3E	x	x
访问计时参数—0x3E	不适用	x
安全数据传输—0x84	不适用	x
控制 DTC 设置—0x85	不适用	x
响应事件—0x86	x ^a	x
链路控制—0x87	不适用	x
按标识符读取数据—0x22	x ^b	x
按地址读取内存—0x23	x ^c	x

按标识符读取缩放数据—0x24	x ^b	x
按周期标识符读取数据—0x2A	不适用	x
动态定义数据标识符—0x2C	x ^d	x
按标识符写入数据—0x2E	x ^b	x
按地址写入内存—0x3D	x ^c	x
清除诊断信息—0x14	x	x
读取 DTC 信息—0x19	x	x
按标识符输入输出控制—0x2F	不适用	x
例行控制—0x31	x ^e	x
请求下载—0x34	不适用	x
请求上传—0x35	不适用	x
传输数据—0x36	不适用	x
请求传输退出—0x37	不适用	x
请求文件传输—0x38	不适用	x
在默认会话期间是否也允许响应事件服务由具体实施情况而定。 安全数据标识符需要安全访问服务，因此需要非默认诊断会话。		
^c	安全内存区域需要安全访问服务，因此需要非默认诊断会话。	
^d	数据标识符可以在默认和非默认诊断会话中动态定义。	
^e	安全例行程序需要安全访问服务，因此需要非默认诊断会话。需要客户端主动停止的例行程序也需要非默认会话。	

重要 —— 服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

5.5.2.2 请求消息

5.5.2.2.1 请求消息定义

表 24 定义了请求消息。

表 24 — 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	诊断会话控制请求服务标识符 SID	M	0x10	DSC
#2	子功能=[诊断会话类型]	M	0x00 – 0xFF	LEV_DS_

5.5.2.2.2 请求消息子功能参数\$level (LEV_) 定义

诊断会话控制服务使用子功能参数“诊断会话类型”来选择服务端的特定行为。表 25 详细说明了诊断会话的解释和用法并定义了以下子功能（抑制肯定响应消息指示位（位 7）未显示）。

表 25 — 请求消息子功能参数定义

位 6-0	描述	Cvt	助记符
0x00	ISOSAE 保留 此值由本标准保留。	M	ISOSAERESRVD

0x01	<p>默认会话</p> <p>此诊断会话启用服务端中的默认诊断会话，并且不支持任何诊断应用程序超时处理规定（例如，不需要使用诊断设备在线服务将会话保持在活动状态）。</p> <p>如果服务端处于非默认诊断会话时，再次启动默认会话，应遵循以下实施规则（另请参阅上面给出的服务端诊断会话状态图）：</p> <p>服务端在发送诊断肯定响应消息时，应停止当前诊断会话，并应在之后启动新请求的诊断会话。</p> <p>如果服务端发送了诊断会话控制肯定响应消息，则客户端应重新锁定服务端（如果该服务端在诊断会话期间被客户端解锁）。</p> <p>如果服务端发送带有诊断会话控制请求服务标识符的否定响应消息，则应继续将会话保持在激活状态。</p> <p>注：如果使用的数据链路需要初始化步骤，则初始化的服务端应默认启动默认诊断会话。初始化步骤后，无需将诊断会话设置为默认会话的诊断会话控制。</p>	M	DS
------	---	---	----

表 25 — (续)

位 6-0	描述	Cvt	助记符
0x02	<p>编程会话</p> <p>此诊断会话启用所有用于支持服务端内存编程的诊断服务。</p> <p>如果服务端在引导软件中运行编程会话，则只能通过三种方式退出编程会话：客户端启动的 ECU 复位(0x11)服务、会话类型等于默认会话的诊断会话控制(0x10)服务或服务端中的会话层超时。</p> <p>如果服务端在引导软件运行时接收到默认诊断会话控制 (0x10) 服务或会话层超时，且存在有效的应用程序软件，则服务端应重新启动应用程序软件。本标准没有明确如何实现有效应用软件重新启动的各种实施方法（例如，在引导软件中可以直接确定有效的应用软件，在 ECU 启动阶段执行 ECU 复位等）。</p>	U	PRGS
0x03	<p>扩展诊断会话</p> <p>此诊断会话可用于启用支持调整服务端内存中诸如“怠速、CO 值等”功能所需的所有诊断服务。也可以用于启用诊断服务，而这些诊断服务与功能调整没有特别的关联（例如，参考表 23 中的定时服务）。</p>	U	EXTDS
0x04	<p>安全系统诊断会话</p> <p>该诊断会话启用支持与安全系统相关功能（例如，安全气囊展开）所需的所有诊断服务。</p>	U	SSDS
0x05 – 0x3F	<p>ISOSAE 保留</p> <p>此值由本文档保留以备将来定义。</p>	M	ISOSAERESRVD
0x40 – 0x5F	<p>车辆制造商特定</p> <p>此值的范围保留为车辆制造商的特定用途。</p>	U	VMS
0x60 – 0x7E	<p>系统供应商特定</p> <p>此值的范围保留为系统供应商的特定用途。</p>	U	SSS
0x7F	<p>ISOSAE 保留</p> <p>此值由本文档保留以备将来定义。</p>	M	ISOSAERESRVD

5.5.2.2.3 请求消息数据参数定义

此服务不支持请求消息中的数据参数。

5.5.2.3 肯定响应消息

5.5.2.3.1 肯定响应消息定义

表 26 定义了肯定响应消息。

表 26 — 肯定响应消息定义

一个数据字节	参数名	Cvt	字节值	助记符
#1	诊断会话控制响应服务标识符 SID	M	0x50	DSCPR
#2	子功能=[诊断会话类型]	M	0x00 – 0xFF	LEV_DS_
#3 : #6	会话参数记录[]#1:=[数据#1 : 数据#4]	M : M	0x00 – 0xFF : 0x00 – 0xFF	SPREC_ DATA_1 : DATA_m

5.5.2.3.2 肯定响应消息数据参数定义

表 27 定义了响应消息数据参数。

表 27 — 响应消息数据参数定义

定义
诊断会话类型
此参数与请求消息子功能参数的位 6~0 相同。
会话参数记录
此参数记录包含服务端报告的会话特定参数值。表 28 和表 29 中定义了会话参数记录的内容。

表 28 和表 29 明确了响应消息数据参数“会话参数记录”的结构，适用于在支持的数据链路上实现该服务。

表 28 — 会话参数记录定义

记录中的字节位置	描述	Cvt	字节值	助记符
#1	会话参数记录[]=[M	0x00 – 0xFF	SPREC_
#2	P2Server_max (高字节) P2Server_max (低字节)	M	0x00 – 0xFF	P2SMH
#3	P2*Server_max (高字节) P2*Server_max (低字节)]	M	0x00 – 0xFF	P2SML
#4			0x00 – 0xFF	P2ESMH P2ESML

表 29 — 会话参数记录内容定义

参数	描述	字节数量	分辨率	最小值	最大值
P2Server_max	服务端支持的默认 P2Server_max 计时，用于当前诊断会话	2	1 ms	0 ms	65 535 ms

P2*Server_max	服务端支持的增强 (NRC 0x78) P2Server_max, 用于当前诊断会话	2	10 ms	0 ms	655 350 ms
---------------	--	---	-------	------	------------

有关 P2 服务端和 P2*服务端的详细信息，参见 ISO14229—2。

5.5.2.4 支持的否定响应码 (NRC_)

该服务应实施下列否定响应码。表 30 中记录了每个响应码将发生的情况。如果服务端检测到错误，则应使用列出的否定响应。

表 30 — 支持的否定响应码

否定响应码	描述	助记符
0x12	不支持该子功能 如果不支持该子功能参数，则应发送此否定响应码。	SFNS
0x13	不正确的消息长度或无效格式 如果消息长度错误，则应发送此否定响应码	IMLOIF
0x22	条件不正确 如果未满足请求诊断会话控制的条件，则应返回此否定响应码。	CNC

5.5.2.5 诊断会话控制消息流示例

5.5.2.5.1 示例 1 — 启动编程会话

此消息流显示了如何在服务端中启用诊断会话“编程会话”。客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误”（“0”）来请求获得响应消息。给定的示例假定 P2 服务端最大值等于 50 毫秒，P2* 服务端最大值等于 5,000 毫秒。

表 31 定义了诊断会话控制请求消息流示例 1。

表 31 — 诊断会话控制请求消息流示例 1

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	诊断会话控制请求服务标识符 SID	0x10	DSC
#2	诊断会话类型=编程会话 抑制肯定响应消息指示位=false	0x02	DS_ECUPRGS

表 32 定义了诊断会话控制肯定响应消息流示例 1。

表 32 — 诊断会话控制肯定响应消息流示例 1

消息方向	服务端→客户端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	诊断会话控制响应服务标识符 SID	0x50	DSCPR
#2	诊断会话类型=编程会话	0x02	DS_ECUPRGS
#3	会话参数记录[P2 服务端最大 (高字节)]	0x00	SPREC_1
#4	会话参数记录[P2 服务端最大 (低字节)]	0x32	SPREC_2
#5	会话参数记录[P2*服务端最大 (高字节)]	0x01	SPREC_3

#6	会话参数记录[P2*服务端最大(低字节)]会话参数记录[P2*服务端最大(低字节)]	0xF4	SPREC_4
----	--	------	---------

5.5.3 ECU 复位 (0x11) 服务

5.5.3.1 服务描述

客户端使用 ECU 复位服务来请求服务端重置。

该服务请求服务端根据嵌入在 ECU 复位请求消息中的复位类型参数值的内容有效地执行服务端重置。ECU 复位肯定响应消息 (如果需要) 应在服务端执行重置之前发送。服务端重置成功后, 将激活默认会话。

重要 —— 服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

本标准没有对从对 ECU 复位请求肯定响应消息发送之后到复位成功完成这段时间内的 ECU 的行为进行定义。建议在此期间 ECU 不接受任何请求消息, 也不发送任何响应消息。

5.5.3.2 请求消息

5.5.3.2.1 请求消息定义

表 33 定义了请求消息。

表 33 — 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	ECU 复位请求服务标识符 SID	M	0x11	ER
#2	子功能=[复位类型]	M	0x00 – 0xFF	LEV_RT_

5.5.3.2.2 请求消息子功能参数\$Level (LEV_) 定义

ECU 复位请求消息使用子功能参数“复位类型”来描述服务端必须如何执行重置 (抑制肯定响应消息指示位 (位 7) 未显示)。

表 34 定义了请求消息子功能参数。

表 34 — 请求消息子功能参数定义

位 6-0	描述	Cvt	助记符
0x00	ISOSAE 保留 该值由本文档保留。	M	ISOSAERESRVD
0x01	硬复位 该值规定了“硬复位”条件, 该条件模拟通常在服务端先前已与其电源 (即电池) 断开之后执行的通电/启动流程。执行的操作由具体实施情况而定, 本标准未定义。这可能导致易失性存储器和非易失性存储器位置被重新初始化为预定值。	U	HR

表 34 — (续)

位 6-0	描述	Cvt	助记符
0x02	钥匙开关复位 该值表示与司机关闭点火钥匙又重新打开相似的情况。此复位条件应该模拟一个钥匙开关顺序（即中断开关电源）。执行的操作由具体实施情况而定，没有标准定义。通常地，非易失性存储器位置的值被保留；易失性存储器将被初始化。	U	KOFFONR
0x03	软复位 该值表示“软复位”条件，该条件会使服务端可以立即重新启动应用程序（如果适用的话）。执行的操作由具体实施情况而定，而没有标准定义。典型的操作是重新启动应用程序，而无需重新初始化先前学到的配置数据、自适应因素和其他长期调整。	U	SR
0x04	启用快速电源关断 这个子功能适用于电池供电而非点火供电的 ECU。因此，关机会强制进入休眠模式而非关断电源。睡眠意味着虽然断电，但仍然可以唤醒（电池供电）。子功能的意图是在点火转换到关闭位置后减少 ECU 的待机时间。 此值请求服务端启用并执行“快速电源关断”功能。一旦关闭“钥匙/点火”，服务端应立即执行该功能。当服务端执行断电功能时，应直接或在限定的准备时间之后转换到休眠模式。如果客户端需要响应消息，并且服务端已经准备好执行“快速电源关断”功能，则服务端应在“快速电源关断”功能开始之前发送肯定响应消息。下一次出现“钥匙开启”或“点火开启”信号时会终止“快速电源关断”功能。 注意该子功能仅适用于支持待机模式的服务端！	U	ERPSD
0x05	禁用快速电源关断 此值请求服务端禁用先前启用的“快速电源关断”功能。	U	DRPSD
0x06 – 0x3F	ISOSAE 保留 此值的范围由本文档保留以备将来定义。	M	ISOSAERESRVD
0x40 – 0x5F	车辆制造商特定 此值的范围保留为车辆制造商的特殊用途。	U	VMS
0x60 – 0x7E	系统供应商特定 此值的范围保留为系统供应商的特殊用途。	U	SSS
0x7F	ISOSAE 保留 此值由本文档保留以备将来定义。	M	ISOSAERESRVD

5.5.3.2.3 请求消息数据参数定义

此服务不支持请求消息中的数据参数。

5.5.3.3 肯定响应消息

5.5.3.3.1 肯定响应消息定义

表 35 定义了肯定响应消息。

表 35 — 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	ECU 复位响应服务标识符 SID	M	0x51	ERPR
#2	子功能= [复位类型]	M	0x00 – 0x7F	LEV_RT_
#3	断电时间	C	0x00 – 0xFF	PDT
C: 如果将子功能参数设置为启用快速电源关断值 (0x04)，则存在此参数；				

5.5.3.3.2 肯定响应消息数据参数定义

表 36 定义了响应消息的数据参数。

表 36 — 响应消息数据参数定义

定义
复位类型 此参数与请求消息的子功能参数的位 6~0 相同。
断电时间 此参数向客户端指示服务端在断电过程中保持待机状态的最小时间。该参数的分辨率为每计数一 (1) 秒。以下值有效： —— 0x00 — 0xFE: 0 — 254 秒断电时间， —— 0xFF: 表示故障或时间不可用。

5.5.3.4 支持的否定响应码 (NRC_)

该服务应实施下列否定响应码。表 37 中定义了每个响应码发生的情况。如果错误情形适用于服务端，则应使用列出的否定响应。

表 37 支持的否定响应码

否定响应码	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS
0x13	消息长度错误或格式无效 如果消息的长度错误，则应发送此否定响应码。	IMLOIF
0x22	条件错误 如果未满足 ECU 复位请求的条件，则应返回此否定响应码。	CNC
0x33	安全访问拒绝 如果请求的重置处于安全状态且服务端未处于解锁状态，则应发送此否定响应码。	SAD

5.5.3.5 ECU 复位消息流示例

该节规定了在服务端中成功执行 ECU 复位服务（该示例）所满足的条件。

服务端状态：(ignition = on,)点火=开启，系统不应处于运行模式（例如，如果系统是发动机管理模块，则发动机应该关闭）。

客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“False”来请求获得响应消息。

在服务端执行复位类型之前，该服务端应发送 ECU 复位的肯定响应消息。

表 38 定义了 ECU 复位请求消息流示例 1。

表 38 — ECU 复位请求消息流示例 1

消息方向		客户端→服务端	
消息类型		请求	
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	ECU 复位请求服务标识符 SID	0x11	ER
#2	复位类型=硬复位， 抑制肯定响应消息指示位=false	0x01	RT_HR

表 39 定义了 ECU 复位的肯定响应消息流示例 1。

表 39 — ECU 复位肯定响应消息流示例 1

消息方向		服务端→客户端	
消息类型		响应	
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	ECU 复位响应服务标识符 SID	0x51	ERPR
#2	复位类型=硬复位	0x01	RT_HR

5.5.4 安全访问 (0x27) 服务

5.5.4.1 服务描述

此服务的目的是提供访问数据和/或诊断服务的手段，这些服务由于保密、排放或安全的原因而受到限制。用于将例程或数据下载或上传到服务端和从服务端读取特定内存位置的诊断服务，可能需要在安全访问的情况下进行。下载到服务端的非正常例程或数据可能会损坏电子设备或其他车辆部件，或者影响车辆排放、保密或安全标准的符合性。**保密概念**利用了种子和密钥之间的关系。

使用该服务的典型示例如下所示：

- 客户端请求“种子”，
- 服务端发送“种子”，
- 客户端发送“密钥”（与接收的种子配对），
- 服务端响应“密钥”有效，并且它将自行解锁。

“请求种子”子功能参数值应始终为奇数，并且同一安全级别对应的“发送密钥”子功能参数值应等于“请求种子”子功能参数值加上 1。

在任何时刻，都只能有一个安全级别处于活动状态。例如，如果与请求种子 0x03 相关联的安全级别处于活动状态，并且诊断设备成功请求解锁与种子 0x01 相关联的安全级别，那么将只有与请求种子 0x01 相关联的安全级别支持的安全功能可以被解锁。先前与请求种子 0x03 相关联的安全级别解锁的任何其它安全功能将不再处于活动状态。安全级别编号是任意的，并不意味着级别之间有任何关系。

客户端应通过发送安全访问服务“请求种子”消息请求服务端“解锁”。服务端应通过使用安全访问服务“请求种子”肯定响应消息发送“种子”来响应。然后客户端应通过使用适当的安全访问服务“发送密钥”请求消息并将“密钥”数字返回服务端来进行回应。服务

端应将此“密钥”与内部存储/计算的密钥进行比较。如果两个数字匹配，则服务端应解锁客户端对特定服务/数据的访问，并通过安全访问服务“发送密钥”肯定响应消息来进行指示。如果这两个数字不匹配，则应将其视为错误访问尝试。无效密钥要求客户端从安全访问“请求种子”消息重新开始（有关安全访问处理详细信息，参考附录 I）。

如果服务端支持安全访问，但当接收到安全访问“请求种子”消息时已解锁所请求的安全级别，则该服务端应使用种子值等于零（0）的安全访问“请求种子”肯定响应消息服务进行响应。服务端不应为当前锁定的给定安全级别发送全零种子。客户端应使用该方法通过检查非零种子来确定服务端是否对特定的安全级别进行锁定。

在服务端上电/复位后或在一定的错误访问尝试次数（参见下文的进一步描述）之后，并在服务端对客户端的安全访问服务“请求种子”消息可以作出积极回应之前，可能会需要车辆制造商定义时间延迟。如果此延迟计时器被支持，则此延时应在达到车辆制造商指定的错误访问尝试次数之后或当服务端上电/复位且先前执行的安全访问服务由于单次错误访问尝试而失败时被激活。如果服务端支持此延迟计时器，那么在成功的安全访问服务“发送密钥”执行之后，服务端应清除在上电/复位时为延迟计时器调用的服务端内部指示信息。如果服务端支持此延迟计时器，并且无法确定在上电/复位之前先前执行的安全访问服务是否失败，那么在上电/复位之后延迟计时器应始终处于活动状态。只有在服务端在上电/复位时被锁定的情况下才需要延迟。车辆制造商应选择是否支持延迟计时器。

尝试进行安全访问不应妨碍正常的车辆通信或其他诊断通信。

如果在服务端被锁定时请求安全服务，则服务端应给出否定响应。

在特定诊断会话期间请求的一些诊断功能/服务可能需要成功的安全访问顺序。在这种情况下，需要以下一系列服务：

- 诊断会话控制服务，
- 安全访问服务，
- 被保护的诊断服务。

服务端中启用的诊断会话（开始的会话）允许不同的访问模式。

重要 —— 服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

5.5.4.2 请求消息

5.5.4.2.1 请求消息定义

表 40 定义了请求消息定义子功能=请求种子。

表 40 —请求消息定义—子功能=请求种子

一个数据字节	参数名称	Cvt	字节值	助记符
#1	安全访问请求服务标识符 SID	M	0x27	SA
#2	子功能=[安全访问类型=请求种子]	M		LEV_
#3 ： #n	安全访问数据记录[]=[参数#1 ： 参数#m]	U ：U	0x00 – 0xFF ： 0x00 – 0xFF	SECACCDR_ PARA1 ： PARAm

表 41 定义了请求消息—子功能=发送密钥。

表 41 —请求消息定义—子功能=发送密钥

一个数据字节	参数名称	Cvt	字节值	助记符
#1	安全访问请求服务标识符 SID	M	0x27	SA
#2	子功能=[安全访问类型=发送密钥]	M	0x02, 0x04, 0x06, 0x08 – 0x7E	LEV_SAT_SK

#3 : #n	安全密钥[]=[密钥#1 (高字节) : 密钥#m (低字节)	M : U	0x00 – 0xFF : 0x00 – 0xFF	SECKEY_ KEY1HB : KEYmLB
---------------	---	----------	---------------------------------	----------------------------------

5.5.4.2.2 请求消息子功能参数\$Level (LEV_) 定义

子功能参数“安全访问类型”向服务端指示该服务正在进行的步骤、客户端想要访问的安全级别以及种子和密钥的格式。如果服务端支持不同级别的安全性，则每个级别应由请求种子值来认定，该请求种子值与发送密钥值具有固定关系：

- “请求种子=0x01”确定了“请求种子=0x01”和“发送密钥=0x02”之间的固定关系。
- “请求种子=0x03”确定了“请求种子=0x03”和“发送密钥=0x04”之间的固定关系，表 42 中为请求种子和发送密钥（抑制肯定响应消息指示位（位 7）未显示）定义了值。

表 42 — 请求消息子功能参数定义

位 6-0	描述	Cvt	助记符
0x00	ISOSAE 此值由本文档保留。	M	ISOSAERESRVD
0x01	请求种子 由车辆制造商定义的具有安全级别的请求种子。	U	RSD
0x02	发送密钥 由车辆制造商定义的具有安全级别的发送密钥。	U	SK
0x03, 0x05, 0x07 – 0x41	请求种子 由车辆制造商定义的具有不同安全级别的请求种子。	U	RSD
0x04, 0x06, 0x08 – 0x42	发送密钥 由车辆制造商定义的具有不同级别的发送密钥。	U	SK
0x43 – 0x5E	ISOSAE 保留 此值由本文档保留以备将来定义。	M	ISOSAERESRVD
0x5F	ISO26021—2 值 对具有不同安全级别的请求种子的定义是为了激活 ISO 26021—2 中定义的报废期的车载打火装置。	U	RSD
0x60	ISO26021—2 发送密钥值 对具有不同安全级别的发送密钥的定义是为了激活 ISO 26021—2 中定义的报废期的车载打火装置。	U	SK
0x61 – 0x7E	系统供应商特定 此值的范围保留为系统供应商的特殊用途。	U	SSS
0x7F	ISOSAE 保留 此值由本文档保留以备将来定义。	M	ISOSAERESRVD

5.5.4.2.3 请求消息数据参数定义

表 43 定义了请求消息的数据参数。

表 43 请求消息数据参数定义

定义
安全密钥（高字节和低字节） 请求消息中的“密钥”参数是由对应于特定“种子”值的安全算法生成的值。
安全访问数据记录 此参数记录是用户可选的，用于在请求种子信息时将数据传输到服务端。例如，它可以包含服务端中验证的客户端的身份识别。

5.5.4.3 肯定响应消息

5.5.4.3.1 肯定响应消息定义

表 44 定义了肯定响应消息。

表 44 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	安全访问响应服务标识符 SID	M	67	SAPR
#2	子功能= [安全性访问类型]	M	00—7F	LEV_SAT_SK
#3 ： #n	安全种子[] =[种子#1 (高字节) ： 种子#m (低字节)]	C : C	0x00 – 0xFF ： 0x00 – 0xFF	SECSEED_ SEED1HB ： SEEDmLB

C: 此参数的存在取决于安全访问类型参数。如果安全访问类型参数指示客户端想要从服务端获取种子，则必须存在。

5.5.4.3.2 肯定响应消息数据参数定义

表 45 定义了响应消息的数据参数。

表 45 响应消息数据参数定义

定义
安全访问类型 此参数与请求消息的子功能参数的位 6~0 相同。
安全种子（高字节和低字节） 种子参数是服务端发送的数据值，客户端在计算访问安全性所需的密钥时将使用该值。如果发送请求消息是通过将子功能设置为请求服务端种子的值，则安全种子数据字节仅会在响应消息中存在。

5.5.4.4 支持的否定响应码（NRC_）

该服务应实施下列否定响应码。表 46 中记录了每个响应码发生的情况。如果错误情形适用于服务端，则应使用列出的否定响应。

表 46 支持的否定响应码

否定响应码	定义	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS
0x13	不正确消息格式或无效格式 如果消息的长度错误，则应发送此否定响应码。	IMLOIF
0x22	条件不正确 如果不满足请求安全访问的条件，则应返回此否定响应码。	CNC
0x24	请求顺序错误 如果在没有收到“请求种子”请求消息的情况下接收到“发送密钥”子功能，则发送此否定响应码。	RSE
0x31	请求超出范围 如果用户可选的安全访问数据记录包含无效数据，则应发送此否定响应码。	ROOR
0x35	无效密钥 如果接收到“发送密钥”子功能值，并且该密钥的值与服务端的内部存储/计算密钥不匹配，则发送此否定响应码。	IK
0x36	超出尝试次数 如果延迟计时器由于超出允许的错误访问尝试的最大次数而处于活动状态，则发送此否定响应码。	ENOAA
0x37	要求的时间延迟未失效 如果请求在延迟计时器处于活动状态时发送，则发送此否定响应码。	RTDNE

5.5.4.5 安全访问消息流示例

5.5.4.5.1 假设

对于下面给定的消息流示例，如果服务端处于“锁定”状态，则必须满足以下条件才能成功解锁服务端：

- 请求种子的子功能：0x01（请求种子）
- 发送密钥的子功能：0x02（发送密钥）
- 服务端的种子（2 字节）：0x3657
- 服务端的密钥（2 字节）：0xC9A9（例如，种子值的二补数）

客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误”（“0”）而请求获得响应消息。

5.5.4.5.2 示例 1 服务端处于“锁定”状态

5.5.4.5.2.1 步骤 1：请求种子

表 47 定义了安全访问请求消息流示例 1（步骤 1）。

表 47 —安全访问请求消息流示例 1（步骤 1）

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	安全访问请求服务标识符 SID	0x27	SA
#2	安全访问类型=请求种子，抑制肯定响应消息指示位=false	0x01	SAT_RSD

表 48 定义了安全访问肯定响应消息流示例 1（步骤 1）。

表 48—安全访问肯定响应消息流示例 1（步骤 1）

消息方向	服务端→客户端		
消息类型	响应		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	安全访问响应服务标识符 SID	0x67	SAPR
#2	安全访问类型=请求种子，	0x01	SAT_RSD
#3	安全种子 [字节#1] = 种子#1 (高字节)	0x36	SECHB
#4	安全种子 [字节#2] = 种子#2 (低字节)	0x57	SECLB

5.5.4.5.2.2 步骤 2：发送密钥

表 49 定义了安全访问请求消息流示例 1（步骤 2）。

表 49—安全访问请求消息流示例 1（步骤 2）

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	安全访问请求服务标识符 SID	0x27	SA
#2	安全访问类型=发送密钥，抑制肯定响应消息指示位=false	0x02	SAT_SK
#3	安全密钥[字节#1]=密钥#1 (高字节)	0xC9	SECKEY_HB
#4	安全密钥[字节#2]=密钥#2 (低字节)	0xA9	SECKEY_LB

表 50 定义了安全访问肯定响应消息流示例 1（步骤 1）。

表 50—安全访问肯定响应消息流示例 1（步骤 2）

消息方向	服务端→客户端		
消息类型	响应		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	安全访问响应服务标识符 SID	0x67	SAPR
#2	安全访问类型=发送密钥	0x02	SAT_SK

5.5.4.5.3 示例 2：服务端处于“解锁”状态

5.5.4.5.3.1 步骤 1：请求种子

表 51 定义了安全访问请求消息流示例 2（步骤 1）。

表 51 — 安全访问请求消息流示例 2（步骤 1）

消息方向		客户端→服务端		
消息类型		请求		
一个数据字节	描述（所有值均为十六进制）		字节值	助记符
#1	安全访问请求服务标识符 SID		0x27	SA
#2	安全访问类型=请求种子 抑制肯定响应消息指示位=false		0x01	SAT_RSD

表 52 定义了安全访问肯定响应消息流示例 2（步骤 2）。

表 52 — 安全访问肯定响应消息流示例 2（步骤 2）

消息方向		服务端→客户端		
消息类型		响应		
一个数据字节	描述（所有值均为十六进制）		字节值	助记符
#1	安全访问响应服务标识符 SID		0x67	SAPR
#2	安全访问模式=请求种子		0x01	SAT_RSD
#3	安全种子[字节#1]=种子#1（高字节）		0x00	SECHB
#4	安全种子[字节#2]=种子#2（低字节）		0x00	SECLB

5.5.5 通信控制（0x28）服务

5.5.5.1 服务描述

该服务的目的是开启/关闭服务端（例如应用程序 通信消息）的某些消息的发送和/或接收。

重要 —— 服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

5.5.5.2 请求消息

5.5.5.2.1 请求消息定义

表 53 定义了请求消息。

表 53 — 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记
#1	通信控制请求服务标识符 SID	M	0x28	CC
#2	子功能=[控制类型]	M	0x00 – 0xFF	LEV_CTRLTP
#3	通信类型	M	0x00 – 0xFF	CTP
#4	节点识别号（高字节）	Ca	0x00 – 0xFF	NIN
#5	节点识别号（低字节）	Ca	0x00 – 0xFF	NIN

a C 参数的存在要求控制类型为 0x04 或 0x05。

5.5.5.2.2 请求消息子功能参数\$Level (LEV_) 定义

子功能参数控制类型包含有关服务端应如何修改通信类型参数中引用的通信类型的信息（表 54 中未显示抑制肯定响应消息指示位（位 7））。

表 54 — 请求消息子功能参数定义

位 6-0	描述	Cvt	助记符
0x00	启用 Rx 和 Tx 此值表示应为指定的通信类型启用消息的接收和传输。	U	ERXTX
0x01	启用 Rx 和禁用 Tx 此值表示对于指定的通信类型，应启用消息的接收并禁用其传输。	U	ERXDTX
0x02	禁用 Rx 并启用 Tx 此值表示对于指定的通信类型，应启用消息的传输并禁用其接收。	U	DRXETX
0x03	禁用 Rx 和 Tx 此值表示对于指定的通信类型，应禁用消息的接收和传输。	U	DRXTX
0x04	启用 Rx 并使用增强的地址信息禁用 Tx 该值表示被寻址的总线主机应将相关的子总线段切换到仅支持诊断的调度模式。	U	ERXDTXWEAI
0x05	使用增强的寻址信息启用 Rx 和 Tx 该值表示被寻址的总线主机应将相关的子总线段切换到应用程序调度模式。	U	ERXTXWEAI
0x06 – 0x3F	ISOSAE 保留 此值范围由本文档保留以备将来定义。	M	ISOSAERESRVD
0x40 – 0x5F	车辆制造商特定 此值的范围保留位车辆制造商的特殊用途。	U	VMS

表 54 — (续)

位 6-0	描述	Cvt	助记符
0x60 – 0x7E	系统供应商特定 此值范围保留为系统供应商的特殊用途。	U	SSS
0x7F	ISOSAE 保留 此值由本文档保留以备将来定义。	M	ISOSAERESRVD

5.5.5.2.3 请求消息数据参数定义

表 55 定义了请求消息的数据参数。

表 55 — 请求消息数据参数定义

定义

通信类型
此参数用于引用要控制的通信类型。通信类型参数是中间码(Bit-code)值，它允许同时控制多种通信类型。（有关通信类型数据参数的编码，见附录 B.1）
节点识别号
此 2 字节参数用于识别车辆某处的子网络上的节点，该节点无法使用较低的 OSI 层（1 到 6）的寻址方法来寻址。只有当子功能参数控制类型被设置为 0x04 或 0x05（有关节点识别号数据参数的编码，见附录 B.4）时，此参数才会存在。

5.5.5.3 肯定响应消息

5.5.5.3.1 肯定响应消息定义

表 56 定义了肯定响应消息。

表 56 — 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	通信控制响应服务标识符 SID	M	0x68	CCPR
#2	子功能=[控制类型]	M	0x00 – 0x7F	LEV_CTRLTP

5.5.5.3.2 肯定响应消息数据参数定义

表 57 定义了肯定响应消息的数据参数。

表 57 — 响应消息数据参数定义

定义
控制类型
此参数与请求消息的子功能参数的位 6~0 相同。

5.5.5.4 支持的否定响应码 (NRC_)

该服务应实施下列否定响应码。表 58 中记录了每个响应码发生的情况。如果错误情形适用于服务端，则应使用列出的否定响应。

表 58 支持的否定响应码

否定响应码	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS
0x13	错误的消息长度或无效格式 如果消息的长度错误，则应发送此否定响应码。	IMLOIF
0x22	条件不正确 当服务端处于临界正常模式活动时使用，因此无法禁用/启用请求的通信类型。	CNC

0x31	请求超出范围 如果服务端检测到通信类型或节点识别号参数中有错误，则服务端应使用此响应码。	ROOR
------	---	------

5.5.5.5 通信控制消息流示例（禁用网络管理消息的传输）

客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误”（“0”）而请求获得响应消息。

表 59 定义了通信控制请求消息流示例。

表 59 — 通信控制请求消息流示例

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	通信控制请求服务标识符 SID	0x28	CC
#2	控制类型=启用 Rx 和禁用 Tx, 抑制肯定响应消息指示位=false	0x01	ERXTX
#3	通信类型=网络管理	0x02	NWMCP

表 60 定义了通信控制肯定响应消息流示例。

表 60 — 通信控制肯定响应消息流示例

消息方向	服务端→客户端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	通信控制响应服务标识符 SID	0x68	CCPR
#2	控制类型	0x01	CTRLTP

5.5.5.6 通信控制消息流示例（将远程网络切换到仅支持诊断的调度模式，该模式与地址为 0x000A 的节点相连接）

客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误”（“0”）请求获得响应消息。

表 61 定义了通信控制请求消息流示例。

表 61 — 通信控制请求消息流示例

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	通信控制请求服务标识符 SID	0x28	CC
#2	控制类型=按增强的地址信息启用 Rx 并禁用 Tx, 抑制肯定响应消息指示位=false	0x04	ERXTX
#3	通信类型=正常消息	0x01	NMCP
#4	节点识别号（高字节）	0x00	NIN
#5	节点识别号（低字节）	0x0A	NIN

表 62 定义了通信控制肯定响应消息流示例。

表 62 通信控制肯定响应消息流示例

消息方向		服务端→客户端	
消息类型		请求	
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	通信控制响应服务标识符 SID	0x68	CCPR
#2	控制类型=按增强的地址信息启用 Rx 并禁用 Tx, 抑制肯定响应消息指示位=false	0x04	ERXTX

5.5.5.7 通信控制消息流示例（切换到具有增强的地址信息的应用程序调度模式，连接到子网的节点 0x000A 被寻址）

客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误”（“0”）而请求获得响应消息。

表 63 定义了通信控制请求消息流示例。

表 63—通信控制请求消息流示例

消息方向		客户端→服务端	
消息类型		请求	
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	通信控制请求服务标识符 SID	0x28	CC
#2	控制类型=按增强的地址信息启用 Rx 和 Tx, 抑制肯定响应消息指示位=false	0x05	ERXTX
#3	通信类型=正常消息	0x01	NMCP
#4	节点识别号（高字节）	0x00	NIN
#5	节点识别号（低字节）	0x0A	NIN

表 64 定义了通信控制肯定响应消息流示例。

表 64—通信控制肯定响应消息流示例

消息方向		服务端→客户端	
消息类型		响应	
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	通信控制响应服务标识符 SID	0x68	CCPR
#2	控制类型=按增强的地址信息启用 Rx 和 Tx, 抑制肯定响应消息指示位= 错误	0x05	ERXTX

5.5.6 诊断设备在线（0x3e）服务

5.5.6.1 服务描述

该服务用于向单（或多）个服务端指示客户端仍然与车辆连接，并且先前已激活的某些诊断服务和/或通信将保持活动状态。

此服务用于将一个或多个服务端保留在默认会话之外的诊断会话中。这可以通过定期发送诊断设备在线请求消息来完成，也可

以在没有其他诊断服务的情况下完成，以防止服务端自动返回到默认会话。在保持单个服务端或多个服务端处于默认会话之外的诊断会话中时，适用于此服务的详细会话要求可以在 **ISO 14229 的实施规范中** 找到。

重要 —— 服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

5.5.6.2 请求消息

5.5.6.2.1 请求消息定义

表 65 定义了请求消息。

表 65 – 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	诊断设备在线请求服务标识符	M	0x3E	TP
#2	子功能=[零子功能]	M	0x00 / 0x80	LEV_ZSUBF

5.5.6.2.2 请求消息子功能参数\$Level (LEV_) 定义

表 66 指定了为该服务定义的子功能参数值（抑制肯定响应消息指示位（位 7）未显示）。

表 66 – 请求消息子功能参数定义

位 6-0	描述	Cvt	助记符
0x00	零子功能 此参数值表示此服务不支持除抑制肯定响应消息指示位之外的子功能值。	M	ZSUBF
0x01 – 0x7F	ISOSAE 保留 此值的范围由本文档保留。	M	ISOSAERESRVD

5.5.6.2.3 请求消息数据参数定义

该服务不支持请求消息中的数据参数。

5.5.6.3 肯定响应消息

5.5.6.3.1 肯定响应消息定义

表 67 定义了肯定响应消息。

表 67 – 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	诊断设备在线响应服务标识符 (SID)	M	0x7E	TPPR
#2	子功能=[零子功能]	M	0x00	LEV_ZSUBF

5.5.6.3.2 肯定响应消息数据参数定义

表 68 定义了肯定响应消息的数据参数。

表 68 — 响应消息数据参数定义

定义
零子功能 该参数与请求报的子功能参数的位 6-0 相同。

5.5.6.4 支持的否定响应码 (NRC_)

该服务应执行以下否定响应码。表 69 列出了每个响应码发生的情况。如果错误情形适用于服务端，则应使用列出的否定响应。

表 69 — 支持的否定响应码

否定响应码	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS
0x13	不正确的消息长度或无效格式 如果消息长度错误，则应发送否定响应码。	IMLOIF

5.5.6.5 诊断设备在线消息流示例

5.5.6.5.1 示例 1 — 诊断设备在线(抑制肯定响应消息指示位 = 错误)

表 70 定义了诊断设备在线.请求消息流示例 1。

表 70 — 诊断设备在线请求消息流示例 1

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	诊断设备在线请求服务标识符 SID	0x3E	TP
#2	零子功能, 抑制肯定响应消息指示位 = 错误	0x00	ZSUBF

表 71 定义了诊断设备在线肯定响应消息流示例 1。

表 71 — 诊断设备在线肯定响应消息流示例 1

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	诊断设备在线响应服务标识符 SID	0x7E	TPPR

#2	零子功能，抑制肯定响应消息指示位= 错误	0x00	ZSUBF
----	----------------------	------	-------

5.5.6.5.2 示例 2 — 诊断设备在线(抑制肯定响应消息指示位 = 肯定)

表 72 定义了诊断设备在线请求消息流示例 2。

表 72 — 诊断设备在线请求消息流示例 2

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	诊断设备在线请求服务标识符 SID	0x3E	TP
#2	零子功能，抑制肯定响应消息指示位 = 肯定	0x80	ZSUBF

服务端没有发送响应。

5.5.7 访问计时参数 (0x83) 服务

5.5.7.1 服务描述

访问计时参数服务用于读取和更改处于活动状态的通信链路的默认时间参数。

这种服务的使用很复杂，取决于服务端的能力和数据链路拓扑。每个诊断会话只支持一个扩展计时参数集。由于服务端支持不同的扩展计时参数集，建议仅将该服务用于物理寻址。

建议使用以下一系列服务：

- 诊断会话控制（诊断会话类型）服务；
- 访问计时参数（读取扩展计时参数集）服务；
- 访问计时参数（设置计时参数到给定值）服务；

对于服务端需要发送响应的情况，客户端和服务端应在服务端发送访问计时参数肯定响应消息之后激活新的计时参数设置。如果不允许响应消息，客户端和服务端应在发送/接收请求消息之后激活新的计时参数。

在成功切换到另一个或相同的诊断会话（例如，通过诊断会话控制、ECU 复位服务或会话计时超时）之后，服务端和客户端应将其计时参数重置为默认值。

访问计时参数服务提供了四种不同的访问服务端计时参数的模式：

- 读取扩展计时参数集；
- 将计时参数设置为默认值；
- 读取当前激活的计时参数；
- 将计时参数设置为给定值；

重要 —— 服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

5.5.7.2 请求消息

5.5.7.2.1 请求消息定义

表 73 定义了请求消息。

表 73 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	访问计时参数请求服务标识符 SID	M	0x83	ATP
#2	子功能=[计时参数访问类型]	M	0x00 – 0xFF	LEV_TPAT_
#3 : #n	计时参数请求记录 [字节#1 : 字节#m]	C : C	0x00 – 0xFF : 0x00 – 0xFF	TPREQR_ B1 : Bm
C: 计时参数请求记录仅在计时参数访问类型=将计时参数设置为给定值时才存在。计时参数请求记录的结构和内容依赖于数据链路层，因此在 ISO 14229 的实施规范中定义。				

5.5.7.2.2 请求消息子功能参数\$Level (LEV_) 定义

访问计时参数服务使用子功能“计时参数访问类型”来选择服务端的特定行为。可能的计时参数标识符的用法和说明详见表 74。其指定乐下列子功能值（抑制肯定响应消息指示位（位 7）未显示）：

表 74 请求消息子功能参数定义

位 6-0	描述	Cvt	助记符
0x00	ISOSAE 保留 此值由本文档保留。	M	ISOSAERESRVD
0x01	读取扩展计时参数集 当接收到计时参数访问类型=读取扩展计时参数集的访问计时参数指示原语时，服务端应读取扩展计时参数集，即服务端能够支持的值。 如果对计时参数集的读取访问成功，则服务端应发送具有肯定响应参数的访问计时参数响应原语。 如果对计时参数集的读取访问不成功，则服务端应发送一个否定响应消息和适当的否定响应码。 该子功能用于为当前活动的诊断会话提供一组额外的计时参数。 当计时参数访问类型=将计时参数设置到给定值时，只能设置该组计时参数（按计时参数访问类型读取=读取扩展计时参数集）。	U	RETPS

表 74 — (续)

位 6-0	描述	Cvt	助记符

0x02	将计时参数设置为默认值 在接收到计时参数访问类型=将计时参数设置为默认值的访问计时参数指示原语时，服务端应将所有计时参数更改为默认值，并在默认计时参数变为活动状态之前发送具有肯定响应参数的访问计时参数响应原语(如果将抑制肯定响应消息指示位设置为“False”，否则在成功确认请求消息之后，计时参数应将变为活动状态)。 如果由于某些原因计时参数不能被更改为默认值，则服务端应维持当前活动的计时参数，并发送具有适当否定响应码的否定响应消息。 默认计时值的定义取决于所使用的数据链路，并在 ISO 14229 的实施规范中进行了规定。	U	STPTDV
0x03	读取当前活动计时参数 在接收到计时参数访问类型=读取当前活动计时参数的访问计时参数指示原语时，服务端应读取当前使用的计时参数。 如果对计时参数的读取访问成功，则服务端应发送具有肯定响应参数的访问计时参数响应原语。 如果由于某些原因不能读取访问当前使用的计时参数，则服务端应发送具有适当否定响应码的否定响应消息。	U	RCATP
0x04	设置计时参数为给定值 当接收到计时参数访问类型=将计时参数设置为给定值的访问计时参数指示原语时，服务端应检查计时参数在当前条件下是否可以改变。 如果条件有效，则服务端应执行所有必要的操作以更改计时参数，并在新的计时参数值变为活动状态之前发送具有肯定响应参数的访问计时参数响应原语(将抑制肯定响应消息指示位设置为“False”，否则计时参数应在成功确认请求消息后变为活动状态)。 如果由于某些原因不能被改变，则服务端应维持当前活动的计时参数，并发送具有适当否定响应码的否定响应消息。 无法将服务端的计时参数设置为“通过计时参数访问类型=读取扩展计时参数集读取”的最小值和最大值之间的任意一组值。服务端的计时参数只能被设置为通过计时参数访问类型=读取扩展计时参数集读取的精确的计时参数。服务端应拒绝这样做的请求。	U	STPTGV
0x05 – 0xFF	ISOSAE 保留 此值由本文档保留以备将来定义。	M	ISOSAERESRVD

5.5.7.2.3 请求消息数据参数定义

表 75 定义了请求消息的数据参数。

表 75 — 请求消息数据参数定义

定义
<p>计时参数请求记录</p> <p>此参数记录包含在服务端中设置的通过计时参数访问类型=将计时参数设置为给定值的计时参数值。该参数记录的内容和结构是数据链路层特定的，并且可以在 ISO 14229 的实施规范（例如 ISO 14229 —3）中找到。</p>

5.5.7.3 肯定响应消息

5.5.7.3.1 肯定响应消息定义

表 76 定义了肯定响应消息。

表 76 — 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	访问计时参数响应服务标识符 SID	M	0xC3	ATPPR
#2	计时参数访问类型	M	0x00 – 0x7F	TPAT_
#3 : #n	计时参数响应记录[字节#1 : 字节#m]	C : C	0x00 – 0xFF : 0x00 – 0xFF	TPRSPR_ B1 : Bm

C: 计时参数响应记录仅在“计时参数访问类型=读取扩展计时参数集或读取当前活动计时参数”的情况下才存在。计时参数响应记录的结构和内容依赖于数据链路层，因而在 ISO 14229 的实施规范中被定义。

5.5.7.3.2 肯定响应消息数据参数定义

表 77 定义了肯定响应消息的数据参数。

表 77 — 响应消息数据参数定义

定义
<p>计时参数访问类型</p> <p>此参数与是请求消息的子功能参数的位 6~0 相同。</p>
<p>计时参数响应记录</p> <p>此参数记录包含“计时参数访问类型=读取扩展时序参数集或读取当前活动计时参数”时从服务端读取的计时参数值。该参数记录的内容和结构是数据链路层特定的，并且可以在 ISO 14229 的实施规范中找到。</p>

5.5.7.4 支持的否定响应码 (NRC_)

该服务应执行下列否定响应码。表 78 列出了每个响应码出现的情况。如果错误情景适用于服务端，则应使用列出的否定响应。

表 78 — 支持的否定响应码

否定响应码	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS
0x13	不正确的消息长度或无效格式 消息的长度或格式是错误的，则应发送此否定响应码。	IMLOIF
0x22	条件不正确 如果未满足请求访问计时参数的条件，则应返回该否定响应码。	CNC
0x31	请求超出范围 如果计时参数请求记录包含无效的计时参数值，则应发送此否定响应码。	ROOR

5.5.7.5 消息流示例访问计时参数

5.5.7.5.1 示例 1 — 将计时参数设置为默认值

此消息流显示了如何在服务端中设置默认计时参数。客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误”（“0”）来请求获得响应消息。

表 79 定义了访问计时参数请求消息流示例 1。

表 79 — 访问计时参数请求消息流示例 1

消息方向	客户端 <input checked="" type="checkbox"/> 服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	访问技术参数请求服务标识符 SID	0x83	ATP
#2	计时参数访问类型= 将计时参数设置为默认值； 抑制肯定响应消息指示位= 错误	0x02	TPAT_STPTDV

表 80 定义了访问计时参数肯定响应消息流示例 1。

表 80 — 访问计时参数肯定响应消息流示例 1

消息方向	服务端 <input checked="" type="checkbox"/> 客户端		
消息类型	响应		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	访问技术参数响应服务标识符 SID	0xC3	ATPPR
#2	计时参数访问类型= 将计时参数设置为默认值	0x02	TPAT_STPTDV

5.5.8 安全数据传输（0x84）服务

5.5.8.1 服务描述

5.5.8.1.1 目的

该服务的目的是传输防止第三方攻击而保护的数据，这可能危及数据安全。

如果客户打算在安全模式下使用本标准中定义的诊断服务，则安全数据传输服务适用。它也可以用于安全模式下在客户端和服务端之间传输符合其他应用协议的外部数据。在这种情况下的安全模式意味着传输的数据受到加密方法的保护。

5.5.8.1.2 安全子层

图 8 说明了安全子层。为了在安全模式下执行诊断服务，必须在服务端和客户端应用程序中添加安全子层。

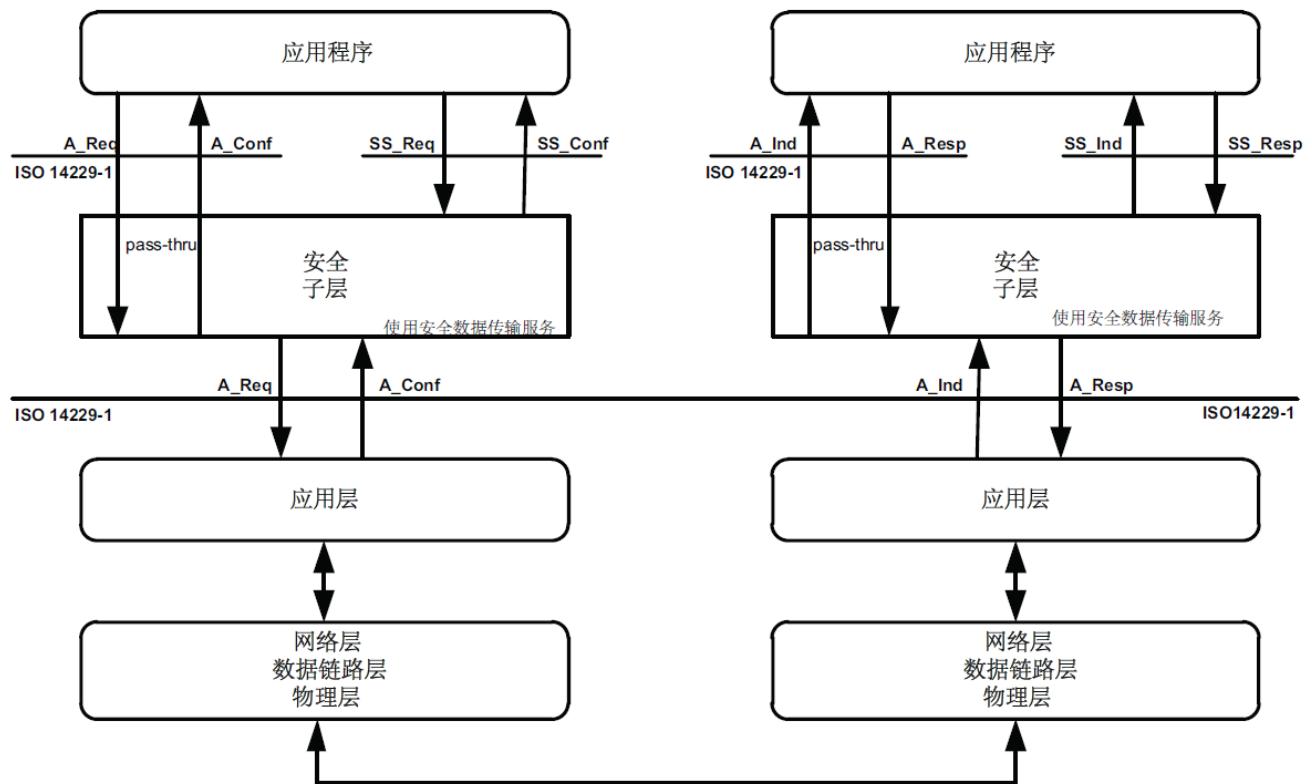


图 8 安全子层的实现

有两种方法可以在客户端和服务端之间执行诊断服务数据传输：

—— 非安全数据传输模式

该应用程序使用本标准中定义的诊断服务和应用层服务原语在客户端和服务端之间交换数据。安全子层在客户端和服务端的“应用程序”和“应用层”之间执行数据的“传递”。

—— 安全数据传输模式

应用程序使用诊断服务或外部服务以及安全子层服务原语在客户端和服务端之间交换数据。安全子层使用安全数据传输服务来发送/接收安全数据。安全链路必须是点对点通信。因此，只允许物理寻址，这意味着只能涉及一个服务端。

安全子层与应用程序的接口符合 ISO/OSI 模型惯例，因此提供以下四个安全子层（SS_）服务原语

- SS_安全模式.req: 安全子层请求

— SS_安全模式.ind:	安全子层指示
— SS_安全模式.resq:	安全子层响应
— SS_安全模式.conf:	安全子层确认

本标准定义了确认的和未确认的服务。在安全模式下，只允许确认的服务（抑制肯定响应消息指示位=False）。基于此要求，下列服务不允许在安全模式下执行：

- 响应事件 (0x86)
- 按周期标识符读取数据 (0x2A)
- 诊断设备在线 (0x3E);

已确认的服务（抑制肯定响应消息指示位=False）使用四个应用层服务原语请求、指示、响应和确认。在安全模式下执行已确认的诊断服务时，这些应用层服务原语被映射到四个安全子层服务原语，反之亦然。

在安全模式下执行诊断服务时，安全子层的任务是加密“应用程序”提供的数据，解密“应用层”提供的数据，并添加、检查和移除特定的安全数据元素。安全子层使用应用层的安全数据传输 (0x84) 服务根据外部协议（请求和响应）发送和接收整个诊断消息或消息，这些消息应在安全模式下交换。

安全子层为了安全执行诊断服务而向应用程序提供“安全服务执行”服务。

“安全服务执行”服务的安全子层请求和指示原语根据以下一般格式指定：

```
SS_SecuredMode.request      (
SA,
TA,
TA_type [,RA]
[,parameter 1, ...]
)
```

```
SS_SecuredMode.indication (
SA,
TA,
TA_type [,RA]
[,parameter 1, ...]
)
```

安全服务执行服务的安全子层响应和确认原语根据以下一般格式指定：

```
SS_SecuredMode.response    (
SA,
TA,
TA_type [,RA,]
Result
[,parameter 1, ...]
)
```

```
SS_SecuredMode.confirm     (
SA,
TA,
TA_type, [RA,]
Result
[,parameter 1, ...]
)
```

安全子层服务原语中的寻址信息直接映射到应用层的寻址信息上，反之亦然。

5.5.8.1.3 安全子层访问

访问用于安全服务执行的安全子层的概念与本标准中描述的应用层接口类似。安全子层使用应用层服务原语。

以下描述了在安全模式下执行已确认的诊断服务：

——客户端应用程序使用安全子层安全服务，请求以安全模式执行诊断服务。安全子层执行所需的操作以建立与服务端的链路，添加特定的安全相关参数，在需要时以安全模式加密要执行的诊断服务的服务数据，并使用应用层安全数据传输服务，请求将安全数据传输到服务端。

——服务端接收应用层安全数据传输服务指示，由服务端的安全子层处理。服务端的安全子层检查安全性特定的参数，解密加密数据并通过安全子层安全服务指示呈现要执行的服务的数据。应用程序执行服务并使用安全子层安全服务执行服务响应以安全模式响应服务。服务端的安全子层添加了特定的安全相关参数，根据需要加密响应消息数据，并使用应用层安全数据传输服务响应将响应数据传输给客户端。

——客户端接收应用层安全数据传输服务确认原语，该原语由客户端的安全子层处理。客户端的安全子层检查安全性特定的参数，解密加密的响应数据，并通过安全子层安全服务执行确认将数据呈现给应用程序。

图9用图形展示了在安全模式下执行确认的诊断服务时安全子层、应用层和应用程序的交互。

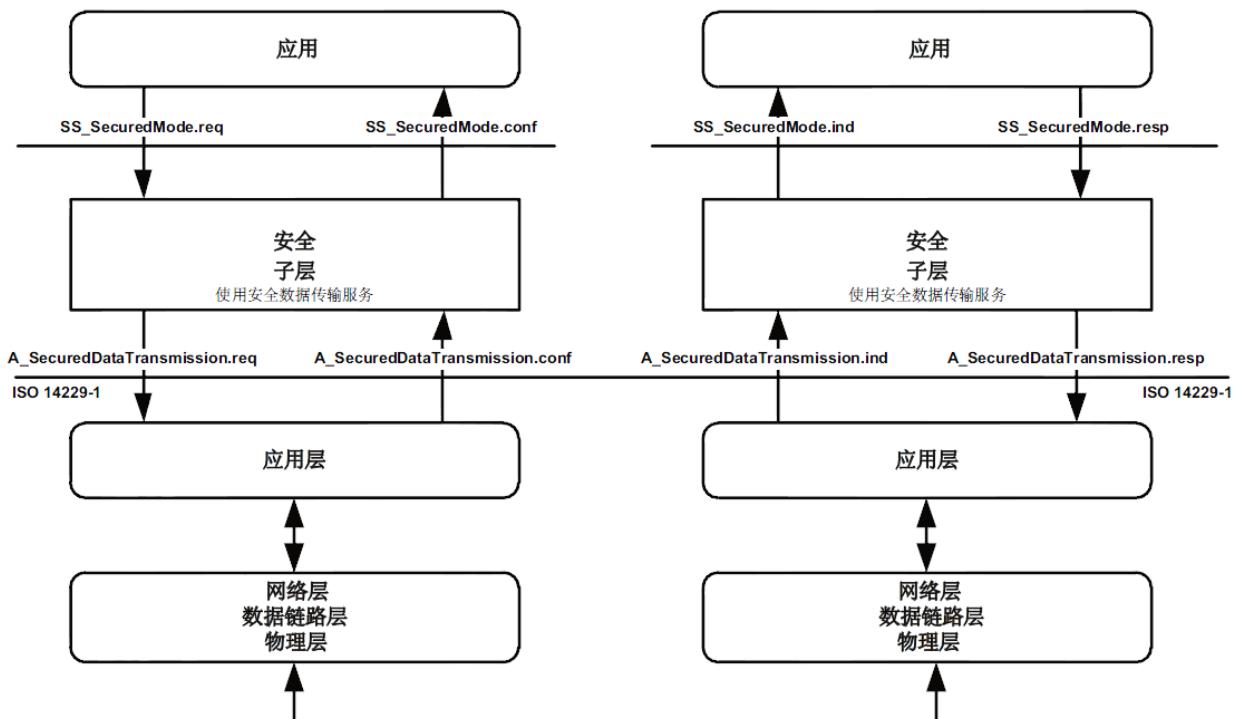


图9—安全子层、应用层和应用程序的交互

重要 —— 服务端和客户端应符合 7.5 中规定的请求和响应消息行为。

5.5.8.2 请求消息

5.5.8.2.1 请求消息定义

安全子层生成应用层安全数据传输请求消息参数。

表 81 定义了请求消息。

表 81 — 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	安全数据传输请求服务标识符 SID	M	0x84	SDT
#2 ： #n	安全数据请求记录[]=[安全数据参数#1 安全数据参数#m]	M ：M	0x00 – 0xFF ： 0x00 – 0xFF	SECDRQR_ SDP_ ：

5.5.8.2.2 请求消息子功能参数\$Level (LEV_) 定义

此服务不使用子功能参数。

5.5.8.2.3 请求消息数据参数定义

表 82 定义了请求消息的数据参数。

表 82 — 请求消息数据参数定义

定义
安全数据请求记录 此参数包含安全子层处理的数据。

5.5.8.3 肯定响应消息

5.5.8.3.1 肯定响应消息定义

表 83 定义了肯定响应消息。

表 83 — 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
1	安全数据传输响应服务标识符 SID	M	0xC4	SDTPR
2 ： n	安全数据响应记录[]=[安全数据参数#1 安全数据参数#m]	M ：M	0x00 – 0xFF ： 0x00 – 0xFF	SECDRQR_ SDP_ ：

5.5.8.3.2 肯定响应消息数据参数定义

表 84 定义了肯定响应消息的数据参数:

表 84 — 响应消息数据参数定义

定义
安全数据响应记录 此参数包含安全子层处理的数据。

5.5.8.4 支持的否定响应码(NRC_)

该服务应执行以下否定响应码。表 85 记录了每个响应码发生的情况。响应码总是在没有加密的情况下发送，即使根据请求应用层协议数据单元 (A_PDU) 中的配置文件，响应 A_PDU 必须加密。如果错误情景适用于服务端，则应使用列出的否定响应。

表 85 — 支持的否定响应码

否定响应码	描述	助记符
0x13	不正确的消息长度或无效格式 如果请求应用层协议数据单元 (A_PDU) 的长度不正确，服务端应使用此响应码。	IMLOIF
0x38 – 0x4F	按扩展数据链路安全性文档进行保留 扩展数据链路安全性保留了此值的范围。	RBEDLSD

注释：上面列出的响应码适用于安全数据传输 (0x84) 服务。如果以安全模式执行的诊断服务需要否定响应，则该否定响应通过安全数据传输肯定响应消息以安全模式发送到客户端。

5.5.9 控制诊断故障码(DTC)设置 (0x85) 服务

5.5.9.1 服务描述

控制诊断故障码 DTC 设置服务应由客户端用来停止或恢复服务端中 DTC 状态位的更新。在读取 DTC 信息的特定子功能的肯定响应中，DTC 的状态位在 DTC 参数状态中进行报告（参见 D.2 以了解位的定义）。

控制 DTC 设置请求消息可用于停止更新单个服务端或一组服务端中 DTC 状态位。如果被寻址的服务端不能停止 DTC 状态位的更新，则应给出控制 DTC 设置否定响应消息表明否定的原因。

当服务端接收到控制 DTC 设置请求且该请求中子功能参数“DTC 设置类型=关闭”时，服务端应暂停对 DTC 状态位的更新（即冻结当前值），直到功能再次启用。在控制 DTC 设置请求被执行并且子功能被设置为“开启”或转换到不支持控制 DTC 设置的会话时，DTC 状态位信息的更新应该继续（例如，会话层超时到默认会话、ECU 复位等）。即使请求的 DTC 设置状态已经激活，如果服务在活动会话中被支持并且所请求的子功能设置为“开启”或“关闭”，服务端仍应发送肯定响应。

如果客户端发送清除诊断信息 (0x14) 服务，则控制 DTC 设置不应禁止重置服务端的 DTC 状态位。单个 DTC 状态位的行为应根据 D.2, D.1—图 D.8 中的定义执行。

DTC 状态位记录了与代表特定故障状况的数字标识符 (DTC) 相关的某些信息。控制 DTC 设置仅打开/关闭 DTC 状态位的更新。控制 DTC 设置服务的目的不是使故障监控被关闭或可靠性功能降级策略被禁用。不建议将故障弱化或故障安全策略与 DTC 状态位直接关联或耦合（例如，接收的清除诊断信息请求不会直接删除任何活动状态的可靠性功能降级）。

重要 —— 服务端和客户端应符合 7.5 中规定的请求和响应消息行为。

5.5.9.2 请求消息

5.5.9.2.1 请求消息定义

表 86 定义了请求消息。

表 86 — 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	控制 DTC 设置请求服务标识符 SID	M	0x85	CDTCS
#2	子功能=[DTC 设置类型]	M	0x00 – 0xFF	LEV_DTCSTP_
#3 : #n	DTC 设置控制选项记录[] =[参数#1 参数#m]	U : U	0x00 – 0xFF : 0x00 – 0xFF	DTCSCOR_ PARA1 : PARAm

5.5.9.2.2 请求消息子功能参数\$Level (LEV_) 定义

控制 DTC 设置请求消息使用子功能参数 DTC 设置类型, 来向服务端指示诊断故障码状态位的更新是应该停止还是重新启动(表 87 中未显示抑制肯定响应消息指示位(位 7))。

表 87 — 请求消息子功能参数定义

位 6 – 0	描述	Cvt	助记符
0x00	ISOSAE 保留 此值由本文档保留。	M	ISOSAERESRVD
0x01	开启 根据正常的操作条件, 服务端应恢复更新诊断故障码的状态位。	M	ON
0x02	关闭 服务端应停止更新诊断故障码的状态位。	M	OFF
0x03 – 0x3F	ISOSAE 保留 此值的范围由本文件保留以备将来定义。	M	ISOSAERESRVD
0x40 – 0x5F	车辆制造商特定 此值的范围保留为车辆制造商的特定用途。	U	VMS
0x60 – 0x7E	系统供应商特定 此值的范围保留为系统供应商的特定用途。	U	SSS
0x7F	ISOSAE 保留 此值由本文档保留以备将来定义。	M	ISOSAERESRVD

5.5.9.2.3 请求消息数据参数定义

表 88 定义了请求消息的数据参数。

表 88 — 请求消息数据参数定义

描述

DTC 设置控制选项记录

当控制 DTC 状态位的更新时，该参数记录是用户可选的，以便向服务端传送数据（例如，它可以包含要打开或关闭的 DTC 列表）。

5.5.9.3 肯定响应消息

5.5.9.3.1 肯定响应消息定义

表 89 定义了肯定响应消息。

表 89 — 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	控制 DTC 设置响应服务标识符 SID	M	0xC5	CDTCSR
#2	DTC 设置类型	M	00–7F	DTCSTP

5.5.9.3.2 肯定响应消息数据参数定义

表 90 定义了肯定响应消息的数据参数。

表 90 — 响应消息数据参数定义

定义
DTC 设置类型
该参数是请求消息的子功能参数的位 6-0 的回显。

5.5.9.4 支持的否定响应码 (NRC_)

该服务应执行以下否定响应码。 表 91 中记录了每个响应码发生的情况。如果错误情形适用于服务端，则应使用列出的否定响应。

表 91 — 支持的否定响应码

否定响应码	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS
0x13	不正确的消息长度或无效格式 如果消息长度错误，则应发送否定响应码。	IMLOIF
0x22	条件不正确 在服务端处于关键正常模式活动时使用，因而无法执行请求的 DTC 控制功能。	CNC

表 91 — (续)

否定响应码	描述	助记符

0x31	请求超出范围 如果服务端检测到 DTC 设置控制选项记录中有错误，则应使用此响应码。	ROOR
------	---	------

5.5.9.5 消息流示例控制 DTC 设置

5.5.9.5.1 示例 1 — 控制 DTC 设置 (DTC 设置类型=关闭)

请注意，该示例不使用此服务将附加数据传输到服务端的能力。客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误” ('0')，以请求获得响应消息。

表 92 定义了控制 DTC 设置请求消息流示例 1。

表 92 — 控制 DTC 设置请求消息流示例 1

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	控制 DTC 设置请求服务标识符 SID	0x85	RDTCS
#2	DTC 设置类型=关闭，抑制肯定响应消息指示位= 错误	0x02	DTCSTP_OFF

表 93 定义了控制 DTC 设置的肯定响应消息流示例 1。

表 93 — 控制 DTC 设置肯定响应消息流示例 1

消息方向	服务端→客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	控制 DCT 设置响应服务标识符 SID	0xC5	RDTCSPR
#2	DTC 设置类型=关闭	0x02	DTCSTP_OFF

5.5.9.5.2 示例 2 — 控制 DTC 设置 (DTC 设置类型= 开启)

该示例不使用此服务将附加数据传输到服务端的能力。客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误” ("0")，以请求获得响应消息。

表 94 定义了控制 DTC 设置请求消息流示例 2。

表 94 — 控制 DTC 设置请求消息流示例 2

消息方向	客户端→服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	DTC 控制设置请求服务标识符 SID	0x85	ENC
#2	DTC 设置类型=开启，抑制肯定响应消息指示位=false	0x01	DTCSTP_ON

表 95 定义了控制 DTC 设置肯定响应消息流示例 2。

表 95 — 控制 DTC 设置肯定响应消息流示例 2

消息方向		服务端→客户端	
消息类型		响应	
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	控制 DTC 设置响应服务标识符 SID	0xC5	RDTCSPR
#2	DTC 设置类型=开启	0x01	DTCSTP_ON

5.5.10 响应事件（0x86）服务

5.5.10.1 服务描述

响应事件服务（ROE）请求服务端启动或停止传输对指定事件的响应。

该服务提供了在服务端中发生指定事件的情况下自动执行诊断服务的可能性。客户端指定事件（包括可选事件参数）以及事件发生时要执行的服务（包括服务参数）。参见图 10 以简要了解客户端和服务端行为。

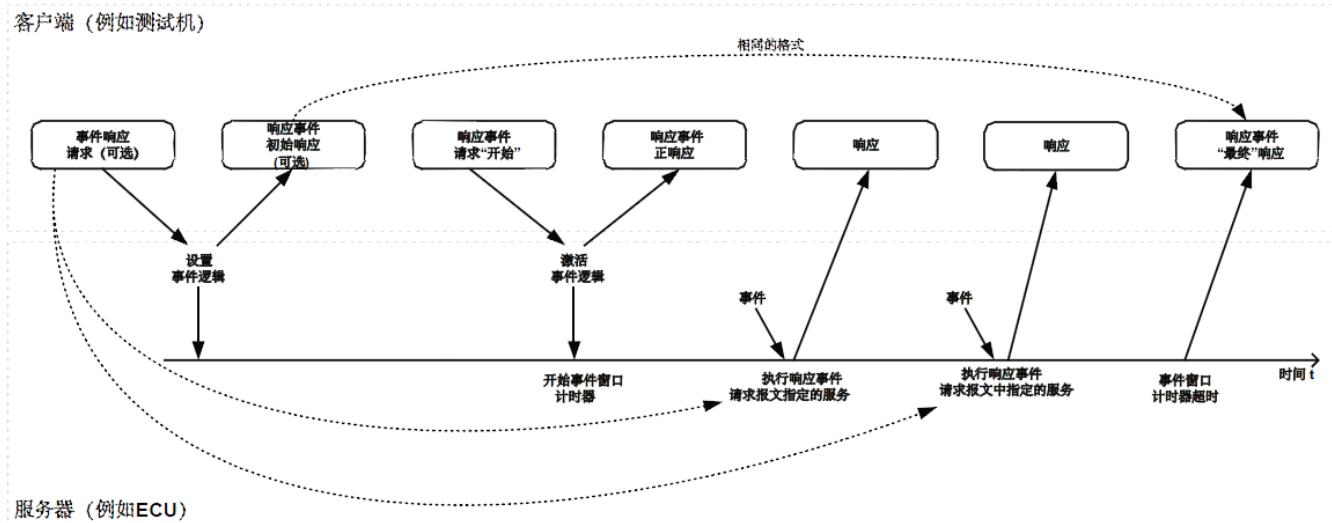


图 10 — 响应事件服务 — 客户端和服务端行为

注意上图假设事件窗口计时器配置为在服务端断电之前超时，因此事件计时窗口结束时会显示最终响应事件肯定响应消息。

服务端应在接收时对响应事件请求消息的子功能和数据内容进行评估。这包括以下子功能和参数：

- 事件类型，
- 事件窗口时间，
- 事件类型记录（事件类型参数#1—#m）。

在响应事件请求消息数据无效的情况下，应发送带有否定响应码 0x31 的否定响应。响应记录服务不是本**评估**的一部分。当指定的事件发生时，响应记录参数服务将被评估，这会触发执行包含在响应记录服务中的服务。在事件发生时，应执行响应记录服务（诊断服务请求消息）。在条件不正确的情况下，应发送带有适当的否定响应码的否定响应消息。应按其发生顺序发出多个事件的信号。

以下列实施细则适用：

- a) 响应事件服务可以在任何会话中设置和激活，包括默认会话。响应事件服务处于活动状态不需要诊断设备在线服务。
- b) 如果在诊断服务正在进行时发生指定的事件，这意味着要么正在接收请求消息，要么正在执行请求，或者正在进行响应消息（这包括否定响应消息处理响应码 0x78）（如果抑制肯定响应消息指示位=false），那么包含在响应记录服务中的请求消息的执行应被推迟，直到完成正在进行的诊断服务为止。
- c) 注意在某些情况下，由于延迟响应服务而导致响应记录服务中包含的数据可能与导致事件的数据值不一致。
- d) 如果发生多个事件时，另一个事件正在进行，那么对于多个事件的处理（例如，忽略除第一个/最后一个事件以外的所有事件或堆积事件）应由车辆制造商指定。
- e) 当事件逻辑得到满足并且事件在事件时间窗口内生成时，服务端应执行包含在响应记录服务中的服务。
- f) 切换到任何非默认会话的诊断会话控制请求应停止响应事件服务，不管激活的会话与当前会话是否相同。在返回默认会话时，所有先前在默认会话中处于活动状态的响应事件服务应重新激活。
- g) 对事件服务的多重响应可以与不同的需求（不同的事件类型，服务对响应记录...）同时运行以启动和停止诊断服务。启动和停止子功能应始终控制所有初始化的响应事件服务。
- h) 如果设置了响应事件服务，则应适用以下内容：

1) 如果将事件类型子功能参数的位 6 设置为 0 (不存储事件)，则服务端断电时事件应结束。在复位或通电后不应继续响应事件诊断服务 (即响应事件服务结束)。

2) 如果事件类型子功能参数的位 6 设置为 1 (存储事件)，则应根据服务端重新启动后的响应事件设置，恢复发送响应回复服务。因此，存储事件只能与无限事件窗口时间结合使用。

i) 客户端只应将“抑制肯定响应消息指示位”=“YES”用于事件类型=停止响应事件、启动响应事件或清除响应事件这些情形。当检测到指定事件时，服务端应始终对事件触发响应进行返回响应。

j) 服务端只有在设置了有限的窗口时间且有限的窗口时间已过时，才会返回响应事件的“最终”响应 (参见图 15) 以指示响应事件 (0x86) 服务。如果在有限窗口时间过后响应事件 ROE 已被任何方式停止 (例如“停止响应事件”子功能或会话更改)，则不会发送最终响应。

k) 为了避免干扰正常的诊断操作，建议只对瞬态事件和条件实施响应事件服务。每发生一次事件，服务端都会返回一次响应。对于持续一段时间的情况，响应服务应只在最初发生时执行一次。如果事件类型被定义为使响应回复服务得以高频率发生，则必须采取适当的措施，以防止连续的响应回复服务。响应回复服务之间的最小间隔时间可以是事件类型记录 (车辆制造商特定) 的一部分。

建议仅使用表 96 中列出的服务，以便在发生指定事件时执行服务。(响应记录服务请求服务标识符)。

表 96 — 与响应事件服务一起使用的推荐服务

推荐服务 (响应服务)	服务标识符请求	响应标识符
按标识符读取数据	0x22	0x62
读取 DTC 信息	0x19	0x59
例程控制	0x31	0x71
按标识符进行输入输出控制	0x2F	0x6F

出于性能原因 (例如，避免错过执行服务对于响应记录请求服务标识符)，建议遵守以下建议：

—— 数据标识符 DID 可能包含可测量的数据 (例如，避免定义事件逻辑读取常量“校准”标签)

—— 响应记录服务记录正面响应的大小可能受限于车辆制造商的特定值

重要 —— 服务端和客户端应符合 7.5 中规定的请求和响应消息行为。

5.5.10.2 请求消息

5.5.10.2.1 请求消息定义

表 97 定义了请求消息。

表 97 — 请求消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	响应事件请求服务标识符 SID	M	0x86	ROE
#2	子功能=[事件类型]	M	0x00 – 0xFF	LEV_ETP
#3	事件窗口时间	M	0x00 – 0xFF	EWT
#4 : #(m-1)+4	事件类型记录[]=[事件类型参数 1 : 事件类型参数 m]	C1 : C1	0x00 – 0xFF : 0x00 – 0xFF	ETR_ ETP1 : ETPm

#n-(r-1)- 1 #n-(r-1) : #n	响应记录服务[] = [服务标识符 服务端参数 1 : 服务端参数 r]	C2 C3 : C3	0x00 – 0xFF 0x00 – 0xFF : 0x00 – 0xFF	STRTR_ SI SP1 : SPr
C1: 如果事件类型需要指定响应事件的附加参数，则存在。				
C2: 如果子功能参数不等于报告激活事件、停止响应事件、启动响应事件、清除响应事件，则必须存在。				
C3: 如果响应服务的服务请求需要附加服务参数，则存在。				

5.5.10.2.2 请求消息子功能参数\$Level (LEV_) 定义

5.5.10.2.2.1 响应事件请求消息子功能参数定义

响应事件请求消息使用子功能参数事件类型，以指定要在服务端中配置的事件，并控制设置的事件响应。表 98 中给出的每个子功能参数值也指定了适用的事件类型记录的长度（下表中未显示抑制肯定响应消息指示位（位 7））。

事件类型子功能参数的位 6 用于表示事件是应该存储在服务端的非易失性存储器中，并在下次启动服务端时重新激活，还是应该在服务端断电的情况下立即终止（存储状态参数）。

表 98 — 事件类型子功能位 6 定义 — 存储状态

位 6 值	描述	Cvt	助记符
0x00	不存储事件 此值表示当服务端断电时，该事件应终止，且服务端在重置或接通电源后不应继续响应事件诊断服务（即响应事件服务终止）。	M	DNSE
0x01	存储事件 此值表示 1) 在默认会话中的响应事件 ROE 开始或停止请求中，事件应根据重置或开机后设置的响应事件进行恢复或停止发送响应回复服务（即恢复响应事件服务）。 2) 在任何响应事件 ROE 设置的事件逻辑请求中，该请求的事件逻辑应持久存储，直到事件逻辑被明确清除（通过清除响应事件）或事件逻辑被相同类别的新响应事件 ROE 设置的事件逻辑请求所覆盖。	U	SE

表 99 定义了请求消息子功能参数。

表 99 — 请求消息子功能参数定义

位 5 — 0 值	描述	Cvt	响应事件子功能类型	助记符
0x00	停止响应事件 用于停止服务端发送事件响应。已设置的事件逻辑不会被清除，但可以通过启动响应事件子功能参数来重新开始。 事件类型记录的长度：0 字节	U	control	STPROE

0x01	<p>关于 DTC 状态变更</p> <p>将该事件认定为检测到的新 DTC，此新 DTC 与该事件指定的 DTC 状态掩码相匹配。</p> <p>事件类型记录的长度：1 个字节</p> <p>实施提示：服务端常用 DTC 计数算法应以特定的周期率（例如约 1 秒）对满足客户端定义的 DTC 状态掩码的 DTC 数量进行计数。如果此计数与先前执行的计数不同，则客户端应生成导致响应服务执行的事件。最后的计数应作为下次计算的参考。</p> <p>此事件类型要求在请求消息中指定 DTC 状态掩码（事件类型参数#1）。</p>	U	setup	ONDTCs
0x02	<p>关于计时器中断</p> <p>将此事件认定为计时器中断，但计时器及其值不属于响应事件服务的一部分。</p> <p>此事件类型需要在请求消息（事件类型记录）中详细说明。</p> <p>事件类型记录长度：1 字节</p>	U	setup	OTI

表 99 — (续)

位 5 — 0 值	描述	Cvt	响应事件子功能的类型	助记符
0x03	<p>关于数据标识符的变更</p> <p>将此事件认定为由数据标识符鉴别的新内部数据记录。数据值由车辆制造商指定。</p> <p>此事件类型要求在请求消息（事件类型记录）详细说明。</p> <p>事件类型记录的长度：2 字节</p>	U	设置	OCODID
0x04	<p>报告激活事件</p> <p>用于表示在肯定响应中，所有事件均已报告，并且已通过响应事件服务在服务端中激活（且当前处于活动状态）。</p> <p>事件类型记录的长度：0 字节</p>	U	控制	RAE
0x05	<p>启动响应事件</p> <p>用于表示服务端激活已设置的事件逻辑（包括事件窗口计时器）并开始发送响应事件。</p> <p>事件类型记录的长度：0 字节。</p>	M	控制	STRTROE
0x06	<p>清除响应事件</p> <p>用于清除已在服务端中设置的事件逻辑（这也会停止服务端发送响应事件。）</p> <p>事件类型记录的长度：0 字节。</p>	U	控制	CLRROE
0x07	<p>关于值的比较</p> <p>从数据标识符鉴别的特定记录中定义数据值的变化，该数据标识符认定数据值事件。通过这个子功能，用户可以在从限定的测量值比较中收集到的特定结果发生时定义一个事件。将包含在分配给限定的数据标识符的数据记录中的特定测量值与给定的比较值进行比较。指定的运算符定义了比较的类型。如果比较结果为肯定，则事件发生。</p> <p>事件类型记录的长度：10 个字节</p>	U	设置	OCOV

0x08 – 0x1F	ISOSAE 保留 此值的范围由本文件保留以备将来定义。	M	—	ISOSAERES RVD
0x20 – 0x2F	车辆制造商特定 此值的范围保留为车辆制造商的特定用途。	U	设置	VMS
0x30 – 0x3E	系统供应商特定 此值的范围保留为系统供应商的特定用途。	U	设置	SSS
0x3F	ISOSAE 保留 此值由本文档保留以备将来定义。	M	—	ISOSAERES RVD

5.5.10.2.2.3 计时器中断参数规范中详细的请求消息子功能

通过计时器中断的子功能，服务端可以在计时器可配置的时间段内触发事件。

事件类型记录使用以下计时器进度表定义计时器的值：

- 慢速率，
- 中速率，
- 快速率.

制造商的具体任务是规定与每个计时器进度表选项相关联的时间速率。见表 100。

表 100 — 比较逻辑参数定义

事件类型记录	事件生成	计时器类型
0x01	每当达到慢速率计时器的值时。	慢速率计时器，例如，计时器在 1 秒后 经过
0x02	每次中速率计时器的值经过。	中速率计时器，例如，计时器在 300 毫秒后 经过
0x03	每次快速率计时器的值经过。	快速率计时器，例如，计时器在 25 毫秒后 经过

5.5.10.2.2.4 关于更改数据标识符参数规范的详细请求消息子功能

通过关于更改数据标识符的子功能，服务端被允许在固定的时间段内对测量结果进行轮询并比较其内容，因而如果服务端丢失某些更改且触发比预期少的事件，这也是可以接受的。

事件类型记录设置了必须对任何更改进行监视的双字节数据标识符 DID 值。

5.5.10.2.2.5 数值比较参数规范中的详细请求消息子功能

在响应记录服务指定读取数据标识符 DIDs 间的比较的情况下，表 101—表 103 规定了请求消息的参数，该消息带有关于数值比较参数的子功能。

表 101 — 数值比较参数定义的子功能

数据字节	参数名称	字节值	备注	详细要求
1	服务标识符	0x86	请求服务标识符 SID	----
2	事件类型	0x07	关于数值比较的子功能	----
3	事件窗口时间	0x02	无限时间窗口规范	----
4	事件类型记录字节 1	0x01	数据标识符 (DID) 高字节	可以是与用于响应记录服务中使用的数据标识符 DID 不同的 DID。可以是动态定义的 DID。

5	事件类型记录字节 2	0x04	数据标识符 (DID) 低字节	----
---	------------	------	-----------------	------

表 101 — (续)

数据字节	参数名称	字节值	备注	详细要求
6	事件类型记录字节 3	0x01	比较逻辑, 见表 102	事件类型记录字节 3 设置比较方法的逻辑。
7	事件类型记录字节 4	0x00	原始参考比较值最高有效位 MSB	事件类型记录字节 4, 5, 6, 7 设置参考比较值。
8	事件类型记录字节 5	0x00	原始参考比较值	----
9	事件类型记录字节 6	0x01	原始参考比较值	----
10	事件类型记录字节 7	0x00	原始参考比较值最低有效位 LSB	----
11	事件类型记录字节 8	0x0A	滞后值	事件类型记录字节 8 定义了从 0% (0x00) 到 100% (0x64) 的滞后值百分比。
12	事件类型记录字节 9	0xBC	本地化字节 1 最高有效位 MSB, 见表 103	事件类型记录字节 9,10 定义了数据标识符中的数值的本地化, 参见表 103。
13	事件类型记录字节 10	0x58	本地化字节 2 最低有效位 LSB, 见表 103	----
14	响应服务字节 1	0x22	服务标识符 SID	响应记录服务设置服务和对数据标识符 DID 进行读取和比较。在第一个肯定响应消息中, 识别事件的数量字段始终设置为 0x00。
15	响应服务字节 2	0xA1	数据标识符 DID1	----
16	响应服务字节 3	0x00	数据标识符 DID2	----

表 102 定义了比较逻辑参数定义。

表 102 — 比较逻辑参数定义

比较逻辑 ID	当以下情形发生时, 事件生成
0x01	比较参数<测量值
0x02	比较参数>测量值
0x03	比较参数=测量值
0x04	比较参数<>测量值

表 103 定义了数值 16 位位域的本地化参数定义。

表 103 – 数值 16 位位域的本地化参数定义

位字段的位置	描述
15	(最高有效位 MSB)，位=0 表示无符号比较，位=1 表示符号比较
14 – 10	位#10（最低有效位 LSB）— 位#14（最高有效位 MSB）包含要比较的数据标识符值的长度。数值 0x00 应用于比较所有 4 个字节。所有其他值应以位为单位设置大小。如果有 5 位，长度最大为 31 位。
9 – 0	肯定响应消息的偏移量，从肯定响应消息中提取数据标识符值。位#0（最低有效位 LSB）— 位#9（最高有效位 MSB）包含起始位编号偏移量。如果有 10 位，偏移量最大为 1023 位。

5.5.10.2.3 请求消息数据参数定义

表 104 定义了请求消息的数据参数。

表 104 — 请求消息数据参数定义

定义
事件窗口时间 参数事件窗口时间用于指定事件逻辑在服务端中处于活动状态的窗口。如果事件窗口时间的参数值设置为 0x02，则响应时间是无限的。如果存在无限的事件窗口并且存储状态等于不存储事件，建议通过某个信号关闭事件窗口（例如，关闭电源）。有关指定的事件窗口时间，见 B.2。 不应使用有限事件窗口和存储状态等于存储事件的组合。 注：如果事件类型等于响应事件 ROE 控制的子功能，则服务端不需考虑该参数。
事件类型记录 此参数记录包含指定事件类型的附加参数。
响应记录服务 此参数记录包含每次在事件类型记录中明确的指定事件发生时要在服务端中执行的服务的服务参数（服务标识符和服务参数）

5.5.10.3 肯定响应消息

5.5.10.3.1 肯定响应消息的定义

表 105 定义了所有除报告激活事件以外的子功能的肯定响应消息。

表 105 — 除报告激活事件以外所有子功能的肯定响应消息的定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	响应事件响应服务标识符 SID	M	0xC6	ROEPR
#2	事件类型	M	0x00 – 0x7F	ETP
#3	识别事件的数量	M	0x00 – 0xFF	NOIE
#4	事件窗口时间	M	0x00 – 0xFF	EWT

表 105 — (续)

一个数据字节	参数名称	Cvt	字节值	助记符
#5 : #(m-1)+5	事件类型记录 [] = [事件类型参数 1 : 事件类型参数 m]	C1 : C1	0x00 – 0xFF : 0x00 – 0xFF	ETR_ ETP1 : ETPm
#n-(r-1)-1 #n-(r-1) :	响应记录服 务 [] = [服务标识符 服务参数 1 : 服务参数 r]	M C2 : C2	0x00 – 0xFF 0x00 – 0xFF : 0x00 – 0xFF	STRTR_ SI SP1 : SPr
C1: 如果事件类型需要指定事件响应的附加参数，则存在。				
C2: 如果响应服务的服务请求需要附加服务参数，则存在。				

表 106 — 定义了子功能=报告激活事件时的肯定响应消息。

表 106 — 子功能=报告激活事件时的肯定响应消息的定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	响应事件响应服务标识符 SID	M	0xC6	ROEPR
#2	事件类型 = 报告激活事件	M	0x04	ETP_RAE
#3	活动事件的数量	M	0x00 – 0xFF	NOIE
#4	活动事件的事件类型#1	C1	0x00 – 0xFF	EVOAE
#5	事件窗口时间#1	C1	0x00 – 0xFF	EWT
#6 : #(m-1)+6	事件类型记录#1[] = [事件类型参数 1 : 事件类型参数 m]	C2 : C2	0x00 – 0xFF : 0x00 – 0xFF	ETR_ ETP1 : ETPm
#p-(o-1)-1 #p-(o-1) : #p	响应记录服务#1[] = [服务标识符 服务参数 1 : 服务参数 o]	C3 C4 : C4	0x00 – 0xFF 0x00 – 0xFF : 0x00 – 0xFF	STRTR_ SI SP1 : SPo
:	:	:	:	:
#n-(r-1)-4-(q)	活动事件的事件类型#k	C1	0x00 – 0xFF	EVOAE
#n-(r-1)-3-(q)	事件窗口时间#k	C1	0x00 – 0xFF	EWT
#n-(r-1)-2-(q-1) : #n	事件类型记录#k[] = [事件类型参数 1 : 事件类型参数 q]	C2 : C2	0x00 – 0xFF : 0x00 – 0xFF	ETR_ ETP1 : ETPm
#n-(r-1)-1 #n-(r-1) : #n	响应记录服务#k[] = [服务标识符 服务参数 1 : 服务参数 r]	C3 C4 : C4	0x00 – 0xFF 0x00 – 0xFF : 0x00 – 0xFF	STRTR_ SI SP1 : SPr
C1: 如果报告活动事件，则存在。				
C2: 如果所报告的活动事件的事件类型（活动事件的事件类型）需要指定事件响应的附加参数，则存在。				
C3: 报告活动事件时必须存在。				
C4: 如果所报告的服务请求响应需要额外的服务参数，则存在。				

5.5.10.3.2 肯定回复消息数据参数定义

表 107 定义了肯定响应消息的数据参数。

表 107 — 响应消息数据参数定义

定义
事件类型 该参数是请求消息的子功能参数的位 6 — 0 的回显。
活动事件的事件类型 此参数是为设置活动事件而发出的请求消息的子功能参数的回显。 适用的值是为事件类型子功能参数指定的值。
激活事件的数量 此参数包含客户端请求报告活动事件数时活动事件的数量。 该数字反映了响应消息中报告的事件数量。
识别事件的数量 此参数包含活动事件窗口期间识别事件的数量，并且仅适用于在事件窗口结束时发送的响应消息（在有限事件窗口的情况下）。 对请求消息的初始响应应在此参数中包含一个零（0）。
事件窗口时间 此参数是请求消息的事件窗口时间参数的回显。 报告活动事件时，此参数包含事件处于活动状态的剩余时间。
事件类型记录 该参数是请求消息的事件类型记录参数的回显。 当报告一个活动事件时，这个参数就是发出请求的事件类型记录的回显，该请求的发出是为了设置活动事件。
响应记录服务 该参数是请求消息的响应记录服务参数的回显。 当报告一个活动事件时，这个参数就是发出请求的响应记录服务的回显，该请求的发出是为了设置活动事件。

5.5.10.4 支持的否定响应码 (NRC_)

此服务应执行以下否定响应码。 表 108 中记录了每个响应码发生的情况。 如果错误情形适用于服务端，则应使用列出的否定响应。

表 108 — 支持的否定响应码

否定响应码	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS
0x13	不正确的消息长度或无效格式 如果请求消息的长度错误，则应发送否定响应码。	IMLOIF
0x22	条件不正确 当服务端处于关键的正常模式活动并因此无法执行所请求的功能时使用。	CNC

0x31	<p>请求超出范围</p> <p>服务端应使用此响应码</p> <p>如果它在事件类型记录参数中检测到错误;</p> <p>如果指定的事件窗口时间无效;</p> <p>如果请求的数据标识符 DID 不被支持;</p> <p>如果请求有限事件窗口和等于存储事件的存储状态的组合</p>	ROOR
------	--	------

5.5.10.5 消息流示例响应事件

5.5.10.5.1 假设

假设对于消息流示例，事件窗口时间等于 0x08 定义了一个 80 秒的事件窗口（事件窗口时间 * 10 秒）。客户端通过将抑制肯定响应消息指示位（子功能参数的位 7）设置为“错误” ('0') 来请求获得响应消息。

请注意，事件窗口时间的定义由汽车制造商特定，除了 B.2 中规定的某些值。

以下条件适用于显示的消息流示例和流程图：

—— 触发信号：

车辆制造商需要定义一个特定的触发信号，以使客户端（外部测试设备，车载自动诊断系统，诊断主站等）启动响应事件的请求消息。该触发信号可以通过事件以及像心跳时间（其应该大于事件窗口时间）的固定计时进度安排来启用。此外，在用作触发信号的数据链路上可能存在同步消息（例如，同步信号）。

—— 开启活动窗口：

接收到响应事件请求消息时，服务端应该疏远请求。如果评估结果为肯定，服务端应设置事件逻辑，并且必须发送对响应事件服务的初始肯定响应消息。要激活事件逻辑，客户端必须请求响应事件子功能启动响应事件。在肯定响应之后，事件逻辑被激活并且事件窗口计时器正在运行。由车辆制造商使用参数事件窗口时间（例如，时间窗口，点火开/关窗口）来详细定义事件窗口。在检测到指定事件类型（EART_）的情况下，服务端必须依照响应事件请求消息中的响应记录服务，立即以响应消息进行响应。

响应消息对应于响应事件请求消息中的响应记录服务。

—— 关闭活动窗口：

建议根据参数事件窗口时间关闭服务端的事件窗口。在此操作之后，服务端必须停止发送事件驱动的诊断响应消息。通过发送响应事件（ROE_）请求消息（包括参数停止响应事件或关闭电源）可以达到相同的目的。

5.5.10.5.2 示例 1 — 响应事件（有限事件窗口）

表 109 定义了响应事件请求消息流示例 1 的设置。

表 109 响应事件请求的设置消息流示例 1

消息方向	客户端 <input checked="" type="checkbox"/> 服务端 <input type="checkbox"/>		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	响应事件请求服务标识符 SID	0x86	ROE
#2	事件类型记录 [事件类型] = 关于 DTC 状态更改, 存储状态 = 不存储事件 抑制肯定响应消息指示位 = 错误	0x01	ET_ODTCSC
#3	事件窗口时间 = 80 秒	0x08	EWT
#4	事件类型记录[事件类型参数] = 测试失败状态	0x01	ETP1

#5	响应记录服务 [服务标识符] = 读取 DTC 信息	0x19	RDTCI
#6	响应记录服务[子功能] =按状态掩码报告 DTC 数量	0x01	RNDTC
#7	响应记录服务 [DTC 状态掩码] = 测试失败状态	0x01	DTCSM

表 110 定义了响应事件初始肯定响应消息流示例 1。

表 110 响应事件初始肯定响应消息流示例 1

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件响应服务标识符 SID	0xC6	ROEPR
#2	事件类型 = 关于 DTC 状态更改	0x01	ET_ODTCSC
#3	识别的事件数量 = 0	0x00	NOIE
#4	事件窗口时间 = 80 秒	0x08	EWT
#5	事件类型记录 [事件类型参数] = 测试失败状态	0x01	ETP1
#6	响应记录服务[服务标识符] = 读取 DTC 信息	0x19	RDTCI
#7	响应记录服务[子功能] =按状态掩码报告 DTC 数量	0x01	RNDTC
#8	响应记录服务 [DTC 状态掩码] = 测试失败状态	0x01	DTCSM

事件逻辑被设置；现在它必须被激活。

表 111 定义了响应事件请求消息流的开始示例 1。

表 111 响应事件请求消息流的开始示例 1

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件请求服务标识符 SID	0x86	ROE
#2	事件类型记录 [事件类型] = 开始响应事件, 存储状态 = 不存储事件 抑制肯定响应消息指示位 = 错误	0x05	ET_STRTROE
#3	事件窗口时间 (不会被评估)	0x08	EWT

表 112 定义了响应事件肯定响应消息流示例 1。

表 112 — 响应事件肯定响应消息流示例 1

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件响应服务标识符 SID	0xC6	ROEPR
#2	事件类型 = 关于 DTC 状态更改	0x01	ET_ODTCSC
#3	识别的事件数量 = 0	0x00	NOIE
#4	事件窗口时间	0x08	EWT

如果发生指定的事件，服务端将根据指定的响应记录服务发送响应消息。

表 113 定义了读取 DTC 信息肯定响应消息流示例 1。

表 113 — 读取 DTC 信息肯定响应消息流示例 1

消息方向	服务端 → 客户端		
消息类型	响应		
一个数据字	描述 (所有值均为十六进制)	字节值	助记符
#1	读取 DTC 信息响应服务标识符 SID	0x59	RDTCL
#2	DTC 状态可用性掩码	0xFF	DTCSAM
#3	DTC 计数[DTC 计数高字节] = 0	0x00	DTCCNT_HB
#4	DTC 计数[DTC 计数低字节] = 4	0x04	DTCCNT_LB

客户端在活动事件窗口期间请求报告服务端中当前活动事件的消息流程如下所示。

表 114 定义了响应事件请求活动事件数量的消息流示例 1

表 114 — 响应事件请求活动事件数量的消息流示例 1

消息方向	客户端 → 服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件请求服务标识符 SID	0x86	ROE
#2	事件类型记录 [事件类型] = 报告激活事件, 存储状态 = 不存储事件 抑制肯定响应消息指示位 = 错误	0x04	ET_RAE
#3	事件窗口时间 (不会被评估)	0x08	EWT

表 115 定义了响应事件报告激活事件肯定响应消息流示例 1。

表 115 — 响应事件报告激活事件肯定响应消息流示例 1

消息方向	服务端 → 客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件响应服务标识符 SID	0xC6	ROEPR
#2	事件类型 = 报告激活事件	0x04	ET_RAE
#3	识别的事件数量 = 1	0x01	NOAE
#4	活动事件的事件类型 = 关于 DTC 状态更改	0x01	ET_ODTCSC
#5	事件窗口时间= 80 秒	0x08	EWT
#6	事件类型记录 [事件类型参数] = 测试失败状态	0x01	ETP1
#7	响应记录服务[服务标识符] = 读取 DTC 信息	0x19	RDTCL
#8	响应记录服务[子功能] = 按状态掩码报告 DTC 数量	0x01	RNDTC
#9	响应记录服务 [DTC 状态掩码] = 测试失败状态	0x01	DTCSM

如果指定的事件窗口时间已过，服务端应发送最终肯定响应。

表 116 定义了响应事件最终肯定响应消息流示例 1。

表 116—响应事件最终肯定响应消息流示例 1

消息方向	服务端	客户端	
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件响应服务标识符 SID	0xC6	ROEPR
#2	事件类型 = 关于 DTC 状态更改	0x01	ET_ODTCSC
#3	识别的事件数量 = 1	0x01	NOIE
#4	事件窗口时间= 80 秒	0x08	EWT
#5	事件类型记录 [事件类型参数] = 测试失败状态	0x01	ETP1
#6	响应记录服务[服务标识符] = 读取 DTC 信息	0x19	RDTCI
#7	响应记录服务[子功能] =按状态掩码报告 DTC 数量	0x01	RNDTC
#8	响应记录服务 [DTC 状态掩码] = 测试失败状态	0x01	DTCSM

5.5.10.5.3 示例 1 流程图

以下流程图显示了两种不同类型的服务端行为：

- 在有限事件窗口内不会发生事件。 在这种情况下，服务端必须在事件窗口结束时发送响应事件的响应。
- 在有限事件窗口内的多个事件 (#1 到 #n)。响应服务的每个肯定响应都与一个已识别的事件 (#1...#n) 相关，并且应该具有相同的服务标识符 (SID)，但可能有不同的内容。在事件窗口结束时，服务端应传送响应事件服务的肯定响应消息，该响应消息表示已识别事件的数量。

图 11 描绘了有限事件窗口 — 在活动事件窗口期间没有事件。

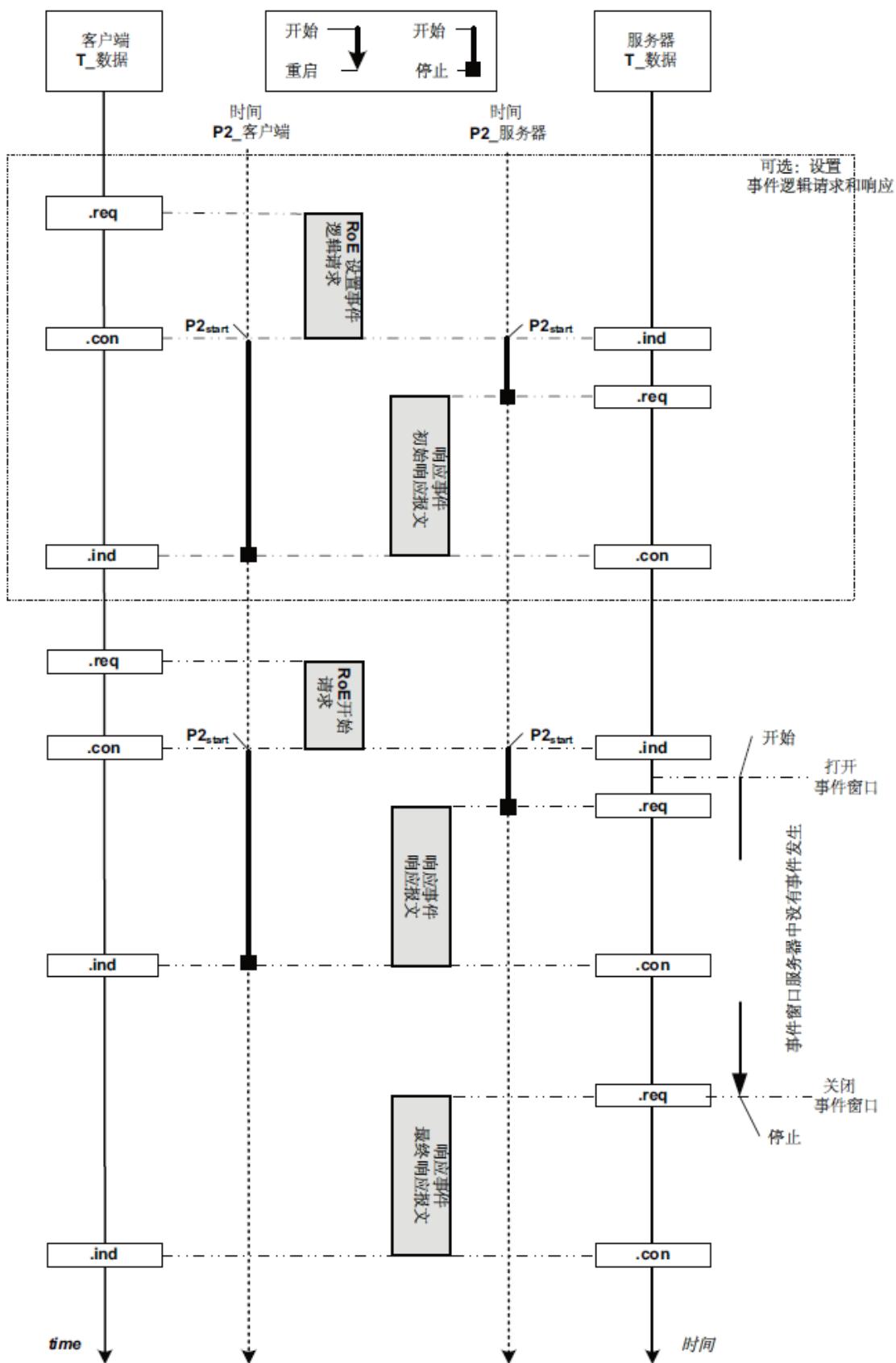


图 11 — 有限事件窗口 — 在活动事件窗口期间没有事件

图 12 描绘了有限事件窗口 — 活动事件窗口期间的多个事件。

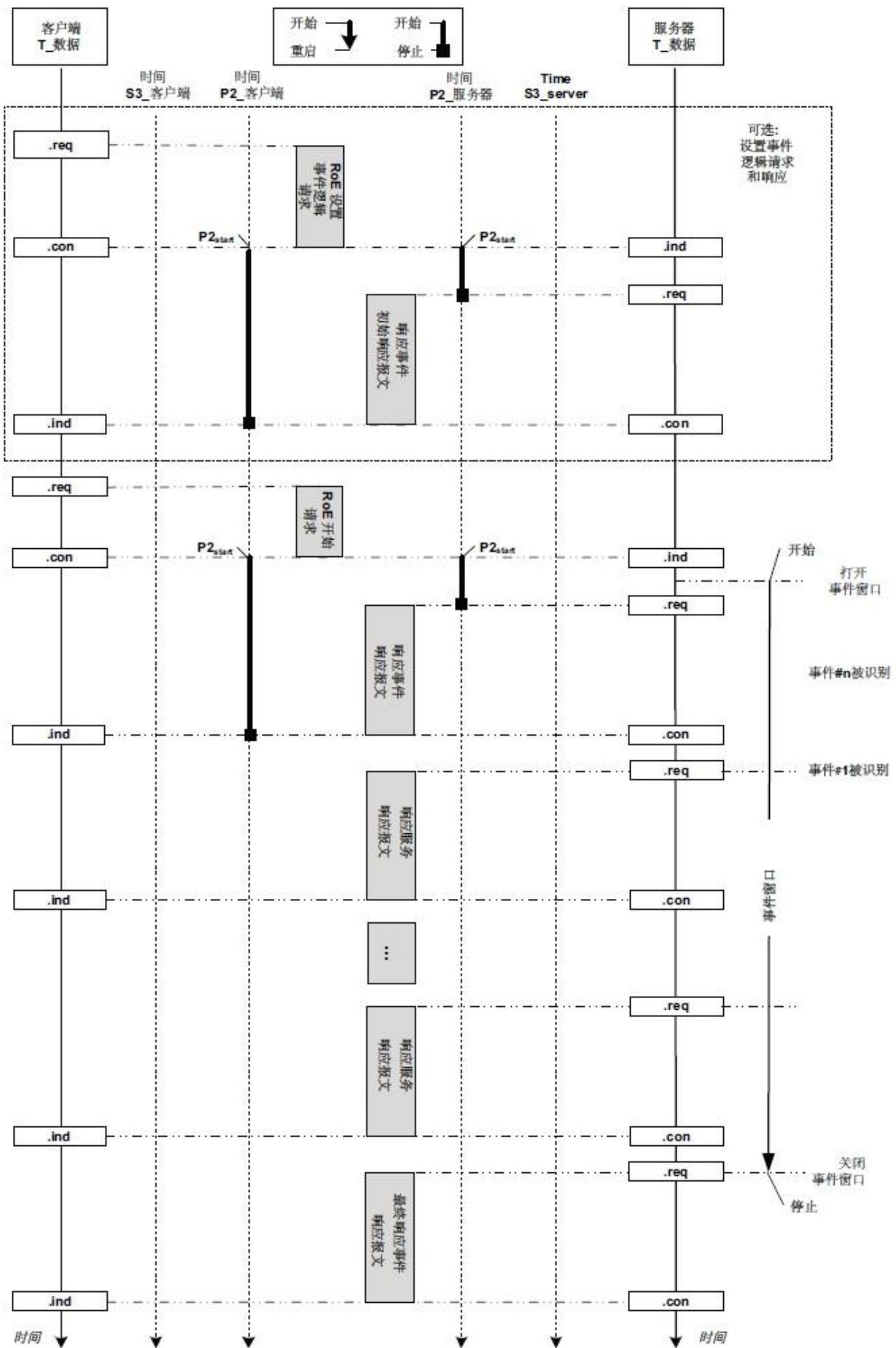


图 12 — 有限事件窗口 — 活动事件窗口期间的多个事件

5.5.10.5.4 示例 2 响应事件（无限事件窗口）

表 117 定义了响应事件请求消息流示例 2。

表 117 — 响应事件请求消息流示例 2

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	响应事件请求服务标识符 SID	0x86	ROE
#2	事件类型记录 [事件类型] = DTC 状态更改, 存储状态 = 不存储事件 抑制肯定响应消息指示位 = 错误	0x01	ET_ODTCSC
#3	事件窗口时间 = 无限	0x02	EWT
#4	事件类型记录 [事件类型参数] = 测试失败状态	0x01	ETP1
#5	响应记录服务[服务标识符] = 读取 DTC 信息	0x19	RDTCI
#6	响应记录服务[子功能] = 按状态掩码报告 DTC 数量	0x01	RNDTC
#7	响应记录服务 [DTC 状态掩码] = 测试失败状态	0x01	DTCSM

表 118 定义了响应事件初始肯定响应消息流示例 2。

表 118 — 响应事件初始肯定响应消息流示例 2

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	响应事件响应服务标识符 SID	0xC6	ROEPR
#2	事件类型 = 关于 DTC 状态更改	0x01	ET_ODTCSC
#3	识别的事件数量 = 0	0x00	NOIE
#4	事件窗口时间 = 无限	0x02	EWT
#5	事件类型记录 [事件类型参数] = 测试失败状态	0x01	ETP1
#6	响应记录服务[服务标识符] = 读取 DTC 信息	0x19	RDTCI
#7	响应记录服务[子功能] = 按状态掩码报告 DTC 数量	0x01	RNDTC
#8	响应记录服务 [DTC 状态掩码] = 测试失败状态	0x01	DTCSM

事件逻辑被设置；现在它必须被激活。

表 119 — 定义了响应事件开始请求消息流示例 2。

表 119 — 响应事件开始请求消息流示例 2

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	响应事件请求服务标识符 SID	0x86	ROE
#2	事件类型记录 [事件类型] = 开始响应事件, 存储状态 = 不存储事件 抑制肯定响应消息指示位 = 错误	0x05	ET_STRTROE

#3	事件窗口时间 (不会被评估)	0x02	EWT
----	----------------	------	-----

表 120 定义了响应事件肯定响应消息流示例 2。

表 120 — 响应事件肯定响应消息流示例 2

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件响应服务标识符 SID	0xC6	ROEPR
#2	事件类型 = 关于 DTC 状态更改	0x05	ET_ODTCSC
#3	识别的事件数量 = 0	0x00	NOIE
#4	事件窗口时间	0x02	EWT

如果发生指定的事件，服务端将根据指定的响应记录服务发送响应消息。

表 121 定义了读取 DTC 信息肯定响应消息流示例 2。

表 121 — 读取 DTC 信息肯定响应消息流示例 2

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取 DTC 信息响应服务标识符 SID	0x59	RDTCI
#2	DTC 状态可用性掩码	0xXX	DTCSAM
#3	DTC 计数[DTC 计数高字节] = 0	0x00	DTCCNT_HB
#4	DTC 计数[DTC 计数低字节] = 4	0x04	DTCCNT_LB

5.5.10.5.3.1 示例 2 — 流程图

以下流程图显示了两种不同类型的服务端行为：

- 无限事件窗口内没有事件发生。
- 无限事件窗口中的多个事件 (#1 至 #n)。 响应服务的每个肯定响应都与一个已识别的事件 (#1..#n) 相关联，并且应该具有相同的服务标识符 (SID)，但可能有不同的内容。

图 13 描述了无限事件窗口 — 活动事件窗口期间没有事件。

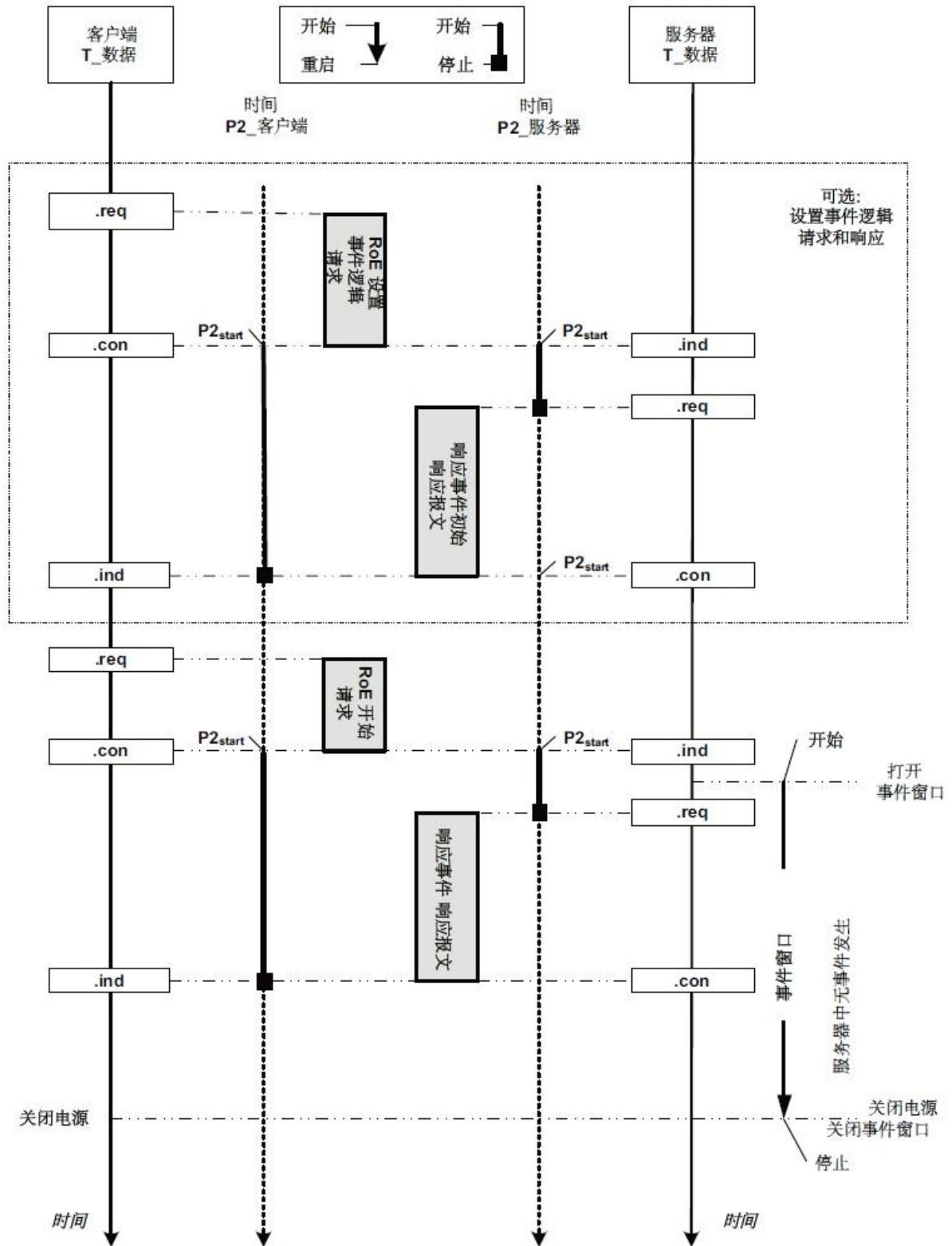


图 13 — 无限事件窗口 — 活动事件窗间无事件

图 14 描绘了无限事件窗口 — 活动事件窗口期间的多个事件。

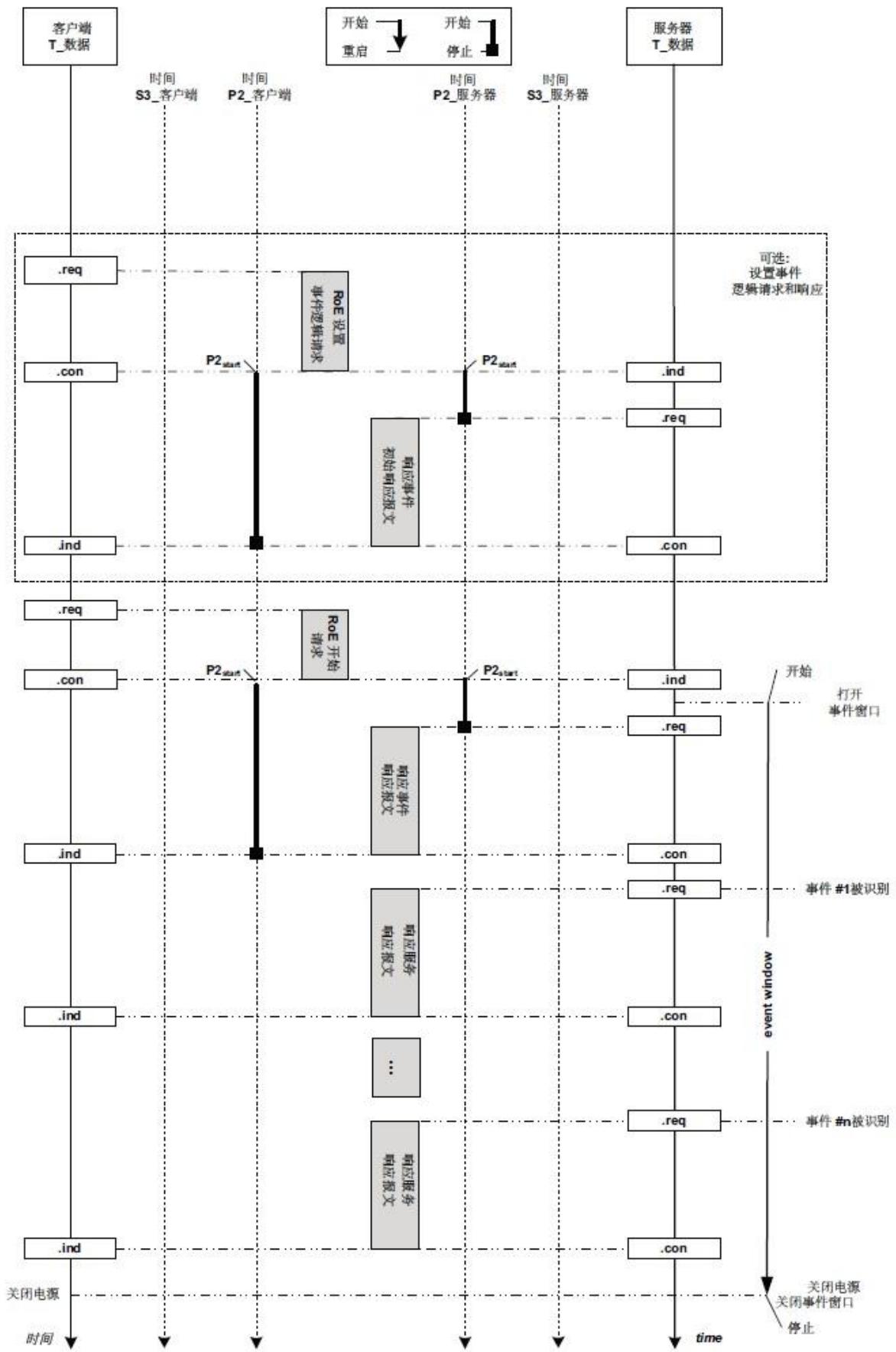


图 14 — 无限事件窗口 — 活动事件窗口期间的多个事件

5.5.10.5.5 示例 3 — 响应事件（无限事件窗口）— “值的比较”的子功能参数

这个例子只解释了假设在例#1 和例#2 中描述的 ROE 服务的通信行为没有改变的情况下，“关于值的比较”的子功能参数的使用。因此这个例子没有描述完整的消息流。相反，只会显示并解释事件窗口设置请求消息和发生事件的肯定响应消息。在上面的例子中已经描述了开始和停止请求消息以及不同的响应消息。以下条件适用：

- 服务 0x22 – 按标识符读取数据被选择为响应服务，
- 数据标识符 0x0104 包含在数据字节 #11 和 #12 处进行比较的测量值（该测量值也可以通过使用服务 0x22 读取），
- 如果测量值 (MV) 高于所谓的比较参数 (CP)，则事件会发生，因此运算符值（请参见下面的描述）选择为 0x01 — “MV> CP”，
- 当滞后值 0xA — 10% 被选中时，
- 当事件窗口时间的值 0x02 — “无限”被选中时，
- 当存储状态（事件类型子功能位 6）的值 1 的二进制 —— “存储事件”被选中时，
- 在任何情况下都会要求回复。

示例的定义：

- 字节#4&5: 数据标识符 0x0104
- 字节#6&7: 阅读和定义阅读类型的本地化。

示例 1 如果读数在数据记录的第 11 个字节中，则以下内容适用：

- $11 \times 8 = 88 \text{ dec} = 000101\ 1000\text{b}$ 位#10 – 位#14: 以位为单位的长度 — 1。
- 如果有 5 位，则最大大小为 32 位 = “长”。

示例 2 对于一个“字”，长度是 $15 \text{ dec} = 0\ 1111\text{b}$ 位#15: 符号输入：1=有符号，0=无符号。

示例 3 总分配为：

- $1011\ 1100\ 0101\ 1000\text{b} = 0xBC58$ 因此 字节#6 包含 0xBC，字节#7 包含 0x58
- 字节#8: 比较运算（运算符）

示例 4 运算符 $MV > CP = 0x01$

- 字节#9—12: 由于 4 字节长度的比较参数，可以传输从‘位’到‘长度’类型的所有数据格式。

示例 5 如果比较值是 $5\ 242 \text{ dec} = 0x0000\ 147A$

- 字节#9 = 0x00, 字节#10 = 0x00, 字节#11 = 0x14 and 字节#12 = 0x7A
- 字节#13: 滞后值（指定为比较参数的百分比）。该值是直接指定的。它仅适用于运算符“<”和“>”。在比较值为零的情况下，滞后值应被定义为绝对值。

示例 6 滞后值 $10\% = 0xA$

表 122 定义了响应事件请求消息示例 3。

表 122 — 响应事件请求消息示例 3

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	响应事件请求服务标识符 SID	0x86	ROE
#2	事件类型记录 [事件类型] = 关于值的比较, 存储状态 = 存储事件 抑制肯定响应消息指示位 = 错误	0x47	ET_OCOV
#3	事件窗口时间 = 无限	0x02	EWT
#4	事件类型记录 [事件类型参数#1] = 记录数据标识符(高字节)	0x01	ETR_ETP
#5	事件类型记录 [事件类型参数#2] = 记录数据标识符(低字节)	0x04	
#6	事件类型记录 [事件类型参数#3] = Valueinfo#1	0xBC	ETR_ETP2
#7	事件类型记录 [事件类型参数#4] = Valueinfo#2	0x58	ETR_ETP3
#8	事件类型记录 [事件类型参数#5] = 运算符	0x01	ETR_ETP4
#9	事件类型记录 [事件类型参数#6] = 比较参数 (字节#4)	0x00	ETR_ETP5
#10	事件类型记录 [事件类型参数#7] = 比较参数(字节#3)	0x00	ETR_ETP6
#11	事件类型记录 [事件类型参数#8] = 比较参数(字节#2)	0x14	ETR_ETP7
#12	事件类型记录 [事件类型参数#9] = 比较参数(字节#1)	0x7A	ETR_ETP8
#13	事件类型记录 [事件类型参数#10] = 滞后 [%]	0x0A	ETR_ETP9
#14	响应记录服务 [服务标识符] = 按标识符读取数据	0x22	RDBI
#15	响应记录服务 e [服务参数#1] = 数据标识符 (最高有效位 MSB)	0x01	DID_B1
	响应记录服务 [服务参数#2] = 数据标识符 (最低有效位 LSB)	0x04	DID_B2

注释：响应消息和随后的初始化顺序未显示。

如果在成功设置和激活 ROE 机制的事件窗口后，测量值高于 5242d，则服务端会作出反应。指定的事件会发生并且服务端会发送以下消息。

表 123 定义了按标识符读取数据的肯定响应消息示例 3。

表 123 — 按标识符读取数据的肯定响应消息示例 3

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述 (所有值均为十六进制)	字节值	助记符
#1	按标识符读数据响应服务标识符 SID	0x62	RDBIPR
#2	数据标识符[字节#1] (最高有效位 MSB)	0x01	DID_B1
#3	数据标识符[字节#2] (最低有效位 LSB)	0x04	DID_B2
#4	数据记录[数据#1]	0xXX	DREC_DATA1
#5	数据记录[数据#2]	0xXX	DREC_DATA2
#6	数据记录[数据#3]	0xXX	DREC_DATA3
#7	数据记录[数据#4]	0xXX	DREC_DATA4
#8	数据记录 [数据#5]	0xXX	DREC_DATA5

#9	数据记录[数据#6]	0xXX	DREC_DATA6
#10	数据记录[数据#7]	0xXX	DREC_DATA7
#11	数据记录[数据#8]	0xXX	DREC_DATA8
#12	数据记录[数据#9]	0xXX	DREC_DATA9
#13	数据记录[数据#10]	0xXX	DREC_DATA10
#14	数据记录[数据#11] 字节#11 的数据内容: 0x14	0x14	DREC_DATA11
#15	数据记录[数据#12] 字节#12 的数据内容: 0x7B	0x7B	DREC_DATA12
:	:	:	:

在测量值至少低于比较参数值的 90% 之前发生另一事件。这种行为由滞后值指定。如果满足该条件并且测量值再次高于比较值，则会发生新事件，并由服务端发送新的按标识符读取数据的响应消息。

5.5.11 链路控制（0x87）服务

5.5.11.1 服务描述

链路控制服务用于控制客户端和服务端之间的通信，以获得用于诊断目的（例如，编程）的总线带宽。该服务可选择应用于数据链路层，这样可以在非默认诊断会话期间重新配置其通信参数（例如，更改总线 CAN 上的波特率或重新配置 FlexRay 周期设计）。

注：有关在特定数据链路层上应用和使用此服务的更多详细信息，参见单个数据链路层特定诊断服务实现 UDSonXYZ' “数据链路” 规范。

该服务用于将数据链路层转换为特定状态，从而允许更高诊断带宽，主要出于编程目的。为了克服功能通信限制（例如，波特率必须同时在多个服务端中转换），转换本身分为两个步骤：

—— 第 1 步：客户端验证是否可以执行转换并通知服务端有关要使用的模态转换机制。在客户端执行第 2 步之前，每个服务端都必须积极回应（抑制肯定响应消息指示位=false）。这一步实际上并不执行模态转换。

—— 第 2 步：客户端实际上在请求模态转换（例如，较高的波特率）。如果步骤 1 已成功执行，则只能请求该步骤。在功能通信的情况下，建议在执行模态转换时（抑制肯定响应消息指示位=肯定）服务端不应有任何响应，因为一个服务端可能已经转换到新模式，而其他服务端仍在进行中。

请求消息中的链路控制类型参数与条件链路控制模态标识符 / 链路记录参数一起提供了一种机制，可以通过预定义的模态转换参数或特殊定义的模态转换参数进行转换。

注：该服务与非默认会话绑定。会话层计时器超时将服务端返回到其（正常）操作模式。这同样适用于执行 ECU 复位服务（0x11）的情况。一旦发生了数据链路模态转换，任何附加的非默认会话请求都不应导致重新转换到默认的操作模式（例如，在编程会话期间）。

重要 —— 服务端和客户端应符合 7.5 中规定的请求和响应消息行为。

5.5.11.2 请求消息

5.5.11.2.1 请求消息定义

表 124 定义了请求消息（链路控制类型=验证带有固定参数的模态转换）。

表 124 — 请求消息定义（链路控制类型=验证带有固定参数的模态转换）

一个数据字节	参数名称	Cvt	字节值	助记符
#1	链路控制请求服务标识符 SID	M	0x87	LC

#2	子功能=[链路控制类型]	M	0x01	LEV_LCTP_
#3	链路控制模态标识符	M	0x00 – 0xFF	LCMI_

表 125 定义了请求消息（链路控制类型=验证带有特定参数的模态转换）

表 125 — 请求消息定义（链路控制类型=验证带有特定参数的模态转换）

一个数据字节	参数名称	Cvt	字节值	助记符
#1	链路控制请求服务标识符 SID	M	0x87	LC
#2	子功能=[链路控制类型]	M	0x02	LEV_LCTP_
#3 #4 #5	链路记录[] = [模态参数高字节 模态参数中字节 模态参数低字节]	M M M	0x00 – 0xFF 0x00 – 0xFF 0x00 – 0xFF	LBR_ MPHB MPMB MPLB

表 126 定义了请求消息（链路控制类型 = 转换模式）

表 126 — 请求消息定义（链路控制类型 = 转换模式）

一个数据字节	参数名称	Cvt	字节值	助记符
#1	链路控制请求服务标识符 SID	M	0x87	LC
#2	子功能=[链路控制类型]	M	0x03	LEV_LCTP_

5.5.11.2.2 请求消息子功能参数\$Level (LEV_) 定义

链路控制请求消息使用链路控制类型子功能参数，来描述要在服务端中执行的操作（抑制肯定响应消息指示位（位 7）未在下表中显示）。

表 127 定义了请求消息子功能参数。

表 127—请求消息子功能参数定义

位 6-0	描述	Cvt	助记符
0x00	ISOSAE 保留 此值由本文档保留。	M	ISOSAERESRVD
0x01	验证带有固定参数的模态转换 此参数用于验证是否可以执行对预定义的参数进行转换，该预定义的参数由链路控制模态标识符数据参数指定。	U	VMTWFP
0x02	验证带有特定参数的模态转换 此参数用于验证是否可以执行对特别定义的参数进行转换，该特殊定义的参数（例如特定波特率）由链路记录数据参数指定。	U	VMTWSP
0x03	转换模式 该子功能参数请求服务端将数据链路转换为在前面的验证消息中请求的模式。	U	TM
0x04 – 0x3F	ISOSAE 保留 此值的范围由本文档保留以备将来定义。	M	ISOSAERESRVD

0x40 – 0x5F	车辆制造商特定 此值的范围保留为车辆制造商的特定用途。	U	VMS
0x60 – 0x7E	系统供应商特定 此值的范围保留为系统供应商的特定用途。	U	SSS
0x7F	ISOSAE 保留 此值由本文档保留以备将来定义。	M	ISOSAERESRVD

5.5.11.2.3 请求消息数据参数定义

表 128 定义了请求消息的数据参数。

表 128 — 请求消息数据参数定义

定义
链路控制模态标识符 该条件参数引用了一个固定的已定义的模态参数来进行转换（参见 B.3）。
链路记录 在子功能参数使用特定参数的情况下，该条件参数记录包含特定的模态参数。链路记录的格式在单个数据链路特定的诊断规范(UDSonXYZ)中指定。

5.5.11.3 肯定响应消息

5.5.11.3.1 肯定响应消息定义

表 129 定义了肯定响应消息。

表 129 — 肯定响应消息定义

一个数据字节	参数名称	Cvt	字节值	助记符
#1	链路控制响应 服务标识符 SID	M	0xC7	LCPR
#2	链路控制类型	M	00—7F	LCTP

5.5.11.3.2 肯定响应消息数据参数定义

表 130 定义了肯定响应消息的数据参数。

表 130 — 响应消息数据参数定义

定义
链路控制类型 该参数是请求消息的链路控制类型子功能参数的位 6 — 0 的回显。

5.5.11.4 支持的否定响应码 (NRC_)

此服务应执行以下否定响应码。表 131 中记录了每个响应码发生的情况。如果错误情形适用于服务端，则应使用列出的否定响应。

表 131—支持的否定响应码

否定响应码	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此否定响应码。	SFNS

0x13	不正确消息或无效格式 如果消息的长度错误，则应发送此否定响应码。	IMLOIF
0x22	条件不正确 如果未满足请求的链路控制的条件，则应返回此否定响应码。	CNC
0x24	请求顺序错误 如果客户请求转换操作模式而没有前面的验证步骤（该步骤指定要转换至的模式），则应返回此否定响应码。	RSE
0x31	请求超出范围 应返回此否定响应码，如果 请求的链路控制模态标识符无效； 特定模态参数（链路记录）无效；	ROOR

5.5.11.5 链路控制消息流示例

5.5.11.5.1 示例 1 – 将波特率转换为固定波特率（PC 波特率 115200 kBit/s）

5.5.11.5.1.1 第 1 步：验证是否满足波特率转换的所有条件

表 132 定义了链路控制请求消息流示例 1（步骤 1）。

表 132—链路控制请求消息流示例 1（步骤 1）

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记
#1	链路控制请求服务标识符 SID	0x87	LC
#2	链路控制类型=验证带有固定参数的模态转换 抑制肯定响应消息指示位=false	0x01	VMTWFP
#3	链路控制模态标识符= PC115200 波特	0x05	BI_PC115200

表 133 定义了链路控制肯定响应消息流示例 1（步骤 1）。

表 133—链路控制肯定响应消息流示例 1（步骤 1）

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制响应服务标识符 SID	0xC7	LCPR
#2	链路控制类型=验证带有固定参数的模态转换	0x01	VMTWFP

5.5.11.5.1.2 第 2 步：转换波特率

表 134 定义了链路控制请求消息流示例 1（步骤 2）。

表 134 — 链路控制请求消息流示例 1（步骤 2）

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制请求服务标识符 SID	0x87	LC
#2	链路控制类型=转换模式 抑制肯定响应消息指示位=肯定	0x83	TM

服务端没有响应。客户端和服务端必须转换其通信链路的波特率。

5.5.11.5.2 示例 2 – 将波特率转换为特定波特率(150kBit/s)

5.5.11.5.2.1 第 1 步：验证是否满足波特率转换的所有条件

表 135 定义了链路控制请求消息流示例 2（步骤 1）。

表 135 — 链路控制请求消息流示例 2（步骤 1）

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制请求服务标识符 SID	0x87	LC
#2	链路控制类型=验证带有特定参数的模态转换， 抑制肯定响应消息指示位=false	0x02	VMTWSP
#3	链路记录[模态参数高字节] (150kBit/s)	0x02	MPHB
#4	链路记录[模态参数中字节]	0x49	MPMB
#5	链路记录[模态参数低字节]	0xF0	MPLB

表 136 定义了链路控制的肯定响应消息流示例 2（步骤 1）。

表 136—链路控制肯定响应消息流示例 2（步骤 1）

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制响应服务标识符 SID	0xC7	LCPR
#2	链路控制类型=验证带有特定参数的模态转换	0x02	VMTWSP

5.5.11.5.2.2 第 2 步：转换波特率

表 137 定义了链路控制请求消息流示例 2（步骤 2）。

表 137—链路控制请求消息流示例 2（步骤 2）

消息方向	客户端□服务端
消息类型	请求

一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制请求服务标识符 SID	0x87	LC
#2	链路控制类型=转换模态 抑制肯定响应消息指示位=肯定	0x83	TM

服务端没有响应。客户端和服务端必须转换其通信链路的波特率。

5.5.11.5.3 示例 3 — 将 FlexRay 循环设计转换为“编程”

以下示例反映了 FlexRay 网络周期设计转换为优化的“编程”模式（例如，利用增强的动态细分进行编程）的场景。

5.5.11.5.3.1 第 1 步：验证是否满足调度器转换的所有条件

表 139 定义了链路控制请求消息流示例 3（步骤 1）。

表 138—链路控制请求消息流示例 3（步骤 1）

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制请求服务标识符 SID	0x87	LC
#2	链路控制类型=验证带有固定参数的模态转换 抑制肯定响应消息指示位=false	0x01	VMTWFP
#3	链路控制模态标识符=编程设置	0x20	PROGSU

表 139 定义了链路控制的肯定响应消息流示例 3（步骤 1）。

表 139—链路控制肯定响应消息流示例 3（步骤 1）

消息方向	服务端□客户端		
消息类型	响应		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制响应服务标识符 SID	0xC7	LCPR
#2	链路控制类型=验证带有固定参数的模态转换	0x01	VMTWFP

5.5.11.5.3.2 第 2 步：转向编程调度

表 140 定义了链路控制请求消息流示例 3（步骤 2）。

表 140—链路控制请求消息流示例 3（步骤 2）

消息方向	客户端□服务端		
消息类型	请求		
一个数据字节	描述（所有值均为十六进制）	字节值	助记符
#1	链路控制请求服务标识符 SID	0x87	LC

#2	链路控制类型=转换模态，抑制肯定响应消息指示位=肯定	0x83	TM
----	----------------------------	------	----

服务端没有响应。客户端和服务端必须转换 FlexRay 通信链路的循环设计。

5.6 数据传输功能单元

5.6.1 综述

表141定义了数据传输功能单元。

表141 数据传输功能单元

服务	描述
通过标识符读取数据	客户端请求读取由提供的数据标识符标识的记录的当前值。
通过地址读取内存	客户端请求读取提供的内存范围中存储的当前值。
通过标识符读取缩放数据	客户端请求读取由提供的数据标识符标识的记录的缩放信息。
通过周期性标识符读取数据	客户端请求调度服务端中的数据以进行定期传输。
动态定义数据标识符	客户端请求动态定义数据标识符，该标识符可能随后由“通过标识符读取数据”服务读取。
通过标识符写数据	客户端请求写入由提供的数据标识符指定的记录。
通过地址写内存	客户端请求重写指定的内存范围。

5.6.2 通过标识符读取数据（0x22）服务

5.6.2.1 服务描述

“通过标识符读取数据”服务允许客户端向服务端请求由一个或多个数据标识符标识的数据记录值。

客户端请求消息包含一个或多个两字节的数据标识符值，用于标识服务端维护的数据记录（允许的数据标识符值见附录C.1）。数据记录的格式和定义应是车辆制造商或系统供应商指定的，可以包括服务端支持的模拟输入和输出信号，数字输入和输出信号，内部数据和系统状态信息。

服务端可以限制同时请求的数据标识符的数量。该数量由车辆制造商和系统供应商协商一致。

服务端在接收到“通过标识符读取数据”请求后，应访问由数据标识符参数指定的记录的数据元素，并在包含相关数据记录参数的单个“通过标识符读取数据”中传输其值。请求消息可以多次包含同一数据标识符。服务端应将每个数据标识符视为单独的参数，并根据需要对每个数据标识符进行数据响应。

重要：服务端和客户端应符合7.5中指定的请求和响应消息行为。

5.6.2.2 请求消息

5.6.2.2.1 请求消息定义

表142定义了请求消息。

表142-请求消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过标识符读取数据 SID	M	0x22	RDBI
#2 #3	数据标识符[]#1 =[字节#1 (MSB) 字节#2]	M M	0x00-0xFF 0x00-0xFF	DID_ HB LB
..
#n-1 #n	数据标识符[]#m = [字节#1 (MSB) 字节#2]	U U	0x00-0xFF 0x00-0xFF	DID_ HB LB

5.6.2.2.2 请求消息子功能参数\$Level (LEV_) 定义

该服务不使用子功能参数。

5.6.2.2.3 请求消息数据参数定义

表143定义了请求消息的数据参数。

表143-请求消息数据参数定义

定义
数据标识符(#1 to #m) 此参数标识客户端请求的服务端数据记录(详细参数定义参见 C.1)。

5.6.2.3 肯定响应消息

5.6.2.3.1 肯定响应消息定义

表144定义了肯定响应消息。

表144 肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过标识符读取数据响应 SID	M	0x62	RDBIPR
#2 #3	数据标识符[]#1 =[字节#1 (MSB) 字节#2]	M M	0x00-0xFF 0x00-0xFF	DID_ HB LB
#4 .. #(k-1)+4	数据记录[]#1 =[数据#1 .. 数据#k]	M .. U	0x00-0xFF .. 0x00-0xFF	DREC_ DATA_1 .. DATA_m
..
#n-(o-1)-2 #n-(o-1)-2	数据标识符[]#m = [字节#1 (MSB) 字节#2]	U U	0x00-0xFF 0x00-0xFF	DID_ HB LB
#n-(o-1) .. #n	数据记录[]#1 =[数据#1 .. 数据#o]	U .. U	0x00-0xFF .. 0x00-0xFF	DREC_ DATA_1 .. DATA_k

5.6.2.3.2 肯定响应消息数据参数定义

表145定义了肯定响应消息的数据参数。

表145-响应消息数据参数定义

定义
数据标识符(#1 to #m) This parameter is an echo of the data-parameter dataIdentifier from the request message. 该参数是与来自请求消息的数据参数的数据标识符相同。
数据记录(#1 to #k/o) “通过标识符读取数据”肯定响应消息使用该参数向客户端提供所请求的数据记录值。数据记录的内容未在本标准中定义，由车辆制造商制定。

5.6.2.4 支持的否定响应码 (NRC_)

本服务应执行以下否定响应码。表146记录了每个响应码出现的情况。如果错误场景应用于服务端，则应使用列出的否定响应。

表146 列出的否定响应码

NRC	描述	助记符
0x13	不正确的消息长度或无效格式 如果请求消息的长度无效，或者客户端每次超出允许请求的数据标识符的最大数量时，应发送该 NRC。	IMLOIF
0x14	响应过长 如果响应消息的总长度超过底层传输协议的限制(例如，当单个请求中请求多个 DID 时)，则发送该 NRC。	RTL

0x22	条件不正确 如果不满足服务端的操作条件，则应发送该 NRC 以执行所需的操作。	CNC
0x31	请求范围 以下情况下，应发送该 NRC： ——设备不支持所请求的数据标识符的值， ——当前会话中不支持所请求的数据标识符； ——所请求的动态定义数据标识符尚未分配。	ROOR
0x33	安全访问拒绝 如至少有一个数据标识符加密且服务端未处于解锁状态，则应发送该 NRC。	SAD

评估序列见图15。

Figure 15 — NRC handling for ReadDataByIdentifier service
图15 通过标识符读取数据服务NRC处理流程

5.6.2.5 通过标识符读取数据消息流示例

5.6.2.5.1 假设

本节中指定了执行通过标识符读取数据服务的示例所需满足的条件。客户端可以在任何时候以独立于服务端的状态请求数据标识符数据。

以下数据标识符示例特指动力系统设备(例如,发动机引擎控制模块)。排放相关系统的术语/定义/缩略语参见ISO 15031 - 2 [6]。

第一个示例读取单个2字节的数据标识符，该数据标识符包含单个信息(其中数据标识符 0xF190包含VIN代码)。

第二个示例演示了用一个请求消息请求多个数据标识符 (其中数据标识符0x010A包含发动机冷却液温度、油门位置,发动机转速,进气歧管绝对压力,空气质量流量,车速传感器、气压、负载,怠速空气控制,和油门踏板位置;数据标识符 0x0110包含电池正电压)。

5.6.2.5.2 示例#1：读取单个数据标识符 0xF190（VIN 代码）

表147定义了通过标识符读取数据请求消息流示例#1。

表147通过标识符读取数据请求消息流示例#1

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述 (所有数值均为十六进制形式)	字节值	助记符
#1	通过标识符读取数据 SID	0x22	RDBI
#2	数据标识符[字节#1] (MSB)	0xF1	DID_B1
#3	数据标识符[字节#2]	0x90	DID_B2

表148定义了通过数据标识符读取数据肯定响应消息流示例#1。

表148-通过数据标识符读取数据肯定响应消息流示例#1

消息方向	服务端 → 客户端		
消息类型	响应		
A_Data 字节	描述 (所有数值均为十六进制形式)	字节值	助记符
#1	通过标识符读取数据响应 SID	0x62	RDBIPR
#2	数据标识符[字节#1] (MSB)	0xF1	DID_B1
#3	数据标识符[字节#2]	0x90	DID_B2
#4	数据记录[数据#1] = VIN 数字 1 = “W”	0x57	DREC_DATA1
#5	数据记录[数据#2] = VIN 数字 2 = “0”	0x30	DREC_DATA2
#6	数据记录[数据#3] = VIN 数字 3 = “L”	0x4C	DREC_DATA3
#7	数据记录[数据#4] = VIN 数字 4 = “0”	0x30	DREC_DATA4
#8	数据记录[数据#5] = VIN 数字 5 = “0”	0x30	DREC_DATA5

#9	数据记录[数据#6] = VIN 数字 6 = “0”	0x30	DREC_DATA6
#10	数据记录[数据#7] = VIN 数字 7 = “0”	0x30	DREC_DATA7

5.6.2.5.3 示例#2：读取多个数据标识符 0x010A 和 0x0110

表149定义了通过标识符读取数据请求消息流示例#2。

表149 通过标识符读取数据请求消息流示例#2

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述 (所有数值均为十六进制形式)	字节值	助记符
#1	通过数据标识符读取数据请求 SID	0x22	RDBI
#2	数据标识符#1[字节#1] (MSB)	0x01	DID_B1
#3	数据标识符#2[字节#2]	0x0A	DID_B2
#4	数据标识符#2[字节#1] (MSB)	0x01	DID_B1
#5	数据标识符#2[字节#2]	0x10	DID_B2

表150定义了通过标识符读取数据肯定响应消息流示例#2。

表150 通过标识符读取数据肯定响应消息流示例#2

消息方向	服务端 → 客户端		
消息类型	响应		
A_Data 字节	描述 (所有数值均为十六进制形式)	Byte Value	Mnemonic
#1	通过数据标识符读取数据响应 SID	0x62	RDBIPR
#2	数据标识符[字节#1] (MSB)	0x01	DID_B1
#3	数据标识符[字节#2] (LSB)	0x0A	DID_B2
#4	数据记录[数据#1] = ECT	0xA6	DREC_DATA1
#5	数据记录[数据#2] = TP	0x66	DREC_DATA2
#6	数据记录[数据#3] = RPM	0x07	DREC_DATA3
#7	数据记录[数据#4] = RPM	0x50	DREC_DATA4
#8	数据记录[数据#5] = MAP	0x20	DREC_DATA5
#9	数据记录[数据#6] = MAF	0x1A	DREC_DATA6
#10	数据记录[数据#7] = VSS	0x00	DREC_DATA7
#11	数据记录[数据#8] = BARO	0x63	DREC_DATA8
#12	数据记录[数据#9] = LOAD	0x4A	DREC_DATA9
#13	数据记录[数据#10] = IAC	0x82	DREC_DATA10
#14	数据记录[数据#11] = APP	0x7E	DREC_DATA11
#15	数据标识符[字节#1] (MSB)	0x01	DID_B1
#16	数据标识符[字节#2] (LSB)	0x10	DID_B2
#17	数据记录[数据#1] = B+	0x8C	DREC_DATA1

5.6.3 通过地址读取内存 (0x23) 服务

5.6.3.1 服务描述

通过地址读取内存服务允许客户端通过提供的读取起始地址和内存大小来请求服务端的内存数据。

通过地址读取内存请求消息用于向服务端请求由参数内存地址和内存大小标识的内存数据。用于内存地址和内存大小参数的字节数由地址和长度格式标识符(低半字节和高半字节)定义。

也可以在内存地址或内存大小参数中使用固定的地址和长度格式标识符和未使用的字节，并将值0x00放在更高的地址位置。

如果内存区域重叠，可以使用额外的内存地址字节作为内存标识符(例如，使用内部和外部闪存)。

服务端通过“通过地址读取内存”肯定响应消息发送数据记录值。数据记录参数的格式和定义应由车辆制造商指定。数据记录参数可以包含服务端支持的模拟输入和输出信号，数字输入和输出信号，内部数据和系统状态信息。

重要：服务端和客户端应满足7.5中指定的请求和响应消息行为。

5.6.3.2 请求消息

5.6.3.2.1 请求消息定义

表151定义了请求消息。

表151 请求消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过地址读取内存请求 SID	M	0x23	R MBA
#2	地址和长度格式标识符	M	0x00 - 0xFF	ALFID
#3 : #(m-1)+3	内存地址[] = [字节#(MSB) : 字节#m]	M : C1	0x00-0xFF : 0x00 - 0xFF	MA_ B1 : Bm
#n-(k-1) : #n	内存大小[] = [字节#1 (MSB) : 字节#k]	M : C2	0x00 - 0xFF : 0x00 - 0xFF	MS_ B1 : Bk

C1: 该参数的存在取决于地址和长度格式标识符的地址长度信息参数。
C2: 该参数的存在取决于地址和长度标识符的内存大小信息。

5.6.3.2.2 请求消息子功能参数\$Level (LEV_) 定义

该服务不适用子功能参数。

5.6.3.2.3 请求消息数据参数定义

表152定义了请求消息的数据参数。

表152 请求消息数据参数定义

定义
地址和长度格式标识符 该参数是单字节值，每个半字节单独编码(示例值参见 H.1):位 7 - 4:内存大小参数的长度(字节数)位 3 - 0:内存地址参数的长度(字节数)
内存地址 参数内存地址是服务端内存中检索数据的起始地址。该地址使用的字节数由地址和长度标识符的低半字节(位 3 - 0)定义。内存地址参数中的字节#m 始终是服务端引用的地址中最低有效字节。地址中的最高有效字节可以用作内存标识符。使用内存标识符的一个例子是具有 16 位寻址和内存地址重叠的双处理器服务端(当给定的地址对任何一个处理器都有效，但产生不同的物理内存设备或使用内部和外部闪存时)。在这种情况下，内存地址参数中另一个未使用的字节可以指定为用于选择所需内存设备的内存标识符。此功能的使用应由车辆制造商/系统供应商定义。
内存大小 通过地址读取内存请求消息中的参数内存大小指从服务端内存中的内存地址指定的地址开始读取的字节数。该大小使用的字节数由地址和长度信息的高半字节(位 7 - 4)定义。

5.6.3.3 肯定响应消息

5.6.3.3.1 肯定响应消息定义

表153定义了肯定响应消息。

表 153 — 肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过地址读取内存响应 SID	M	0x63	RMBAPR

#2	数据记录[] =[数据#1 .. 数据#m]	M .. U	0x00-0xFF .. 0x00-0xFF	DREC_ DATA_1 .. DATA_m
----	----------------------------------	--------------	------------------------------	---------------------------------

5.6.3.3.2 肯定响应消息数据参数定义

表154定义了肯定响应消息的数据参数。

表154 响应消息数据参数定义

定义
数据记录
通过地址读取内存肯定响应消息使用该参数向客户端提供所请求的数据记录值。本标准中未定义数据记录的内容且应反映所请求的内存内容。数据格式应由车辆制造商/系统供应商定义。

5.6.3.4 支持的否定响应码(NRC)

本服务应执行以下否定响应码。表155记录了每个响应码出现的情况。如果错误场景应用于服务端，则应使用列出的否定响应。

表 155 — 支持的否定响应码

NRC	描述	助记符
0x13	不正确的消息长度或无效格式 如请求消息的长度有误，应发送该 NRC。	IMLOIF
0x22	条件不正确 如果不满足服务端的操作条件，则应发送该 NRC 以执行所需的操作。	CNC
0x31	请求范围超出 以下情况下，应发送该 NRC： ——区间 [0xMA, (0xMA + 0xMS -0x1)] 内的任一内存地址无效； ——区间 [0xMA, (0xMA + 0xMS -0x1)] 内的任一内存地址受限； ——请求消息中的内存大小参数为零。	ROOR
0x33	安全访问拒绝 如区间 [0xMA, (0xMA + 0xMS -0x1)] 内任一内存地址加密且服务端被锁定，则应发送该 NRC。	SAD

评估序列见图 16。

Figure 16 — NRC handling for ReadMemoryByAddress service

图 16 通过地址读取内存服务的 NRC 处理

5.6.3.5 通过地址读取内存消息流示例

5.6.3.5.1 假设

本条指定了执行通过地址读取内存服务示例所需满足的条件。该示例中的服务不受服务端的任何限制。

5.6.3.5.2 示例#1：通过地址读取内存 -4-字节（32 位）寻址

客户端从服务端的内存中自内存地址0x2048 1392开始读取259个数据字节。

表 156 定义了通过地址读取内存请求消息流示例#1。

表 156 通过地址读取内存请求消息流示例#1

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制形式）	字节值	助记符

#1	通过地址读取内存请求 SID	0x23	RMBA
#2	地址和长度格式标识符	0x24	ALFID
#3	内存地址 [字节#1] (MSB)	0x20	MA_B1
#4	内存地址 [字节#2]	0x48	MA_B2
#5	内存地址 [字节#3]	0x13	MA_B3
#6	内存地址 [字节#4]	0x92	MA_B4
#7	内存大小 [字节#1] (MSB)	0x01	MS_B1
#8	内存大小 [字节#2]	0x03	MS_B2

表157定义了通过地址读取内存肯定响应消息流示例#1。

表 157 通过地址读取内存肯定响应消息流示例#1

消息方向	服务端 → 客户端		
消息类型	响应		
A_Data 字节	描述 (所有数值均为十六进制形式)	字节值	助记符
#1	通过地址读取内存响应 SID	0x63	RMBAPR
#2	数据记录[数据#1] (内存单元#1)	0x00	DREC_DATA_1
.. #259+1	.. 数据记录[数据#259] (内存单元#259)	.. 0x8C	.. DREC_DATA_259

5. 6. 3. 5. 3 示例#2：通过地址读取内存-2 字节（16 位）寻址

客户端从服务端的内存中自内存地址0x4813开始读取5个数据字节。

表 158 定义了通过地址读取内存请求消息流示例#2。

表 158 通过地址读取内存请求消息流示例#2

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制形式)	字节值	助记符
#1	通过地址读取内存请求 SID	0x23	RMBA
#2	地址和长度格式标识符	0x12	ALFID
#3	内存地址 [字节#1 (MSB)]	0x48	MA_B1
#4	内存地址 [字节#2 (LSB)]	0x13	MA_B2
#5	内存大小 [字节#1]	0x05	MS_B1

表159定义了通过地址读取内存肯定响应消息流示例#2。

表 159 通过地址读取内存肯定响应消息流示例#2

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制形式)	字节值	助记符
#1	通过地址读取内存响应 SID	0x63	RMBAPR
#2	数据记录 [数据#1] (内存单元#1)	0x43	DREC_DATA_1
#3	数据记录 [数据#2] (内存单元#2)	0x2A	DREC_DATA_2
#4	数据记录 [数据#3] (内存单元#3)	0x07	DREC_DATA_3
#5	数据记录 [数据#4] (内存单元#4)	0x2A	DREC_DATA_4
#6	数据记录 [数据#5] (内存单元#5)	0x55	DREC_DATA_5

5. 6. 3. 5. 4 示例#3：通过地址读取内存，3-byte (24-bit) 寻址

客户端从服务端的外部RAM单元中自内存地址0x204813开始读取3个数据字节。

表160 定义了通过地址读取内存请求消息流示例#3。

表160-通过地址读取内容请求消息流示例#3

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制形式）	字节值	助记符
#1	通过地址读取内存请求 SID	0x23	RMBA
#2	地址和长度格式标识符	0x23	ALFID
#3	内存地址 [字节#1 (MSB)]	0x20	MA_B1
#4	内存地址 [字节#2]	0x48	MA_B2
#5	内存地址 [字节#3 (LBS)]	0x13	MA_B3
#6	内存大小 [字节#1 (MBS)]	0x00	MS_B1
#7	内存大小 [字节#2 (LBS)]	0x03	MS_B2

表161定义了通过地址读取内存首个肯定响应消息，示例#3。

表161 通过地址读取内存首个肯定响应消息，示例#3

消息方向	服务端 → 客户端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制形式）	字节值	助记符
#1	通过地址读取内存响应 SID	0x63	RMBAPR
#2	数据记录 [数据#1] (内存单元#1)	0x00	DREC_DATA_1
#3	数据记录 [数据#2] (内存单元#2)	0x01	DREC_DATA_2
#4	数据记录 [数据#3] (内存单元#3)	0x8C	DREC_DATA_3

5.6.4 通过标识符读取缩放数据 (0x24) 服务

5.6.4.1 服务描述

通过标识符读取缩放数据服务允许客户端从由数据标识符标识的服务端请求缩放数据记录信息。

客户端请求消息包含一个数据标识符值，该值标识服务端维护的数据记录(允许的数据标识符值参见附录C.1)。数据记录的格式和定义应由车辆制造商指定，可以包括服务端支持的模拟输入和输出信号、数字输入和输出信号、内部数据和系统状态信息。

服务端在接收到通过标识符读取缩放数据请求后，服务端应访问与指定的数据标识符参数相关联的缩放信息，并在通过数据标识符读取缩放数据的肯定响应中传输缩放信息值。

重要：服务端和客户端应满足7.5中指定的请求和响应消息行为。

5.6.4.2 请求消息

5.6.4.2.1 请求消息定义

表162定义了请求消息。

表162 请求消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过标识符读取缩放数据请求 SID	M	0x24	RSDBI
#2 #3	数据标识符[] =[字节#1 (MSB) 字节#2]	M M	0x00-0xFF 0x00-0xFF	DID_ HB LB

5.6.4.2.2 请求消息子功能参数\$Level (LEV_) 定义

该服务不适用子功能参数。

5.6.4.2.3 请求消息数据参数定义

表163定义了请求消息的数据参数。

表163 请求消息数据参数定义

定义
数据标识符 此参数标识客户端请求的服务端数据记录(详细参数定义请参见 C.1)。

5.6.4.3 肯定响应消息

5.6.4.3.1 肯定响应消息定义

表164定义了肯定响应消息。

表164 肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过标识符读取缩放数据响应 SID	M	0x64	RSDBIPR
#2 #3	数据标识符[]#1 =[字节#1 (MSB) 字节#2 (LBS)]	M M	0x00-0xFF 0x00-0xFF	DID_ HB LB
#4	缩放字节#1	M	0x00-0xFF	SB_1
#5 .. #(p-1)+5	缩放字节扩展[]#1 =[缩放字节扩展参数#1 .. 缩放字节扩展参数#p]	C1 .. C1	0x00-0xFF .. 0x00-0xFF	SBE_ PAR1 .. PARp
..
#n-r	缩放字节#k	C2	0x00-0xFF	SB_k
#n-(r-1) .. #n	缩放字节扩展[]#k =[缩放字节扩展参数#1 .. 缩放字节扩展参数#r]	C1 .. C1	0x00-0xFF .. 0x00-0xFF	SBE_ PAR1 .. PARr
C1: 该参数的存在取决于缩放字节高半字节。如果缩放字节高半字节被编码为公式、单元/格式或无掩码映射报告位，则必须出现该参数。 C2: 该参数的存在取决于缩放信息的编码是否需要多于一个字节。				

5.6.4.3.2 肯定响应消息数据参数定义

表165定义了肯定响应消息的数据参数。

表165 响应消息数据参数定义

定义
数据标识符 该参数与来自请求消息的数据参数的数据标识符相同。
数据记录(#1 到 #k) 通过标识符读取缩放数据肯定响应消息使用该参数向客户端提供所请求的缩放数据记录值(详细参数定义参见附录C.2)。
缩放字节扩展 (#1 到 #p / #1 到 #r) 该参数用于为缩放字节提供附加信息，缩放字节的高半字节编码为公式、单元/格式或不带掩码的位元(详细参数定义参见附录C.3)。

5.6.4.4 支持的否定响应码 (NRC)

该服务应执行以下否定响应码。表 166 记录了每个响应代码出现的情况。如果错误场景应用于服务端，则应使用列出的否定响应。

表 166 支持的否定响应码

NRC	描述	助记符
0x13	不正确的消息长度或无效格式 如请求消息的长度有误，应发送该 NRC。	IMLOIF
0x22	条件不正确 如果不满足服务端的操作条件，则应发送该 NRC 以执行所需的操作。	CNC
0x31	请求范围超出 以下情况下，应返回该 NRC： ——设备不支持请求的数据标识符值； ——设备支持请求的数据标识符值，但是指定的数据标识符没有可用的缩放信息。。	ROOR
0x33	安全访问拒绝 如果数据标识符加密且服务端未处于解锁状态，则应发送该 NRC。	SAD

5.6.4.5 通过标识符读取缩放数据消息流示例

5.6.4.5.1 假设

本节指定了执行通过标识符读取缩放数据的示例所需满足的条件。客户端可以在任何时候不依赖于服务端的状态请求数据标识符缩放数据。

第一个示例读取与两个字节的数据标识符 0xF190 相关联的缩放信息，该信息包含单条信息(17个字符VIN码)。

第二个示例演示了如何使用公式和单元标识符来指定服务端中的数据变量。

第三个示例演示了如何使用通过标识符读取缩放数据来返回位映射的数据标识符所支持的位(有效性掩码)，该参数通过使用通过标识符读取缩放数据在没有掩码的情况下进行报告。

5.6.4.5.2 示例#1：通过 0xF190 标识符（VIN 码）读取缩放数据

表167定义了通过标识符读取缩放数据请求消息流示例#1。

表167 通过标识符读取缩放数据请求消息流示例#1

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述 (所有数值均为十六进制形式)	字节值	助记符
#1	通过标识符读取缩放数据请求 SID	0x24	RSDBI
#2	数据标识符 [字节#1] (MSB)	0xF1	DID_B1
#3	数据标识符 [字节#2] (LSB)	0x90	DID_B2

表168定义了通过标识符读取缩放数据肯定响应消息流示例#1。

表168 通过标识符读取缩放数据肯定响应消息流示例#1

消息方向	服务端 → 客户端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制形式)	字节值	助记符
#1	通过标识符读取缩放数据响应 SID	0x64	RSDBIPR
#2	数据标识符 [字节#1] (MSB)	0xF1	DID_B1
#3	数据标识符 [字节#2] (LSB)	0x90	DID_B2
#4	缩放字节#1 {ASCII, 15 数据字节}	0x6F	SB_1
#5	缩放字节#1 {ASCII, 2 数据字节}	0x62	SB_2

5.6.4.5.3 示例#2:通过标识符 0x0105 (车速) 读取缩放数据

表169定义了通过标识符读取缩放数据请求消息流示例#2。

表169 通过标识符读取缩放数据请求消息流示例#2

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节	描述 (所有数值均为十六进制形式)	字节值	助记符
#1	通过标识符读取缩放数据请求 SID	0x24	RSDBI
#2	数据标识符 [字节#1] (MSB)	0x01	DID_B1
#3	数据标识符 [字节#2]	0x05	DID_B2

表170定义了通过标识符读取缩放数据肯定响应消息流示例#2。

表170通过标识符读取缩放数据肯定响应消息流示例#2

消息方向	服务端 → 客户端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制形式)	字节值	助记符
#1	通过标识符读取缩放数据响应 SID	0x64	RSDBIPR
#2	数据标识符 [字节#1] (MSB)	0x01	DID_B1
#3	数据标识符 [字节#2] (LSB)	0x05	DID_B2
#4	缩放字节#1 {无符号数字, 1 数据字节}	0x01	SBYT_1
#5	缩放字节#1 {公式, 5 数据字节}	0x95	SB_2
#6	缩放字节扩展#2[字节#1] {公式标识符 = C0*x+C1}	0x00	SBE_21
#7	缩放字节扩展#2[字节#2] {C0 高字节}	0xE0	SBE_22
#8	缩放字节扩展#2[字节#3] {C0 低字节} [C0 = 75*10P-2P]	0x4B	SBE_23
#9	缩放字节扩展#2[字节#4] {C1 高字节}	0x00	SBE_24
#10	缩放字节扩展#2[字节#5] {C1 低字节} [C1=30*10POP]	0x1E	SBE_25
#11	缩放字节#3 {单元/格式, 1 数据字节}	0xA1	SB_3
#12	缩放字节扩展#3[字节#1] {单元 ID, km/h}	0x30	SBE_31

利用附录C.2中包含信息解码缩放字节、常量(C0、C1)和单元，计算车速数据变量，公式如下：

$$\text{车速} = (0.75 * x + 30) \text{ km/h}$$

其中'x'是存储在服务端中的实际数据，由数据标识符0x0105标识。

10.4.5.4 示例#3：通过0x0967标识符读取缩放数据

该示例展示了客户端如何确定服务端支持数据标识符的哪些位元，该服务端在没有有效掩码的情况下被格式化为位映射记录

该示例数据标识符 (0x0967) 定义见表171。

Table 171 — 数据定义示例

数据字节	位	描述
#1	7-4	未使用
	3	开启中速风扇
	2	检测到中速风扇输出错误
	1	清除监视器浸泡时间状态标志
	0	由于加油事件，清洗监视器空闲测试被阻止
#2	7	检查油帽灯亮着
	6	检测到油帽灯输出错误
	5	检测到风扇控制 A 输出错误
	4	检测到风扇控制 B 输出错误
	3	检测到高速风扇输出错误
	2	高速风扇输出开启

	1	清除监视空闲测试(轻微泄漏)准备运行
	0	件事清除监视器轻微泄漏

表172定义了通过标识符读取缩放数据请求消息流示例#3。

表172 通过标识符读取缩放数据请求消息流示例#3

消息方向	客户端 → 服务端	字节值	助记符
消息类型	请求		
A_Data 字节	描述 (所有数值均为十六进制形式)		
#1	通过标识符读取缩放数据请求 SID	0x24	RSDBI
#2	数据标识符 [字节#1] (MSB)	0x09	DID_B1
#3	数据标识符 [字节#2]	0x67	DID_B2

表173定义了通过标识符读取缩放数据肯定响应消息流示例#3。

表173 通过标识符读取缩放数据肯定响应消息流示例#3

消息方向	服务端 → 客户端	字节值	助记符
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制形式)		
#1	通过标识符读取缩放数据响应 SID	0x64	RSDBIPR
#2	数据标识符 [字节#1] (MSB)	0x09	DID_B1
#3	数据标识符 [字节#2] (LSB)	0x67	DID_B2
#4	缩放字节#1 {无掩码映射报告位, 2 数据字节}	0x22	SBYT_1
#5	缩放字节扩展#2[字节#1] {数据记录#1 有效掩码}	0x03	SBE_11
#6	缩放字节扩展#2[字节#1] {数据记录#1 有效掩码}	0x43	SBE_12

以上示例中假设服务端中的该数据标识符支持的位 (例如, 包含信息) 是字节#1、位 1 和 0, 以及字节#2、位 6、1 和 0。

5.6.5 通过周期标识符读取数据 (0x2A) 服务

5.6.5.1 服务描述

通过周期标识符读取数据服务允许客户端请求向由一个或多个周期数据标识符标识的服务端周期性的传输数据记录值。

客户端请求消息包含一个或多个单字节周期数据标识符值, 用于标识服务端维护的数据记录。周期数据标识符代表在该服务预留的数据标识符范围之外的数据标识符低字节(0xF2XX, 允许的周期数据标识符值见 C.1), 例如, 该服务中使用的周期数据标识符 0xE3 是数据标识符 0xF2E3。

数据记录的格式和定义应有车辆制造商定义, 可以包括服务端支持的模拟输入和输出信号、数字输入和输出信号、内部数据和系统状态信息。

在接收到除停止发送之外的通过周期数据标识符读取数据请求后, 服务端应检查执行服务的条件是否正确。

周期性标识符只能在给定的时间内使用单一的传输模式。在接收请求消息且该消息中将传输模式参数设置为同一周期数据标识符的新调度时, 应对周期数据标识符的调度进行更改。根据车辆制造商的要求, 应支持不同周期的多个进度。

应注意, 如果条件正确, 服务端发送一个肯定响应消息, 其中只包括服务标识符。一旦接收到初始请求消息并作出了肯定响应, 服务端不应发送否定响应消息。

在初始肯定响应消息之后, 服务端应访问周期数据标识符参数指定的记录的数据元素, 并将其值以包含相关数据记录参数的周期数据标识符作为独立的周期性数据响应消息进行传输。

在初始肯定响应消息之后, 为将周期数据标识符数据传输到客户端而定义的单独的周期数据响应消息应包括周期性数据标识符和周期性数据标识符的数据, 但不是肯定响应服务标识符。在 ISO 14229 的适当的实施规范中描述了周期性响应消息到数据链路层的映射。

对于特定的传输模式, 当仅有单个周期数据标识符被调度时, 所记录的周期速率被定义为两个连续的具有相同周期数据标识符的响应消息之间的时间间隔。如果多个周期性数据标识符被同时调度, 相同的周期性数据标识符之间的有效周期将基于以下设计参数而变化:

——周期性调度程序的调用速率；

——为每个调度程序调用而分配的可用的协议特定的周期数据响应消息信息标识符的数量(例如，CAN 标识符)；

——被定义为可同时并行传输的周期性数据标识符的数量。

如果同时传输多个周期数据标识符，该参数值将影响同一周期数据标识符之间的有效周期增加的程度。因此，上述所有设计参数应由车辆制造商指定。每次调用周期调度程序时，都应明确是否有任何周期性数据标识符可以进行传输。

注：周期速率是周期调度程序调用速率的整数倍。

例如，两个不同的 ECU 应用可能都支持快速传输模式，周期速率为 10 毫秒，并且有一个唯一的周期数据响应消息地址信息 ID。如果第一个应用每隔 10ms 调用周期调度程序，当调度两个周期数据标识符时，相同的周期数据标识符之间的时间间隔将增加到 20ms，当调度四个周期数据标识符时，时间将增加到 40ms。如果第二个应用每隔 5ms 调用周期调度程序，当调度两个周期数据标识符时，相同的周期数据标识符之间的时间将保持在 10ms，当调度四个周期数据标识符时，时间将增加到 20ms。的更多示例参见 10.6.5。

在接收到传输模式为“停止发送”的通过周期标识符读取数据请求后，服务端应停止请求消息中包含的周期性数据标识符的传输，或当请求消息中未指定特定的周期性数据标识符时，停止所有周期性数据标识符的传输。该传输模式的响应消息只包含服务标识符。

服务端可以限制同时支持的周期性数据标识符的数量，该数量为车辆制造商和系统供应商一致同意的。超过可以同时支持的周期性数据标识符的最大数量将导致否定响应，并且该请求中的周期性数据标识符都不会被调度。不应在单个请求消息中重复相同的周期数据标识符，如果客户端违反此规则，则服务端将忽略一个周期性数据标识符以外的所有标识符。

应注意，服务端和客户端应满足 7.5 中指定的请求和响应消息行为。

5.6.5.2 请求消息

5.6.5.2.1 请求消息定义

表174定义了请求消息。

表 174 请求消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过周期性标识符读取数据请求 SID	M	0x2A	RDBPI
#2	传输模式	M	0x00 - 0xFF	TM
#3	周期数据标识符[]#1	C	0x00 - 0xFF	PDIDI
:	:	:	:	:
#m+2	周期数据标识符[]#m	U	0x00 - 0xFF	PDIDm

C: 如果传输模式等于“慢速发送”、“中速发送”或“快速发送”，则在请求消息中必须显示第一个周期数据标识符。在传输模式等于“停止发送”的情况下，为了停止所有调度的周期数据标识符，可以不显示周期数据标识符，或者客户端可以明确的指定一个或多个周期数据标识符停止。

5.6.5.2.2 请求消息子功能参数\$Level (LEV_) 定义

该服务不使用子功能参数。

5.6.5.2.3 请求消息数据参数定义

表175定义了请求消息的数据参数。

表 175 请求消息数据参数定义

定义
传输模式 该参数标识服务端要使用的请求的周期数据标识符的传输速率。
周期数据标识符(#1 到 #m) 该参数标识了客户端正在请求的服务端数据记录（参数定义细节见C.1和以上服务描述）。应允许使用单个请求请求多个周期数据标识符。

5.6.5.3 肯定响应消息

5.6.5.3.1 肯定响应消息定义

必须区分初始肯定响应消息，该消息标明服务端接受服务和随后的包含周期数据标识符数据的周期性数据消息。

表176定义了服务端接受请求后要传输初始肯定响应消息。

表176肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	通过周期标识符读取数据响应 SID	M	0x6A	RDBP IPR

周期性数据标识符的数据以由请求的传输模式参数确定的速率周期性地（包含更新的数据）发送。

.在初始肯定响应之后，对于请求中的每个被支持的周期性数据标识符，服务端应开始发送单个周期性数据响应消息，定义如下。

表 177 定义了周期性数据响应消息数据的定义。

表 177 周期性数据响应消息数据定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	周期性数据标识符	M	0x00~0xFF	PDID
#2 ： #k+2	数据记录[]=[数据#1 ： 数据#k]	M ： U	0x00~0xFF ： 0x00~0xFF	DREC_ DATA_1 ： DATA_k

5. 6. 5. 3. 2 肯定响应消息数据参数定义

该服务不支持肯定响应消息中的响应消息数据参数。

表 178 定义了已定义的周期性消息的周期性消息数据参数。

表 178 周期性消息数据参数定义

定义
周期性数据标识符 该参数引用来请求消息的周期性数据标识符。
数据记录 通过周期性数据标识符读取数据肯定响应消息使用该参数向客户端提供所请求的数据记录值。本标准中未定义数据记录的内容，由车辆制造商指定。

5. 6. 5. 4 支持的否定响应码 (NRC_)

该服务应执行以下的否定响应码。表179记录了每个响应代码出现的情况。如果错误场景适用于服务端，则应使用列出的否定响应。

Table 179 — 否定响应码

NRC	描述	助记符
0x13	不正确的消息长度或无效格式 如果请求消息的长度无效或客户端超过了允许每次请求的周期数据标识符的最大数量，则应发送此 NRC。	IMLOIF
0x22	条件不正确 如果未满足服务端的操作条件，则该 NRC 将被发送，以执行所需的操作。例如，如果客户端请求具有不同传输模式的周期性数据标识符，并且服务端不能同时支持多个传输模式，就会出现这种情况。	CNC
0x31	请求范围超出 以下情况下，应发送该 NRC： ——设备不支持请求的周期性数据标识符的值； ——请求的动态定义数据标识符尚未被分配； ——客户端超过了允许同时调度的周期性数据标识符的最大数量。	ROOR
0x33	安全访问拒绝 如果至少一个周期性数据标识符加密且服务端未处于解锁状态，则应发送该 NRC。	SAD

Figure 18 — NRC handling for ReadDataByPeriodicIdentifier service

图 18 通过周期性标识符读取数据服务的 NRC 处理

5.6.5.5 通过周期性标识符读取数据的消息流示例

5.6.5.5.1 假设

以下示例展示了通过周期性标识符读取数据的行为。客户端可以在任何时候独立于服务端状态请求周期性数据标识符数据。

以下周期性数据标识符示例是针对动力系统设备的(例如，发动机控制模块)。

更多关于接收的术语/定义/排放相关系统的缩略语细节参考 ISO 15031 - 2° [6]。

5.6.5.5.2 示例#1-以中等速率读取多个周期性数据标识符 0xE3 和 0x24

5.6.5.5.2.1 假设

该示例演示了用单个请求请求多个数据标识符 (其中周期性数据标识符0xE3 (= 数据标识符 0xF2E3)包含发动机冷却剂温度、节流阀位置、发动机转速、车辆速度传感器；周期性数据标识符 0x24 (= dataIdentifier 0xF224)包含电池正电压、歧管绝对压力、质量空气流量、车辆气压和极端的负载值)。

客户端以中等速率请求传输，在获取周期数据一段时间后，客户端仅停止传输周期性数据标识符 0xE3。

5.6.5.5.2.2 步骤#1：请求周期性数据标识符的周期性传输

表180定义了通过周期性标识符读取数据请求消息流示例-步骤#1。

表 180 通过周期性标识符读取数据请求消息流示例-步骤#1

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 类型	描述 (所有值均为十六进制形式)	字节值	助记符
#1	通过周期性标识符读取数据请求 SID	0x2A	RDBPI
#2	传输模式 = 以中等速率发送	0x02	TM_SAMR
#3	周期性数据标识符#1	0xE3	PDID1
#4	周期性数据标识符#2	0x24	PDID2

表181定义了通过周期性标识符读取数据初始肯定响应消息流示例-步骤#1。

表 181 通过周期性标识符读取数据初始肯定响应消息流示例-步骤#1

消息方向	服务端 → 客户端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制形式)	字节值	助记符
#1	通过周期性标识符读取数据响应 SID	0x6A	RDBPIPR

表 182 定义了通过周期性标识符读取数随后的肯定响应消息流示例-步骤#1。

表 182 通过周期性标识符读取数随后的肯定响应消息流示例-步骤#1

消息方向	服务端 → 客户端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制形式)	字节值	助记符
#1	周期性数据标识符#1	0xE3	PDID1
#2	数据记录[数据#1] = ECT	0xA6	DREC_DATA_1
#3	数据记录[数据#2] = TP	0x66	DREC_DATA_2
#4	数据记录[数据#3] = RPM	0x07	DREC_DATA_3
#5	数据记录[数据#4] = RPM	0x50	DREC_DATA_4
#6	数据记录[数据#5] = VSS	0x00	DREC_DATA_5

表 183 定义了通过周期性标识符读取数随后的肯定响应消息#2 流示例-步骤#1。

表 183 通过周期性标识符读取数随后的肯定响应消息#2 流示例-步骤#1

Message	Server → client
---------	-----------------

direction 消息方向	服务端 → 客户端		
Message Type 消息类型	Response 响应		
A_Data Byte A_Data 字节	Description (all values are in hexadecimal) 描述 (所有值均为十六进制形式)	Byte Value 字节值	Mnemonic 助记符
#1	periodicDataIdentifier#2 周期性数据标识符#2	0x24	PDID2
#2	dataRecord [data#1] = B+ 数据记录[数据#1] = B+	0x8C	DREC_DATA_1
#3	dataRecord [data#2] = MAP 数据记录[数据#2] = MAP	0x20	DREC_DATA_2
#4	dataRecord [data#3] = MAF 数据记录[数据#3] = MAF	0x1A	DREC_DATA_3
#5	dataRecord [data#4] = BARO 数据记录[数据#4] = BARO	0x63	DREC_DATA_4
#6	dataRecord [data#5] = LOAD 数据记录[数据#5] = LOAD	0x4A	DREC_DATA_5

服务端以适用于服务端的中等速率发送上述所示的后续响应消息。

5.6.5.2.2.3 步骤#2:停止周期性数据标识符的传输

表184定义了通过标识符读取数据请求消息流示例-步骤#2。

表 184 通过标识符读取数据请求消息流示例-步骤#2

消息方向	客户端→服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制形式)	Byte Value 字节值	Mnemonic 助记符
#1	通过周期性标识符读取数据请求 SID	0x2A	RDBPI
#2	传输模式 = 停止发送	0x04	TM_SS
#3	周期性数据标识符#1	0xE3	PDID

表 185 定义了通过标识符读取数据肯定响应消息流示例-步骤#2。

表 185 通过标识符读取数据肯定响应消息流示例-步骤#2

服务端仅停止传输周期性数据标识符 0xE3。 周期性数据标识符 0x24 仍以服务端中速传输。

5.6.5.3 示

例#2-通过周期性标识符读取数据服务	消息方向	服务端 → 客户端		
	消息类型	响应		
	A_Data 字节	描述 (所有值均为十六进制形式)	字节值	助记符
	#1	通过周期性标识符读取数据响应 SID	0x6A	RDBPIPR

务周期调度速率的图形和表格示例

5.6.5.3.1 通过周期性标识符读取数据示例综述

本条包含被调度的周期数据的示例，包括通过周期性标识符读取数据（0x2A）服务的图形和表格示例。

该示例包含客户端和服务端应用程序之间传输的消息（请求/响应）的图形描述，随后是一个表，该表显示了服务端定期调度程序的可能实现，其变量以及每次更改检查定期调度程序是否已执行的后台功能的方式。

以下示例中，定义了下列应用：

——快速周期速率为25ms且中等周期速率为300ms

——每隔12.5ms检查一次周期性调度程序，这意味着使用此周期调用（轮询）周期性调度程序后台函数。每次调用后台周期性调度程序时，将遍历调度程序条目，直到发送单个周期性标识符，或者直到调度程序中的所有标识符都已被检查且没有一个准备好发送。

——可被同时调度的周期性数据标识符的最大数量为4。

——分配一个唯一的周期性数据响应消息地址信息ID。

由于周期性调度器轮询速率为 12.5ms，因此快速速率循环计数器将设置为 2 () 每次发送一个快速速率的数据

标识符，中等速率循环计数器将被重置为 24。

5.6.5.5.3.2 示例#2-以中等速率读取多个周期性数据标识符0xE3和0x24

在 $t = 0,0 \text{ ms}$ 时刻，客户端开始以中等速率(300 ms)将请求发送到调度 2 周期性数据标识符 (0xF2E3 和 0xF224)。出于本示例的目的，服务端在第一次 $t = 25,0\text{ms}$ 时接收请求并执行周期性调度程序后台函数。

表 19 示例#2-以中等速率调度周期性数据标识符

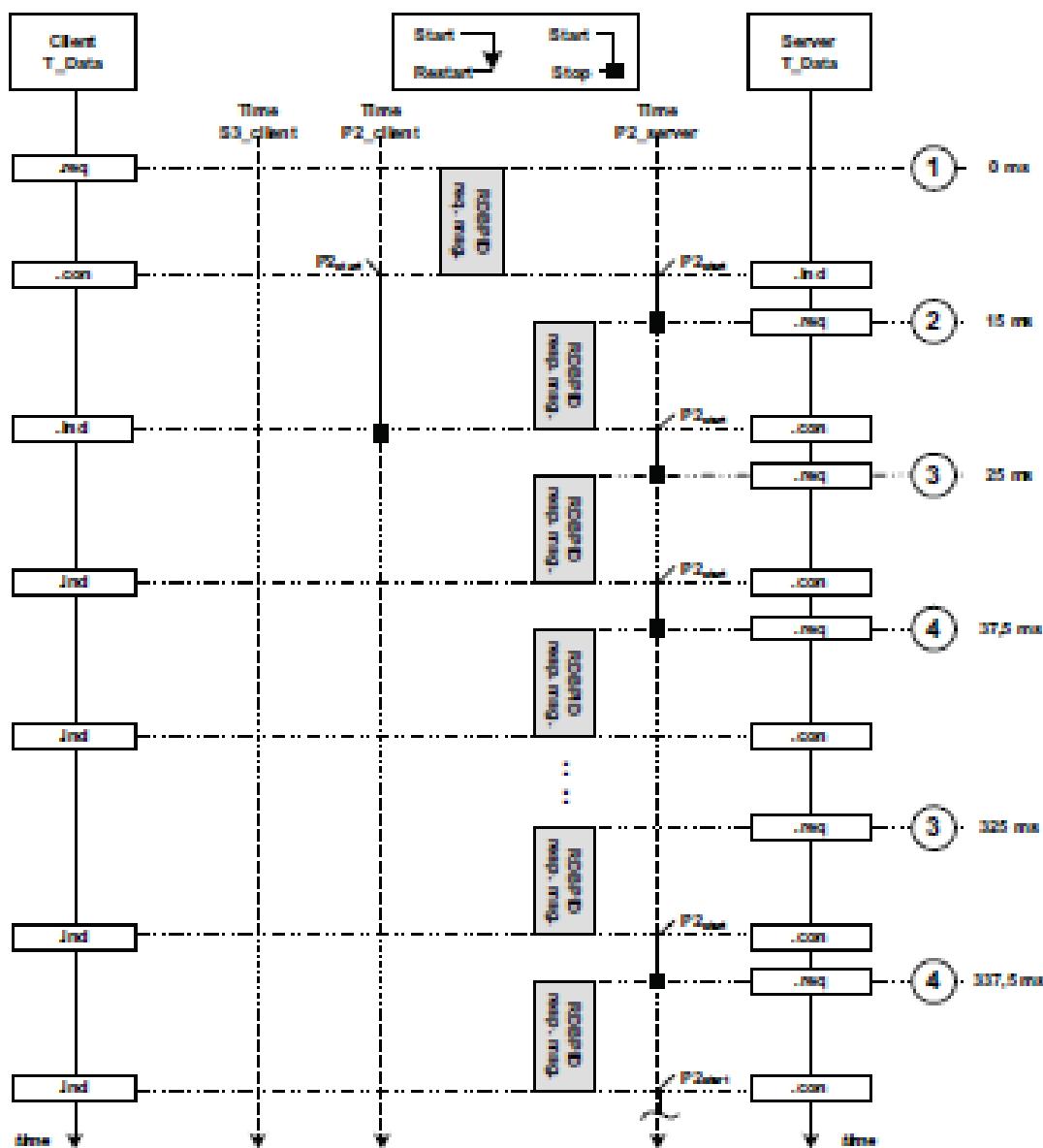


表 186 显示了服务端中的周期性调度程度的可能应用，该表显示了服务端定期调度程序的可能实现，其变量以及每次更改检查定期调度程序是否已执行的后台功能的方式。

表 186 示例#2: 周期性调度程度表

时间 t ms	周期性调 度程序传 输索引	发送的 周期性 标识符	周期性 调度程 序循环#	调度程 序[0]传 输计数	调度程 度[1]传 输计数
25, 0	0	0xE3	1	0->24	0
37, 5	1	0x24	2	23	0->24
50, 0	0	无	3	22	23
62, 5	0	无	4	21	22
75, 0	0	无	5	20	21
87, 5	0	无	6	19	20
100, 0	0	无	7	18	19

112, 5	0	无	8	17	18
125, 0	0	无	9	16	17
137, 5	0	无	10	15	16
150, 0	0	无	11	14	15
162, 5	0	无	12	13	14
175, 0	0	无	13	12	13
187, 5	0	无	14	11	12
200, 0	0	无	15	10	11
212, 5	0	无	16	9	10
225, 0	0	无	17	8	9
237, 5	0	无	18	7	8
250, 0	0	无	19	6	7
262, 5	0	无	20	5	6
275, 0	0	无	21	4	5
287, 5	0	无	22	3	4
300, 0	0	无	23	2	3
312, 5	0	无	24	1	2
325, 0	0	0xE3	25	0->24	1
337, 5	1	0x24	26	23	0->24
350, 0	0	无	27	22	23
362, 5	0	无	28	21	22

5.6.6 动态定义数据标识符 (0x2C) 服务

5.6.6.1 服务描述

动态定义数据标识符服务允许客户端在服务端中动态定义一个数据标识符，随后可以通过该服务读取该数据标识符。

此服务的目的是为客户端提供将一个或多个数据元素集合到一个数据超集的能力，该数据超集可以通过“通过标识符读取数据”或“通过周期性标识符读取数据”服务进行请求。集合在一起的数据元素可通过以下方式被引用：

- 源数据标识符，位置和大小；或

- 内存地址和内存大小；或

——上述列出的两个方法的组合，使用多个请求来定义单个数据元素。动态定义的数据标识符将包含数据参数定义的串联。

该服务允许更灵活地处理诊断应用程序的特殊数据需求，该应用程序超出了可以通过静态定义的数据标识符读取的信息，还可以避免经常请求/响应事务关联，来减少带宽利用率。

动态定义的数据标识符的定义可以通过单个请求消息或多个请求消息来完成。这允许定义引用源标识符和内存地址的单个数据元素。服务端应串联单个数据元素的定义。通过清除当前定义并从新的定义开始，可以实现动态定义的数据标识符的重新定义。当多个动态定义数据标识符请求消息用于配置单个数据标识符并且在后续请求该数据标识符时服务端检测到最大字节数溢出（例如周期性数据标识符的定义），服务端应该保留数据标识符的定义，因为它在请求之前，否则会导致溢出。

尽管该服务不禁止此类功能，但不建议客户端动态定义的数据记录之间相互引用，因为删除被引用的记录可能会在引用记录中产生数据一致性问题。

该服务还提供了清除现有动态定义数据记录的能力。如果指定的数据记录标识符在服务端支持的有效动态数据标识符范围内，则清除数据记录的请求应得到肯定响应。

服务端应维护动态定义的数据记录，直到它被清除或由车辆制造商指定（例如在会话转换或服务端断电时删除动态定义的数据记录）。

服务端可以通过两种不同的方式实现数据记录：

- 包含多个元素数据记录的复合数据记录，这些元素数据记录不会单独被引用。

——唯一的 2 字节标识“标记”或数据标识符 (DID) 值用于服务端中支持的单个元素数据记录(元素数据记录或 DID 的示例是发动机转速或进气温度)。该数据记录实现是复合数据记录实现的子集，因为它只引用单个元素数据记录，而不引用包含多个元素数据记录的复合数据记录。

动态定义数据标识符服务支持两种实现数据记录的类型，以定义动态数据标识符

——复合的数据块：位置参数必须引用复合数据块中的起始点，大小参数必须反映被放置在动态定义的数据标识符中的数据的长度。诊断设备不只是在动态数据记录中包含复合数据块的元素数据记录的一部分。

——2字节 DID：位置参数必须设置为 1，大小参数必须反映 DID 的长度(元素数据记录的长度)。诊断仪负责在动态数据记录中不包含 2 字节的 DID 值的一部分。

在动态定义的数据记录中，数据的顺序应与客户端请求消息中指定的顺序相同。同时，客户端的请求中指定的数据的第一个位置应是确定的，以使它应该按照前一句中提到的顺序要求，在最接近动态数据记录开始的地方出现。

除了通过逻辑引用(记录数据标识符)定义动态数据标识符之外，该服务还提供了通过绝对内存地址和内存长度信息定义动态定义的数据标识符的功能。这种定义动态数据标识符的机制建议只在服务端的开发阶段使用。

应注意，服务端和客户端应满足7.5中指定的请求和响应消息行为。

5. 6. 6. 2 请求消息

5. 6. 6. 2. 1 请求消息定义

表190定义了请求消息-子功能 = 通过标识符定义。

表 190 请求消息-子功能 = 通过标识符定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	动态定义数据标识符请求 SID	M	0x2C	DDDI
#2	子功能=[定义类型 = 通过标识符定义]	M	0x01	LEV_DBID
#3 #4	动态定义数据标识符[] = [字节#1 (MSB) 字节#2 (LSB)]	M M	0xF2 / 0xF3 0x00 - 0xFF	DDDDI_HB LB
#5 #6	源数据标识符[] #1 = [字节#1 (MSB) 字节#2 (LSB)]	M M	0x00 - 0xFF 0x00 - 0xFF	SDI_HB LB
#7	源数据记录中的位置#1	M	0x01 - 0xFF	PISDR#1
#8	内存大小#1	M	0x00 - 0xFF	MS#1
:	:	:	:	:
#n-3 #n-2	源数据标识符[] #m 字节#1 (MSB) 字节#2 (LSB)]	U U	0x00 - 0xFF 0x00 - 0xFF	SDI_HB LB
#n-1	源数据记录位置#m	U	0x01 - 0xFF	PISDR#m
#n	内存大小#m	U	0x00 - 0xFF	MS#m

表 191 定义了请求消息 -子功能 = 通过内存地址定义

表 191-请求消息 - 子功能 = 通过内存地址定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	动态定义数据标识符请求 SID	M	0x2C	DDDI
#2	子功能 = [定义类型 = 通过内存地址定义]	M	0x02	LEV_DBMA
#3 #4	动态定义数据标识符[] =[字节#1 (MSB) 字节#2 (LSB)]	M M	0xF2 / 0xF3 0x00 - 0xFF	DDDDI_HB LB
#5	地址和长度格式标识符	M1	0x00 - 0xFF	ALFID
#6 :(m-1)+6	内存地址[] = [字节#1 (MSB) : 字节#m]	M : C1	0x00 - 0xFF : 0x00 - 0xFF	MA_B1 : Bm
#m+6 :(m+6+(k-1))	内存大小[] = [字节#1 (MSB) : 字节#k]	M : C2	0x00 - 0xFF : 0x00 - 0xFF	MS_B1 : Bk
:	:	:	:	:

#n-k-(m-1) : #n-k	内存地址[] = [字节#1 (MSB) : 字节#m]	U : U/C1	0x00 - 0xFF : 0x00 - 0xFF	MA_B1 : Bm
#n-(k-1) : #n	内存大小[] = [字节#1 (MSB) : 字节#k]	U : U/C2	0x00 - 0xFF : 0x00 - 0xFF	MS_B1 : Bk
M1: 地址和长度格式标识符参数仅在请求消息的开头出现一次，并在整个请求消息中定义每个内存位置引用的地址长度和长度信息。 C1: 该参数的存在取决于地址长度信息参数的地址长度和长度。 C2: 该参数的存在取决于地址和长度标识符的内存大小信息。				

表192 定义了请求消息-子功能 = 清除动态定义的数据标识符

表 192 请求消息-子功能 = 清除动态定义的数据标识符

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	动态定义数据标识符请求 SID	M	0x2C	DDDI
#2	子功能 = [定义类型 = 清除动态定义的数据标识符]	M	0x03	LEV_CDD DID
#3 #4	动态定义数据标识符[] =[字节#1 (MSB) 字节#2 (LSB)]	M M	0xF2 / 0xF3 0x00 - 0xFF	DDDDI_ HB LB
C: 该参数的存在需要服务端清除字节#1 中包含的动态定义的数据标识符和字节# 2。如果该参数不存在，服务端中所有动态定义的数据标识符都将被清除。				

5.6.6.2.2 请求消息子功能参数\$Level (LEV_)定义

表193为该服务的请求消息定义为有效的子参数(未显示抑制肯定响应位 (位7))。

表193-请求消息子功能参数定义

Bits6-0	描述	Cvt	助记符
0x00	ISOSAEReserved 该值为本文件为未来定义预留。	M	ISOSAERESRVD
0x01	由标识符定义 该参数用于向服务端说明动态数据标识符的定义应通过数据标识符引用来进行。	M	DBID
0x02	由内存地址定义 该参数用于向服务端说明动态数据标识符的定义应通过内存地址引用来进行。	U	DBMA
0x03	清除动态定义的数据标识符 该参数用于清除指定的动态数据标识符。请注意，服务端应肯定响应来自客户端的明确请求，即使请求时不存在指定的动态数据标识符。 但是，指定的动态数据标识符必须在有效范围内(允许范围见 C.1)。如果在请求时定期报告指定的动态数据标识符，则应首先停止该动态标识符，然后清除该动态标识符。	U	CDDDI
0x04-0x7F	ISOSAEReserved 该值为未来定义预留。	M	ISOSAERESRVD

5.6.6.2.3 10.6.2.3 请求消息数据参数定义

表 194 定义了请求消息的数据参数。

表 194 请求消息数据参数定义

定义
动态定义数据标识符 该参数指定在将来对通过标识符读取数据服务或通过周期性数据标识符读取数据服务的调用中如何引用由客户端定义的动态数据记录。动态定义的数据标识符应在通过标识符读取数据服务中作为数据标识符处理(详见C.1)。在通过周期性数据标识符读取数据服务中，应作为周期性记录标识符处理(有关此参数值的要求，请参见通过周期性数据标识符读取数据服务，以便能定期请求动态定义的数据标识符)。

源数据标识符
该参数仅用于子功能=由标识符定义。此参数逻辑上指定要包含到动态数据记录中的信息源。例如，这可以是用于引用引擎速度的2字节DID，或用于引用包含引擎速度、车辆速度、进气空气温度等信息的复合块的2字节DID(请参见C.1了解更多细节)。
源数据标识符中的位置
该参数仅用于子功能=由标识符定义。该1字节参数用于指定要包含在动态数据记录中的源数据记录的摘录的起始字节位置。一个位置应该引用源数据标识符引用的数据记录的第一个字节。
地址和长度格式标识符
该参数是一个1字节的值，每个半字节单独编码(例如，参见H.1): Bit 7- 4: 内存大小参数的长度(字节数)； Bit 3 -0: 内存地址参数的长度(字节数)。
内存地址
该参数仅用于子功能=由内存地址定义。此参数指定内存源要包含到动态数据记录中的信息地址。用于此地址的字节数由地址和长度格式标识符的低半字节(bit3 - 0)定义。内存地址参数中的字节#m总是服务端中引用的地址中的最高有效字节。地址中最高有效字节可以用作内存标识符。
内存大小
该参数用于指定将包含在动态数据记录中的源数据记录/内存地址的总字节数。 对于子功能=由标识符定义，源数据记录中的位置参数也用于指定源数据标识符的起始位置，内存大小从该位置应用。内存大小使用的字节数为一个字节。 对于子功能=由内存地址定义，该参数反应了起始于指定的内存地址的动态定义的数据标识符中包含的字节数。用于该大小的字节数由地址和长度格式标识符的高半字节定义 (bit 7 – 4)。

5.6.6.3 肯定响应消息

5.6.6.3.1 肯定响应消息定义

表 195 定义了肯定响应消息。

表 195 肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	动态定义数据标识符请求 SID	M	0x6C	DDDI
#2	子功能 = [定义类型]	M	0x00-0x7F	DM
#3 #4	动态定义数据标识符[] =[字节#1 (MSB) 字节#2 (LSB)]	M M	0xF2 / 0xF3 0x00 - 0xFF	DDDDI_- HB LB
C: 如果请求消息中存在动态定义的数据标识符参数，则需要存在该参数，否则不应包含该参数。				

5.6.6.3.2 肯定响应消息数据参数定义

表 196 定义了肯定响应消息的数据参数。

定义
定义类型 该参数与来自请求消息的子功能参数的位6-0相同。
动态定义的数据标识符 该参数与来自请求消息的数据参数动态定义的数据标识符相同。

5.6.6.4 支持的否定响应码 (NRC_)

该服务应执行以下否定响应码。表197记录了每个响应码发生的场景。如果错误场景应用于服务端，则应使用列出的否定响应码。

NRC	描述	助记符
0x12	不支持该子功能 如果不支持子功能参数，则应发送此 NRC。	SFNS

0x13	不正确的消息长度或无效格式 如果请求消息的长度无效或客户端超过了允许每次请求的周期数据标识符的最大数量，则应发送此 NRC。	IMLOIF
0x22	条件不正确 如果未满足服务端的操作条件，则该 NRC 将被发送，以执行所需的操作。	CNC
0x31	请求范围超出 以下情况下，应发送该 NRC： ——设备不支持请求的周期性数据标识符的值； ——请求的动态定义数据标识符尚未被分配； ——客户端超过了允许同时调度的周期性数据标识符的最大数量。	ROOR
0x33	安全访问拒绝 如果至少一个周期性数据标识符加密且服务端未处于解锁状态，则应发送该 NRC。	SAD

5.7 存储数据传输功能单元

5.7.1 概述

表 249 存储数据传输功能单元

服务	描述
清除诊断信息	允许客户端从服务端清除诊断信息（包括 DTC、捕获的数据等）。
读取 DTC 信息	允许客户端从服务端请求诊断信息（包括 DTC、捕获的数据等）。

5.7.2 清除诊断信息 (0x14) 服务

5.7.2.1 服务描述

客户端使用清除诊断信息服务来清除一个或多个服务端内存中的诊断信息。

当清除诊断信息服务被完全处理后，服务端应发送一个肯定响应。即使没有存储 DTC，服务端也应发送肯定响应。如果服务端支持内存中 DTC 状态信息的多个副本（例如，RAM 中的一个副本和 EEPROM 中的一个副本），则服务端应清除读取 DTC 信息状态报告服务使用的副本。根据适当的备份策略（例如在电源锁存阶段）更新其他副本，如长期存储器中的备份副本。

注：在电源锁存阶段受到干扰的情况下（例如，在电源锁存阶段电池断开），这可能导致数据不一致。

各个 DTC 状态位的行为应根据 D.2, D.1-D.8 中的定义来实现。

客户端的请求消息包含一个参数。参数 groupOfDTC 允许客户端清除一组 DTC（例如，动力、车身、底盘等）或特定的 DTC。详情参见 D.1。除非另有说明，否则服务端应从存储器中清除所请求组的与排放有关的和与排放无关的 DTC 信息。

通过该服务重置/清除的 DTC 信息包括但不限于：

- ↓ ——DTC 状态字节（参见 11.3 中的读取 DTC 信息服务）；
- ↓ ——捕获的 DTC 快照数据（DTCSnapshotData，参见 11.3 中的读取 DTC 信息服务）；
- ↓ ——捕获的 DTC 扩展数据（DTCExtendedData，参见 11.3 中的读取 DTC 信息服务）；
- ↓ ——其他与 DTC 相关的数据，例如 DTC 特有的最早/最近 DTC、标志、计数器、定时器等。

存储在服务端中可选可用的 DTC 镜像内存中的任何 DTC 信息不受该服务的影响（有关 DTC 镜像内存定义，参见 11.3 中的读取 DTC 信息(0x19)服务）。

a) **重要**—服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

5.7.2.2 请求消息

5.7.2.2.1 请求消息定义

表 250 定义了请求消息。

表 250 请求消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	清除诊断信息请求 SID	M	0x14	CDTCI
#2	groupofdtc [] = [Groupofdtc 高字节	M	0x00 - 0xff	GODTC_
#3	groupofdtc 中字节	M	0x00 - 0xff	HB
#4	groupofdtc 低字节]	M	0x00 - 0xff	MB
				LB

5.7.2.2.2 请求消息子功能参数\$Lead (LeVig) 定义

该服务没有使用子功能参数。

5.7.2.2.3 请求消息数据参数定义

表 251 定义了请求消息的数据参数。

定义
groupOfDTC
此参数包含一个 3 字节的值，指示要清除的 DTC 组（例如，动力、车身、底盘）或特定 DTC。值/范围的值的定义包括在 D.1 中。

表 251 请求消息数据参数定义

5.7.2.3 肯定响应信息

5.7.2.3.1 肯定响应消息定义

表 252 定义了肯定响应消息。

表 252 肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	清除诊断信息肯定响应 SID	M	0x54	CDTCIPR

5.7.2.3.2 肯定响应消息数据参数定义

该服务在肯定响应消息中不使用数据参数。

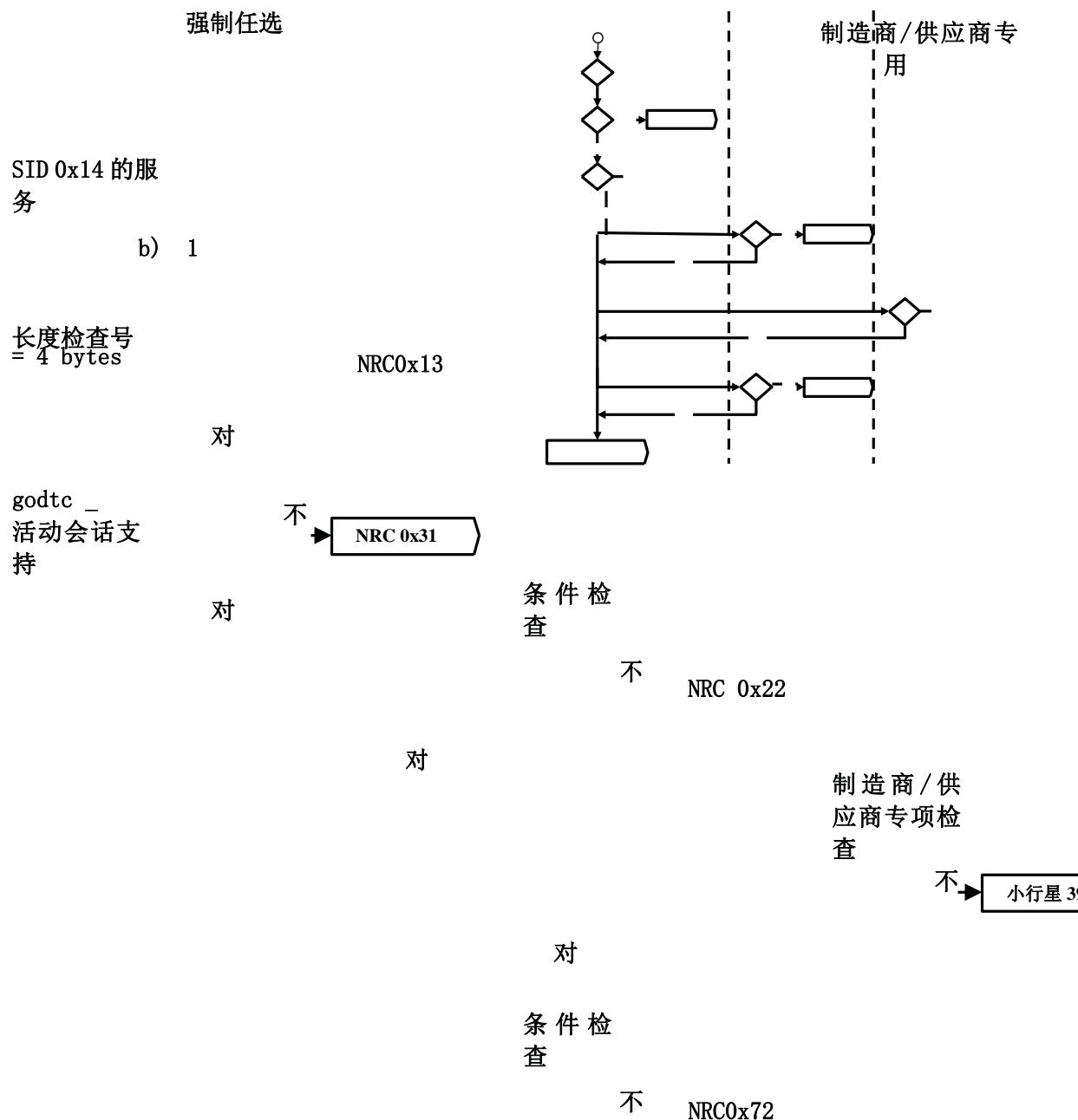
5.7.2.4 支持的否定响应码 (NRCI)

该服务应实施下列否定响应代码。每个响应代码发生的环境记录在表 253 中。如果错误场景应用于服务端，则应使用列出的否定响应。

表 253 支持的否定响应码

NRC	描述	助记符
0x13	不正确消息格式无效格式 如果消息的长度错误，则应发送此 NRC。	IMLOIF
0x22	条件不正确 如果服务端内部条件阻止清除存储在服务端中的 DTC 相关信息，则应使用本 NRC。	CNC
0x31	请求范围 如果不支持指定的 groupOfDTC 参数，则应返回此 NRC。	ROOR
0x72	通用编程失败 如果服务端在写入存储器位置时检测到错误，则应返回此 NRC。	GPF

评估顺序记录在图 23 中。



对

积极反应

钥匙

1 cdtci + godtc _

c) 图 23 为 CysRealLogic 信息服务的 NRC 处理

5.7.2.5 清除诊断信息消息流示例

客户端向单个服务端发送清除诊断信息请求消息。表 254 定义了清除诊断信息请求消息流示例#1。客户端向单个服务端发送清除诊断信息请求消息。

表 254 - 清除诊断信息请求消息流示例 1

消息方向	客户端 → 服务端		
消息类型	请求		
A_Data 字节_	描述 (所有值均为十六进制)	字节值	助记符
#1	清除诊断信息请求 SID	0x14	CDTCI
#2	group0fDTC [DTC 高字节] (“排放相关系统”)	0xff	DTCHB
#3	group0fDTC [DTC 中字节]	0xff	DTCMB
#4	group0fDTC [DTC 低字节]	0x33	DTCLB

表 255 定义了清除诊断信息肯定响应消息流示例#1。

表 255-清除诊断信息肯定响应消息流示例#1

消息方向	客户端 → 服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	清除诊断信息响应 SID	0x54	CDTCIPR

5.7.3 读写 DTC 信息 (0x19) 服务

5.7.3.1 服务描述

5.7.3.1.1 一般描述

该服务允许客户端从车辆内的任一服务端或服务端组读取服务端驻留诊断故障代码(DTC)信息的状态。除非特定子功能另有要求，服务端应返回所有 DTC 信息(例如，与排放相关和与排放无关)。该服务允许客户端执行以下操作：

- 检索与客户端定义的 DTC 状态掩码匹配的 DTC 数目；
- 检索与客户端定义的 DTC 状态掩码匹配的所有 DTC 列表；
- 检索与客户端定义的 DTC 状态掩码匹配的特定功能组内的 DTC 列表；
- 用“永久 DTC”状态检索所有 DTC。

——检索与客户端定义的 DTC 相关联的 DTCSnapshot 数据(有时称为冻结帧)：DTC 快照是与 DTC 相关联的特定数据记录，存储在服务端的内存中。DTC 快照的典型用法是在检测到系统故障时存储数据。DTC 快照将充当系统故障发生时的数据值的快照。存储在 DTC 快照中的数据参数应与 DTC 相关联。DTC 特定的数据参数旨在减轻技术人员进行故障隔离的过程。

——从 DTC 内存或 DTC 镜像内存中检索与客户端定义的 DTC 和状态掩码组合相关联的 DTCExtendedData。DTCExtendedData 数据由与 DTC 相关的扩展状态信息组成。DTCExtendedData 包含 DTC 参数值，这些值在请求时已经标识。DTCExtendedData 的一个典型用途是存储与 DTC 相关联的动态数据，例如：

- DTC B1 故障指示计数器，表示在 OBD 系统运行而故障处于活动状态的时间量(发动机运行小时数)；
- DTC 发生计数器，计数已报告“testFailed”的驾驶循环次数；
- DTC 老化计数器，计数自最近一次失效以来的驾驶循环次数，不包括测试没有报告“testPassed”或“testFailed”的驾驶循环；
- 用于 OBD 的特定计数器(例如，如果在无故障模式下可以执行驾驶循环，则直到“检查发动机”灯被关闭为止，剩余的驾驶循环的数量)；
- 上次发生的时间(等)；
- 测试失败计数器，如果在几个步骤中进行验证，则计数报告的“testFailed”的数量和可能的其他计数器的数量；
- 未完成的测试计数器，计数自最近一次测试完成以来的驾驶循环次数(即，由于测试报告“testPassed”或“testFailed”)。
- 检索与客户端定义的严重性掩码匹配的 DTC 的数目；
- 检索与客户端定义的严重性掩码记录匹配的 DTC 列表；
- 检索客户端定义的 DTC 的严重性信息；
- 检索服务端支持的所有 DTC 的状态；
- 检索服务端失败的第一个 DTC；
- 检索服务端最近确认的 DTC；
- 检索服务端确认的第一个 DTC；
- 检索服务端中最近确认的 DTC；
- 从与客户端定义的 DTC 状态掩码匹配的 DTC 镜像存储器中检索 DTC 列表；
- 从 DTC 镜像存储器中检索客户端定义的 DTC 掩码和客户端定义的 DTCExtendedData 记录编号的镜像存储器 DTCExtendedData 记录数据，

从与客户机定义的 DTC 状态掩码匹配的 DTC 镜像存储器中检索 DTC 的数量，

检索与客户端定义的 DTC 状态掩码匹配的“仅”排放相关的 OBD DTC 的数量。与排放相关的 OBD DTC 在检测到这种 DTC 时使故障指示器打开/显示消息，

- ↓ 检索与客户端定义的 DTC 状态掩码匹配的“仅”排放相关的 OBD DTC 的状态。与排放相关的 OBD DTC 在检测到这种 DTC 时使故障指示器打开/显示消息，
- ↓ 检索所有当前已检测到或尚未检测到“挂起”或“确认”的 DTC，
- ↓ 从 DTC 内存中检索与客户端定义的 DTCExtendedData 记录状态相关联的 DTCExtendedData。

- ↓ 从与客户端定义的 DTC 状态掩码匹配的用户定义的 DTC 内存中检索 DTC 列表，
- ↓ 检索用户定义的 DTC 存储器 DTCExtendedData 记录数据，用于客户端定义的 DTC 掩码掩码和客户端定义的 DTCExtendedData 记录用户定义的 DTC 镜像存储器中的编号，
- ↓ 从用户定义的 DTC 内存中检索客户端定义的 DTC 掩码的用户定义的 DTC 内存 DTCSnapshotRecord 数据，

该服务使用一个子功能来确定客户机正在请求哪种类型的诊断信息。关于每个子功能参数的进一步细节在下面的子句中提供。

这项服务使用下列术语：

- ↓ 启用标准：服务端/车辆制造商/系统供应商特定的标准，用于控制服务端何时实际执行特定的内部诊断。

↓ 测试通过标准：服务端/车辆制造商/系统供应商的特定条件，用于定义被诊断系统是否在正常、可接受的操作范围内正常工作(例如，不存在故障，并且诊断系统被分类为“OK”)。

↓ 测试失败标准：服务端/车辆制造商/系统供应商特定的故障条件，用于定义被诊断的系统是否已经通过测试。

↓ 确认故障标准：服务端/车辆制造商/系统供应商特定的故障条件，这些故障条件定义被诊断的系统是否确实有问题（确认），保证将 DTC 记录存储在长期存储器中。

↓ 发生计数器：由某些服务端维护的计数器，用于记录给定 DTC 测试报告测试失败唯一发生的实例数。

↓ 老化：某些服务端评估每个内部诊断的过去结果以确定是否可以从长期记忆中清除已确认的 DTC 的过程，例如，在校准了无故障周期的次数的情况下。

一个给定的 DTC 值（例如，0x080511）在对 readDTCIInformation 的肯定响应中不应被报告一次以上，除非读取 DTCSnapshotRecords，其中响应可能包含用于同一 DTC 的多个 DTCSnapshotRecords。

当使用分页缓冲区处理来读取 DTC 时（特别是对于子功能=reportDTCByStatusMask），在创建响应时可能会减少 DTC 的数量。在这种情况下，应答应填写 DTC 0x000000 和 DTC 状态 0x00。客户端应将这些 DTC 视为不存在于响应消息中。

d) 重要-服务端和客户端应满足 7.5 中规定的请求和响应消息行为。

5.7.3.1.2 检索与客户端定义的状态掩码匹配的 DTC 的数量（子功能=0x01reportNumberOfDTCByStatusMask）

客户端可以通过发送具有 reportNumberOfDTCByStatusMask 的子功能集的该服务的请求来检索与客户端定义的状态掩码匹配的 DTC 数量的计数。对该请求的响应包含 DTCSstatusAvailabilityMask，它提供服务端为屏蔽目的而支持的 DTC 状态位的指示。在 DTCSstatusAvailabilityMask 之后，响应包含 DTCFormat.er，它报告关于 DTC 格式和编码的信息。DTCFormat.er 后面跟着 DTCCount 参数，它是一个 2 字节的无符号数字，包含服务端内存中可用的 DTC 的数量，基于客户端提供的状态掩码。

子功能 reportNumberOfMirrorMemoryDTCByStatusMask 具有与子功能 reportNumberOfDTCByStatusMask 相同的功能，不同之处在于它从 DTC 镜像内存返回 DTC 的数量。

5.7.3.1.3 检索与客户端定义的状态掩码匹配的 DTC 列表（子功能=0x02reportDTCByStatusMask）

客户端可以检索 DTC 列表，该列表通过发送具有设置为 reportDTCByStatusMask 的子功能字节的请求来满足客户端定义的状态掩码。这个子功能允许客户机请求服务端报告所有“testFailed”或“.ed”OR 等的 DTC。

评估应按以下步骤进行：服务端应在客户端请求中指定的掩码与服务端支持的每个 DTC 相关的实际状态之间执行按位逻辑.-ing 操作。除了 DTCSstatusAvailabilityMask 之外，服务端还应返回 AND 操作的结果非零的所有 DTC（即 (statusOfDTC 和 DTCSstatusMask) != 0）。如果客户端指定包含服务端不支持的位的状态掩码，则服务端应仅使用其支持的位来处理 DTC 信息。如果服务端中的 DTCS 没有匹配掩蔽准则

在客户端的请求中指定，在肯定响应消息中的 DTCStatusAvailabilityMask 字节之后，不应提供 DTC 或状态信息。

. DTC 状态信息应根据来自客户端的成功清除诊断信息请求而被清除（有关在服务端中接收清除诊断信息服务请求的情况下的 DTC 状态比特处理的进一步描述，参见 D.2 中的 DTC 状态比特定义）。

5.7.3.1.4 检索 DTCSnapshot 记录标识（子功能=0x03reportDTCSnapshot.cation）

客户机可以通过发送对该服务的请求以及设置为 reportDTCSnapshot.cation 的子功能来检索所有捕获的 DTCSnapshot 记录的 DTCSnapshot 记录标识信息。服务端应返回所有存储的 DTCSnapshot 记录的 DTCSnapshot 记录标识信息列表。服务端在单个 DTCSnapshot 记录的响应消息中放置的每个项目都应包含 DTCRecord（包含 DTC 编号（高、中、低字节））和 DTCSnapshot 记录编号。如果为单个 DTC 存储了多个 DTCSnapshot 记录，则服务端应为每次出现在响应中放置一个项目，为每次出现使用不同的 DTCSnapshot 记录编号（用于以后检索记录数据）。

注意：服务端可以支持为单个 DTC 存储多个 DTCSnapshot 记录，以跟踪 DTC 每次出现时的状态。此功能的支持、“发生”标准的定义以及要支持的 DTCSnapshot 记录的数量应由系统供应商/车辆制造商定义。

DTCSnapshot 记录标识信息应根据来自客户的成功清除诊断信息请求进行清除。车辆制造商有责任规定在存储器溢出（存储 DTC 的存储器空间和 DTCSnapshot 数据完全占用在服务端中）的情况下删除存储的 DTC 和 DTCSnapshot 数据的规则。

5.7.3.1.5 为客户端定义的 DTC 掩码检索 DTCSnapshot 记录数据（子功能=0x04reportDTCSnapshotRecordByDTCNumber）

客户端可以通过发送对该服务的请求，并将子功能设置为 reportDTCSnapshotRecordByDTCNumber，来为结合 DTCSnapshot 记录号定义的 DTCMaskRecord 的客户端检索捕获的 DTCSnapshot 记录数据。服务端应搜索其支持的 DTC，以便与客户端指定的 DTCMaskRecord（包含 DTC 编号（高、中、低字节））精确匹配。客户机请求中提供的 DTCSnapshotRecordNumber 参数应指定请求 DTCSnapshot 记录数据的指定 DTC 的特定事件。

注意 1 DTCSnapshotRecordNumber 不与 DTCStoredDataRecordNumber 共享相同的地址空间。

如果服务端支持为单个 DTC 存储多个 DTCSnapshot 记录的能力（系统供应商/车辆制造商将定义此功能的支持）。然后建议服务端还实现 reportDTCSnapshot.cation 子功能参数。建议客户机在通过 reportDTCSnapshotRecordByDTCNumber 请求特定 DTCSnapshotRecordNumber 之前，首先请求标识使用子功能参数 reportDTCSnapshot.cation 存储的 DTCSnapshot 记录。

. 还建议支持子功能参数 reportDTCSnapshotRecord.cation，以便让客户端有机会直接识别存储的 DTCSnapshot 记录，而不是通过服务端上所有存储的 DTC 进行解析以确定是否存在 DTCSnap。镜头记录被存储。

系统供应商/车辆制造商有责任定义在这些服务端中捕获的 DTCSnapshot 记录是否存储与 ECU 文档中的故障发生信息相关的数据。

与 DTC 号码和状态 0fDTC 一起，服务端应响应客户端的请求，返回单个预定义的 DTCSnapshotRecord。如果已经为客户端定义的 DTCMaskRecord 和 DTCSnapshotRecordNumber 参数（DTCSnapshotRecordNumber 不等于 0xFF）识别出故障。

注 2. 准确的失效准则应由系统供应商/车辆制造商确定。

DTCSnapshot 记录可以包含多个数据参数，这些数据参数可用于在故障发生时重构车辆状况（例如，B+、RPM、时间戳）。

车辆制造商应规定 DTCSnapshotRecord 的格式和内容。DTCSnapshotRecord 中报告的数据首先包含一个 data.er，用于标识以下数据。这种数据标识符/数据组合可以在 DTCSnapshotRecord 中重复。DTCSnapshotRecord 中的一个或多个数据标识符的使用允许针对单个 DTC 存储不同类型的数据。指示每个 DTCSnapshotRecord 中包含的记录数据标识符数量的参数应随每个 DTCSnapshotRecord 一起提供，以帮助数据检索。

服务端应在单个响应消息中报告一个 DTCSnapshot 记录，除非客户端已经将 DTCSnapshotRecordNumber 设置为 0xFF。因为这将导致服务端在单个响应消息中使用为客户机定义的 DTCMaskRecord 存储的所有 DTCSnapshot 记录进行响应。DTCANSTATUS 记录只包含在响应消息中的一次。如果客户机在其请求中对参数 DTCSnapshotRecordNumber 使用 0xFF，则服务端应按数字升序报告为特定 DTC 捕获的所有 DTCSnapshot 记录。

如果客户端指定的 DTCMaskRecord 或 DTCSnapshotRecordNumber 参数无效或服务端不支持，则服务端将作出否定响应。这与客户机指定的 DTCMaskRecord 和/或 DTCSnapshotRecordNumber 参数确实有效并由服务端支持，但没有与其相关联的 DTCSnapshot 数据的情况不同（例如因为指定的 DTC 或记录号从未发生过故障事件。服务端应发送仅包含 DTCAndStatusRecord（请求的 DTC 编号（高、中、低字节）以及 statusOfDTC 的回声）的肯定响应。

DTCSnapshot 信息应根据来自客户端的成功清除诊断信息请求进行清除。在存储器溢出（存储 DTC 的存储器空间和 DTC 快照数据完全占用在服务端中）的情况下，车辆制造商有责任指定删除存储的 DTC 和 DTCSnapshot 数据的规则。

5.7.3.1.6 为客户端定义的记录编号检索 DTCSavedData 记录数据（子功能=0x05 reportDTCSavedDataByRecordNumber）

客户端可以通过发送对该服务的请求，并将子功能设置为 reportDTCSavedDataByRecordNumber，来检索 DTCSavedDataRecordNumber 的捕获 DTCSavedData 记录数据。服务端应搜索其存储的 DTCSavedData 记录，以查找客户机提供的记录编号的匹配。

DTCSavedDataRecordNumber 不与 DTCSnapshotRecordNumber 共享相同的地址空间。

系统供应商/车辆制造商有责任确定在这些服务端中捕获的 DTCSavedData 记录是否存储与首次或最近发生的故障相关的数据。

注意：准确的失效标准应由系统供应商/车辆制造商确定。

DTCSavedData 记录可以包含多个数据参数，这些数据参数可用于在故障发生时重构车辆状况（例如，B+、RPM、时间戳）。

车辆制造商应定义 DTCSavedDataRecordNumber 的格式和内容。DTCSavedDataRecord 中报告的数据首先包含一个 data.er，用于标识以下数据。此数据标识符/数据组合可以在 DTCSavedDataRecord 中重复。在 DTCSavedDataRecord 中使用一个或多个数据标识符允许针对单个 DTC 存储不同类型的数据。指示每个 DTCSavedDataRecord 中包含的记录 Data.er 数量的参数应当与每个 DTCSavedDataRecord 一起提供，以帮助数据检索。

服务端应在单个响应消息中报告一个 DTCStoredDataRecord，除非客户机已将 DTCStoredDataRecordNumber 设置为 0xFF，因为这将使服务端使用存储在单个响应消息中的所有 DTCStoredDataRecords 进行响应。

如果客户机请求按记录编号报告所有 DTCStoredDataRecords，则必须在每个存储的 DTCStoredDataRecord 的响应消息中重复 DTCAndStatusRecord。

如果客户端指定的 DTCStoredDataRecordNumber 参数无效或服务端不支持，则服务端将作出否定响应。这与客户机指定的 DTCStoredDataRecordNumber 参数确实有效并由服务端支持，但没有与其相关联的 DTCStoredDataRecord 数据的情况不同（例如因为指定的记录号从未发生过故障事件。服务端应发送仅包含 DTCStoredDataRecordNumber（请求的记录编号的回声）的肯定响应。

DTCStoredDataRecord 信息应根据来自客户端的成功清除诊断信息请求进行清除。如果内存溢出（存储 DTC 和 DTCStoredDataRecord 数据的内存空间完全占用在服务端中），则车辆制造商有责任指定删除存储的 DTC 和 DTCStoredDataRecord 数据的规则。

5.7.3.1.7 为客户端定义的 DTC 掩码和客户端定义的 DTCExtendedData 记录号检索 DTCExtendedData 记录数据（子功能=0x06reportDTCExtDataRecordByDTCNumber）

客户端可以通过发送对该服务的请求，并将子功能设置为 reportDTCExtDataRecordByDTCNumber，来检索与 DTCExtendedData 记录号一起定义的 DTCMaskRecord 的客户端 DTCExtendedData。服务端应搜索其支持的 DTC，以便与客户端指定的 DTCMaskRecord（包含 DTC 编号（高、中、低字节））精确匹配。在这种情况下，客户机请求中提供的 DTCExtDataRecordNumber 参数应指定请求 DTCExtendedData 的指定 DTC 的特定 DTCExtendedData 记录。

与 DTC 号码和 statusOfDTC 一起，服务端应响应客户端的请求（DTCExtDataRecordNumber 不等于 0xFE 或 0xFF）返回单个预定义的 DTCExtendedData 记录。

车辆制造商应定义 DTCExtDataRecord 的格式和内容。DTCExtDataRecord 中报告的数据的结构由 DTCExtDataRecordNumber 以类似于记录 Data.er 中数据的定义的方式定义。响应中可以包括多个 DTCExtDataRecordNumbers 以及相关的 DTCExtDataRecords。使用一个或多个 DTCExtDataRecordNumbers 允许为单个 DTC 存储不同类型的 DTCExtDataRecords。

服务端应在单个响应消息中报告一个 DTCExtendedData 记录，除非客户机已经将 DTCExtDataRecordNumber 设置为 0xFE 或 0xFF。因为这将导致服务端在单个响应消息中使用为客户机定义的 DTCMaskRecord 存储的所有 DTCExtendedData 记录进行响应。

如果客户端指定的 DTCMaskRecord 或 DTCExtDataRecordNumber 参数无效或服务端不支持，则服务端将作出否定响应。这包括客户端发送 DTCExtDataRecordNumber of 0xFE，但服务端不支持 OBD 扩展数据记录（0x90 - 0xEF）的情况。这与客户机指定的 DTCMaskRecord 和/或 DTCExtDataRecordNumber 参数确实有效并由服务端支持，但是没有与其相关联的 DTC 扩展数据的情况不同（例如，因为扩展数据的内存溢出。在 reportDTCExtDataRecordByDTCNumber 的情况下，服务端应发送仅包含 DTCAndStatusRecord（请求的 DTC 编号（高、中、低字节）和 statusOfDTC 的回声）的肯定响应。

在接收清除诊断信息服务时清除 DTCExtendedData 信息是在 11.2.1 中指定的。车辆制造商有责任规定在存储器溢出（存储 DTC 的存储器空间和 DTC 扩展数据完全占用在服务端中）的情况下删除存储的 DTC 和 DTC 扩展数据的规则。

5.7.3.1.8 检索与客户端定义的严重性掩码记录匹配的 DTC 数量（子功能=0x07 reportNumberOfDTCBy.ityMaskRecord）

客户端可以通过发送具有设置为 reportNumberOfDTCBy.ityMaskRecord 的子功能的该服务请求，来检索与客户端定义的严重性状态掩码记录匹配的 DTC 数量的计数。服务端应扫描所有支持的 DTC，在客户端指定的掩码记录与每个存储的 DTC 的实际信息之间执行按位逻辑-and 操作。

```
((statusofdtc & dtcstatusmask) !!!!!!! = 0) && ((严重程度和 dtcseveritymask) !!!!!!! 0) == true)
```

对于产生真实结果的每个-and 操作，服务端应将计数器递增 1。如果客户端在掩码记录中指定了包含服务端不支持的位的状态掩码，则服务端应仅使用其支持的位来处理 DTC 信息。一旦所有支持的 DTC 都检查过一次，服务端将向客户端返回 DTCStatusAvailabilityMask 和产生的 2 字节计数。

注意：如果服务端内没有 DTC 符合客户机请求中指定的屏蔽标准，则服务端向客户机返回的计数应为 0。所报告的与 DTC 状态掩码匹配的 DTC 数量对于发出请求的时间点有效。所报告的 DTC 数量与通过子功能 reportDTCByStatusMask 读取的 DTC 的实际列表之间没有关系，因为读取 DTC 的请求是在不同的时间点完成的。

5.7.3.1.9 检索与客户端定义的严重性掩码记录匹配的严重性和功能单元信息（子功能=0x08 reportDTCBy.ityMaskRecord）

客户端可以通过发送具有设置为 reportDTCBy.ityMaskRecord 的子功能字节的请求来检索满足客户端定义的严重性掩码记录的 DTC 严重性和功能单元信息的列表。这个子功能允许客户端请求服务端报告具有一定严重性和状态的“testFailed”或“.ed”或“etc”的所有 DTC。评估应按以下方式进行：

服务端应在客户端请求中指定的 DTCSeverityMask 和 DTCStatusMask 与服务端支持的每个 DTC 相关联的实际 DTCSeverity 和 statusOfDTC 之间执行按位逻辑-and 操作。

除了 DTCStatusAvailabilityMask 之外，服务端还应返回 AND 操作的结果为真的所有 DTC，

```
((statusofdtc & dtcstatusmask) !!!!!!! = 0) && ((严重程度和 dtcseveritymask) !!!!!!! 0) == true)
```

如果客户端在掩码记录中指定了包含服务端不支持的位的状态掩码，则服务端应仅使用其支持的位来处理 DTC 信息。如果服务端内的 DTC 与客户端请求中指定的屏蔽标准不匹配，则在肯定响应消息中的 DTCStatusAvailabilityMask 字节之后将不提供 DTC 或状态信息。

5.7.3.1.10 检索客户端定义的 DTC 的严重性和功能单元信息（子功能=0x09 report.ityInformationOfDTC）

客户端可以通过发送具有 report.ityInformationOfDTC 的子功能集的该服务的请求来检索客户端定义的 DTCMaskRecord 的严重性和功能单元信息。服务端应搜索其支持的 DTC，以便与客户端指定的 DTCMaskRecord（包含 DTC 编号（高、中、低字节））精确匹配。

5.7.3.1.11 检索服务端支持的所有 DTC 的状态（子功能=0x0AreportSupportedDTC）

客户端可以通过发送带有 reportSupportedDTC 子功能集的服务请求来检索服务端支持的所有 DTC 的状态。对该请求的响应包含 DTCStatusAvailabilityMask，它提供服务端为屏蔽目的而支持的 DTC 状态位的指示。在 DTCStatusAvailabilityMask 之后，响应还包含 listOfDTCAndStatusRecord，它包含服务端支持的每个诊断故障代码的 DTC 号和相关状态。

5.7.3.1.12 检索第一个/最近失败的 DTC（子功能=0x0BreportFirstTestFailedDTC，子功能=0x0DreportMostRecentTestFailedDTC）

客户端可以通过发送一个请求，分别将子功能字节设置为“reportFirstTestFailedDTC”或“reportMostRecentTestFailedDTC”，从服务端检索第一个/最近失败的 DTC。与 DTCStatusAvailabilityMask 一起，服务端应将第一个或最近失败的 DTC 号码和相关状态返回给客户端。

如果自从上次客户端请求服务端清除诊断信息以来没有失败的 DTC 被记录，那么在肯定响应消息中的 DTCStatusAvailabilityMask 字节之后不应提供 DTC/状态信息。此外，如果自从上次客户机请求服务端清除诊断信息以来只有一个 DTC 出现故障，那么一个失败的 DTC 将返回给来自客户机的 reportFirstTestFailedDTC 和 reportMostRecentTestFailedDTC 请求。

第一次/最近一次失效的 DTC 的记录应独立于已确认的 DTC 的老化过程。

如上所述。第一/最近失败的 DTC 信息应根据来自客户端的成功清除诊断信息请求而被清除(参见 D.2 中的 DTC 状态位定义，以获得关于在将来接收清除诊断信息服务请求的情况下的 DTC 状态位处理的进一步描述)。Rver)。

5.7.3.1.13 检索第一个/最近检测到的确认的 DTC（子功能=0x0CreportFirstCon.edDTC，子功能=0x0EreportMostRecentCon.edDTC）

客户机可以通过发送请求从服务端检索第一个/最近确认的 DTC，该请求的子功能字节分别设置为“reportFirstCon.edDTC”或“reportMostRecentCon.edDTC”。与 DTCStatusAvailabilityMask 一起，服务端应将第一或最近确认的 DTC 号码和相关状态返回给客户端。

如果自从上次客户端请求服务端清除诊断信息以来没有确认的 DTC 被记录，那么在肯定响应消息中的 DTCStatusAvailabilityMask 字节之后不应提供 DTC/状态信息。此外，如果自从上次客户机请求服务端清除诊断信息以来只有 1 个 DTC 得到确认，那么一个确认的 DTC 将返回到客户机的 reportFirstCon.edDTC 和 reportMostRecentCon.edDTC 请求。

第一批经确认的直接转账簿，过去一度发生转账失败的，应当予以保存。但随后在来自客户端的请求时间之前满足老化标准(与上述 DTC 被确认之后被确认的任何其他 DTC 无关)。同样，最近确认的 DTC 的记录，如果 DTC 在过去某一时刻得到确认，则应予以保存。但是随后在来自客户端的请求时间之前满足老化标准(假设在上述 DTC 失败之后没有其他 DTC 得到确认)。

如上所述，第一/最近确认的 DTC 信息应根据来自客户端的成功清除诊断信息请求进行清除。

5.7.3.1.14 从服务端 DTC 镜像内存中检索与客户端定义的状态掩码匹配的 DTC 列表(子功能=0x0F reportMirrorMemoryDTCByStatusMask)

子功能 reportMirrorMemoryDTCByStatusMask 的处理与为 reportDTCByStatusMask 定义的处理相同，只是使用存储在服务端的 DTC 镜像内存中的 DTC 执行所有状态掩码检查。DTC 镜像内存是服务端中无法被清除诊断信息(0x14)服务擦除的附加可选错误内存。DTC 镜像存储器镜像普通 DTC 存储器，如果擦除普通错误存储器，则可以用作示例。

5.7.3.1.15 从 DTC 镜像存储器中检索镜像存储器 DTCExtendedData 记录数据，用于客户端定义的 DTC 掩码和客户端定义的 DTCExtendedData 记录编号（子功能=0x10 reportMirrorMemoryDTCExtDataRecordByDTCNumber）

子功能 reportMirrorMemoryDTCExtDataRecordByDTCNumber 的处理与为 reportCExtdataRecordByDTCNumber 定义的处理相同，只是数据是从 DTC 镜像内存中检索出来的。DTC 镜像内存是服务端中无法被清除诊断信息(0x14)服务擦除的附加可选错误内存。DTC 镜像存储器镜像普通 DTC 存储器，并且可以用于例如如果普通错误存储器被擦除。

5.7.3.1.16 检索与客户端定义的状态掩码匹配的镜像内存 DTC 的数量（子功能=0x11 reportNumberOfMirrorMemoryDTCByStatusMask）

客户端可以通过发送与客户端定义的状态掩码匹配的镜像内存 DTC 数量的计数，其中子功能设置为 reportNumberOfMirrorMemoryDTCByStatusMask。对该请求的响应包含 DTCStatusAvailabilityMask，它提供服务端为屏蔽目的而支持的 DTC 状态位的指示。在 DTCStatusAvailabilityMask 之后，响应包含 DTCTFormat.er，它报告关于 DTC 格式和编码的信息。DTCTFormat.er 后面跟着 DTCCCount 参数，它是一个 2 字节的无符号数字，包含基于客户端提供的状态掩码在服务端内存中可用的 DTC 数量。

5.7.3.1.17 检索与客户端定义的状态掩码匹配的“仅与排放相关的 OBD” DTC 的数量（子功能=0x12 reportOfEmissionsOBDDTCByStatusMask）

客户机可以通过发送与该服务的请求以及设置为 reportNumberOfEmissionsOBDDTCByStatusMask 的子功能来检索与客户机定义的状态掩码匹配的“仅与排放相关的 OBD” DTC 数量的计数。对该请求的响应包含 DTCStatusAvailabilityMask，它提供服务端为屏蔽目的而支持的 DTC 状态位的指示。在 DTCStatusAvailabilityMask 之后，响应包含 DTCTFormat.er，它报告关于 DTC 格式和编码的信息。DTCTFormat.er 后面跟着 DTCCCount 参数，它是一个 2 字节的无符号数字，包含基于客户端提供的状态掩码在服务端内存中可用的“仅与排放相关的 OBD” DTC 的数量。

5.7.3.1.18 检索与客户端定义的状态掩码匹配的“仅与排放相关的 OBD” DTC 的列表（子功能=0x13 reportEmissionsOBDDTCByStatusMask）

客户端可以检索“仅与排放相关的 OBD” DTC 的列表，该列表通过发送具有设置为 reportEmissionsOBDDTCByStatusMask 的子功能字节的请求来满足客户端定义的状态掩码。这个子功能允许客户端请求服务端报告所有“testFailed”或“.ed”或“etc”的“排放相关 OBD” DTC。评估应如下进行：服务端应在客户端请求中指定的掩码与服务端支持的每个“与排放有关的 OBD” DTC 相关的实际状态之间执行按位逻辑-and 操作。除了 DTCStatusAvailabilityMask 之外，服务端还应返回 AND 操作的结果非零的所有“排放相关 OBD” DTC (即，statusOfDTC & DTCStatusMask) != 0)。如果客户端指定包含服务端不支持的位的状态掩码，则服务端应只使用它所支持的位来处理 DTC 信息。如果服务端内的“排放相关 OBD” DTC 与客户机请求中指定的屏蔽标准不匹配，则在肯定响应消息中的 DTCStatusAvailabilityMask 字节之后将不提供 DTC 或状态信息。

. ”当来自客户端的清除诊断信息请求成功时，应清除与排放相关的 OBD “DTC 状态信息（参见 D. 2 中的 DTC 状态位定义，以获得在服务端）。

5. 7. 3. 1. 19 检索 “prefailed” DTC 状态列表（子功能=0x14 reportDTCFault.on-.）

客户端可以检索在客户端请求时已经检测到或尚未检测到为“挂起”或“确认”的所有当前“预置”DTC 的列表。DTCFault.on. 的用意是识别一个不断增长的或间歇性的问题的简单方法，它不能被特定 DTC 的 statusOfDTC 字节识别/读取。DTCFault.on. 的内部实现应是车辆制造商特定的。“预配”DTC 的用例是在制造厂测试 DTC 时加速故障检测，这些 DTC 需要制造测试所不能接受的成熟时间。维修或安装新部件后，服务有类似的使用情况。

5. 7. 3. 1. 20 检索具有“永久 DTC”状态的 DTC 列表（子功能=0x15reportDTCWithPermanent.）

客户机可以检索具有“永久 DTC”状态的 DTC 列表，如 3. 1 所述。

5. 7. 3. 1. 21 为客户端定义的 DTCExtendedData 记录编号检索 DTCExtendedData 记录数据（子功能=0x16 reportDTCExtDataRecordByRecordNumber）

客户端可以通过发送对该服务的请求，并将子功能设置为 reportDTCExtDataRecordByRecordNumber 来检索客户端定义的 DTCExtendedData 记录编号的 DTCExtendedData。服务端应在所有支持的 DTC 中搜索与客户端指定的 DTCExtDataRecordNumber 的精确匹配。在这种情况下，客户机请求中提供的 DTCExtDataRecordNumber 参数应当为请求 DTCExtendedData 的所有支持的 DTC 指定特定的 DTCExtendedData 记录。

服务端应返回 DTCExtendedData 记录以及每个支持的 DTC 的 DTC 号和状态 OfDTC，其中包含请求的 DTCExtDataRecordNumber 的数据。

车辆制造商应定义 DTCExtDataRecord 的格式和内容。DTCExtDataRecord 中报告的数据的结构由 DTCExtDataRecordNumber 以与记录 Data.er 中数据的定义类似的方式定义。

如果客户端指定的 DTCExtDataRecordNumber 参数无效或服务端不支持，则服务端将作出否定响应。

在接收清除诊断信息服务时清除 DTCExtendedData 信息是在 11. 2. 1 中指定的。车辆制造商有责任规定在存储器溢出（存储 DTC 的存储器空间和 DTC 扩展数据完全占用在服务端中）的情况下删除存储的 DTC 和 DTC 扩展数据的规则。

5. 7. 3. 1. 22 从与客户端定义的状态掩码匹配的功能组检索 WWH-OBDDTC 列表（子功能=0x42 reportWWHOBDDTCByMaskRecord）

DTCSeverityMask（具有严重性和类）的实现和使用在 ISO° 27145-3[17]中定义。

11.3.1.1 检索具有“永久 DTC”状态的 WWH-OBD DTC 列表（子功能=0x55reportWWHOBDDTCWithPermanent.）

客户端可以检索具有“永久 DTC”状态的 WWH-OBD DTC 的列表，如 3. 1 所述。

11.3.1.2 从服务端的用户定义的 DTC 内存中检索与客户端定义的 DTC 状态掩码匹配的 DTC 列表(子功能=0x17 reportUserDefMemoryDTCByStatusMask)

客户端可以从用户定义的内存中检索 DTC 列表，通过发送具有设置为 reportUserDefMemoryDTCByStatusMask 的子功能字节的请求，满足客户端定义的状态掩码。这个子功能允许客户端请求服务端从用户定义的内存中报告所有“testFailed”或“.ed”OR“etc”的 DTC。

评估应如下进行：服务端应在客户端请求中指定的掩码与服务端在用户定义的存储器中支持的每个 DTC 相关联的实际状态之间执行按位逻辑.-ing 操作。除了 DTCStatusAvailabilityMask 之外，服务端还应返回 AND 操作的结果非零的所有 DTC（即 (statusOfDTC 和 DTCStatusMask) != 0）在特定的记忆中。如果客户端指定了包含服务端不支持的位的状态掩码，则服务端应仅使用其支持的位来处理 DTC 信息。如果在服务端中没有 DTC 与该特定内存中客户机请求中指定的屏蔽标准匹配，则不应在肯定响应消息中的 DTCStatusAvailabilityMask 字节之后提供 DTC 或状态信息。

DTC 状态信息不应当在来自客户端的清除诊断信息请求成功时清除，而应由制造商特定的例程控制清除。

11.3.1.3 从 DTC 用户定义内存（子功能=0x18 reportUserDefMemoryDTCSnapshotRecordByDTCNumber）中检索用户定义的内存 DTCSnapshot 记录数据，用于客户端定义的 DTC 掩码和客户端定义的 DTCSnapshotNumber

. 客户机可以通过发送具有 reportUserDefMemoryDTCSnapshotRecordByDTCNumber 的子功能集的该服务的请求来检索结合 DTCSnapshot 记录号和用户定义的内存标识符的客户机定义的 DTCMaskRecord 的捕获 DTCSnapshot 记录数据。服务端应搜索其支持的 DTC，以便与客户端指定的 DTCMaskRecord（包含 DTC 编号（高、中、低字节））精确匹配。客户机请求中提供的 DTCSnapshotRecordNumber 参数应指定指定 DTC 的特定发生以及请求 DTCSnapshot 记录数据的已定义内存。

注意 1 DTCSnapshotRecordNumber 不与 DTCStoredDataRecordNumber 共享相同的地址空间。

系统供应商/车辆制造商有责任确定在这些服务端中捕获的 DTCSnapshot 记录是否存储与首次或最近发生的故障相关的数据。

与 DTC 号码和状态 OfDTC 一起，服务端应响应客户端的请求，从特定用户存储器返回单个预定义的 DTCSnapshotRecord。如果已经为客户端定义的 DTCMaskRecord 和 DTCSnapshotRecordNumber 参数（DTCSnapshotRecordNumber 不等于 0xFF）和该特定内存识别出故障。

注 2. 准确的失效准则应由系统供应商/车辆制造商确定。

DTCSnapshot 记录可以包含多个数据参数，这些数据参数可用于在故障发生时重构车辆状况（例如，B+、RPM、时间戳）。

车辆制造商应在用户定义的存储器中定义 DTCSnapshotRecord 的格式和内容（即，DTCSnapshotRecord 的内容可以在不同存储器之间不同）。DTCSnapshotRecord 中报告的数据首先包含一个 data.er，用于标识以下数据。这种数据标识符 / 数据组合可以在 DTCSnapshotRecord 中重复。在用户定义的内存中，DTCSnapshotRecord 中的一个或多个数据标识符的使用允许针对单个 DTC 存储不同类型的数据。DTCSnapshotRecord，用于不同的故障发生当然。每个 DTCSnapshotRecord 应提供指示每个 DTCSnapshotRecord 中包含的记录数据标识符数量的参数，以帮助数据检索。

服务端应在单个响应消息中报告一个 DTCSnapshot 记录，除非客户端已经将 DTCSnapshotRecordNumber 设置为 0xFF。因为这将使服务端在单个响应消息中用为客户机定义的 DTCMaskRecord 和用户定义的内存存储的所有 DTCSnapshot 记录进行响应。DTCANSTATUS 记录只包含在响应消息中的一次。

如果客户端指定的 DTCMaskRecord、DTCSnapshotRecordNumber、UserDefMemory 参数无效或服务端不支持，则服务端将作出否定响应。这与客户机指定的 DTCMaskRecord 和/或 DTCSnapshotRecordNumber 参数确实有效并由服务端支持用于该特定内存，但是没有与其相关联的 DTCSnapshot 数据的情况不同（例如：因为指定的 DTC 或记录号从未发生过故障事件。服务端应发送仅包含 DTCAndStatusRecord（请求的 DTC 编号（高、中、低字节）的回声）以及 statusOfDTC 的肯定响应。

DTCSnapshot 信息应根据制造商的特定条件（例如例行控制）请求从客户端清除。车辆制造商有责任指定在存储器溢出（用于存储的 DTC 的存储器空间和用于该特定存储器的 DTC 快照数据完全占据服务端中的 DTC 快照数据）的情况下删除存储的 DTC 和 DTCSnapshot 数据的规则。

11.3.1.4 为客户端定义的 DTC 掩码和客户端定义的 DTCExtendedData 记录号从 DTC 内存中检索用户定义的内存 DTCExtendedData 记录数据（子功能=0x19 reportUserDefMemoryDTCExtDataRecordByDTCNumber）

客户机可以通过发送具有 reportUserDefMemoryDTCExtDataRecordByDTCNumber 子功能的该服务请求，来检索结合 DTCExtendedData 记录号和用户 DefMemoryIdenit. 的客户机定义的 DTCMaskRecord 的 DTCExtendedData。服务端应通过其支持的 DTC 搜索与客户端指定的 DTCMaskRecord（包含 DTC 编号（高、中、低字节））和用户 DefMemory.er 的精确匹配。在这种情况下，客户机请求中提供的 DTCExtDataRecordNumber 参数应指定请求 DTCExtendedData 的指定 DTC 的特定 DTCExtendedData 记录。

与 DTC 号码和 statusOfDTC 一起，服务端应响应客户端的请求（DTCExtDataRecordNumber 不等于 0xFE 或 0xFF）返回单个预定义的 DTCExtendedData 记录。

车辆制造商应定义 UserDefDTCExtDataRecord 的格式和内容。DTCExtDataRecord 中报告的数据的结构由 DTCExtDataRecordNumber 为特定用户定义的内存定义，其方式与记录 Data.er 中数据的定义类似。响应中可以包括多个 DTCExtDataRecordNumbers 以及相关的 DTCExtDataRecords。使用一个或多个 DTCExtDataRecordNumbers 允许为单个 DTC 存储不同类型的 DTCExtDataRecords。

服务端应在单个响应消息中报告一个 DTCExtendedData 记录，除非客户机已经将 DTCExtDataRecordNumber 设置为 0xFE 或 0xFF。因为这将导致服务端响应为客户端定义的 DTCMaskRecord 存储的所有 DTCExtendedData 记录，这些记录位于单个响应消息中用户定义的内存之外。

如果客户端指定的 DTCMaskRecord 或 DTCExtDataRecordNumber 参数无效，或者服务端不支持，或者不在该特定内存中，则服务端将作出否定响应。这与客户机指定的 DTCMaskRecord 和/或 DTCExtDataRecordNumber 参数确实有效并由服务端支持，但是没有与其相关联的 DTC 扩展数据的情况不同（例如，因为扩展数据的内存溢出。在 reportDTCExtDataRecordByDTCNumber 的情况下，服务端应发送仅包含 DTCAndStatusRecord（请求的 DTC 编号（高、中、低字节）和 statusOfDTC 的回声）的肯定响应。

车辆制造商有责任指定在用户定义的存储器中删除存储的 DTC 和 DTC 扩展数据的规则。

11.3.2 请求消息

11.3.2.1 请求消息定义

表 256 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

e) 表 256-请求消息定义-子功能=reportNumberOfDTCByStatusMask、reportDTCByStatusMask、
reportMirrorMemoryDTCByStatusMask、reportNumberOfMirrorMemoryDTCByStatusMask、
reportNumberOfEmissionsOBDDTCByStatusMask、reportEmissionsOBDDTCByStatusMask

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能 reporttype = = [reportnumberofdtcbystatusmask reportdtcbystatusmask reportmirrormemorydtcbystatusmask reportnumberofmirrormemorydtcbystatusmask ask]	米	0x01 0x02 0x0F 0x11 0x12	Lev_rnodtcbsm rdtcbsm rmmdtcbsm rnommtdtcbsm RN00EBDTCBSM ROBDTCBSM
#3	DTCSTATUS 掩膜	米	0x00 - 0xff	DTCSM

表 257 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

f) 表 257- 请求 消 息 定 义 -sub
function=reportDTCSnapshot.cation, reportDTCSnapshotRecordByD
TCNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能 [reporttype = = 报告数	米	0x03 0x04	Lev_rdtcssi rdtcsshdtc
#3	DTCMASKILTECT[0]= #3 dtchighbyte #4 dtcmiddlebyt #5	C - C	0x00 - 0xff 0x00 - 0xff 0...00 0...ff	dtcmrec_dtchb_dtcmb dtclb
#6	dtcsnapshotrecordnumber	C	0x00 - 0xff	二次谐波合

C: 只有在子功能参数等于 reportDTCSnapshotRecordByDTCNumber 的情况下，DTCMaskRecord 记录和 DTCSnapshotRecordNumber 参数才存在。

表 258 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

g) 表 258-请求消息定义-sub function=reportDTCStoredDataByRecordNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能[reporttype == 报告数据库记录号	米	0x05	Lev _ rdtcsdbrn
#3	dtcstoreddatarecordnumber	米	0x00 - 0xff	二磷酸胞苷

表 259 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

h) 表 259- 请求 消 息 定 义 -sub

function=reportDTCExtDataRecordByDTCNumber, reportMirrorMemoryDTCExtDataRecordByDTCNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	sub-function = [reportType = reportDTCExtDataRecordByDTCNumber reportMirrorMemoryDTCExtDataRecordByDTCNumber]	米	0x06 0x10	Lev _ rdtcedrbdn rmdedrbdn
#3 #4 #5	DTCMASKILTECT[0]= dtchighbyte dtcmiddlebyt ^	米 米	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcmb _ dtchb dtcmb dtclb
#6	dtceextdatarecordnumber	米	0x00 - 0xff	德特雷恩

表 260 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

i) 表 260- 请求 消 息 定 义 -sub

function=reportNumber0fDTCBy. ityMaskRecord, reportDTCSeverityInformation

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能[reporttype == 报告数据记录]	米	0x07 0x08	Lev _ rnodtcbsmr rdtchsmr
#3 #4	dccStReMySaskField[]= DTCSeverityMas k	米	0x00 - 0xff 0x00 - 0xff	dtcsvmrec _ dtcsvm dtcsm

表 261 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

j) 表 261-请求消息定义-sub function=report. ityInformation0fDTC

A_Data 字节	参数名	Cvt	字节值	助记符
-----------	-----	-----	-----	-----

#1	读取请求信息	米	0x19	RDTCI
#2	子功能[reporttype == reportseverityinformationofdtc]	米	0x09	Lev _ rsiodtc
#3 #4 #5	DTCMASKILTECT[0]= dtchighbyte dtcmiddlebyt 。	米 米 米	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcnrec _ 二氯乙烷 二甲基氯化 DTCLB

表 262 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

k) 表 262-请求消息定义-子功能=reportSupportedDTC、 reportFirstTestFailedDTC、
 reportFirstCon. edDTC、 reportMostRecentTestFailedDTC、 reportMostRecentCon. edDTC、
 reportDTCFault. on. 、 reportDTCWithPermanent.

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能[reporttype == reportSupportedDTC reportFirstTestFailedDTC reportFirstCon. edDTC reportMostRecentTestFailedDTC reportMostRecentCon. edDTC reportDTCFault. on. 、 reportDTCWithPermanent.]	米	0x0A 0x0B 0x0C 0x0D 0x0e 0x14	狮子_ rsupdtc rftfdtc rfcdtc rmrtfdtc rmrcdtc rdtcfdc

表 263 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

1) 表 263-请求消息定义-sub function=reportDTCExtDataRecordByRecordNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能[reporttype == reportdtcextdatarecordbyrecordnumber]	米	0x16	Lev _ rdtcedrbr
#3	dtcextdatarecordnumber	米	0x00 - 0xef	德特雷恩

表 264 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

m) 表 264-请求消息定义-sub function=reportUserDefMemoryDTCByStatusMask

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能[reporttype == reportuserdefmemorydtcbystatusmask]	米	0x17	Lev _ rudmdtcbsm
#3	DTCSTATUS 掩膜	米	0x00 -	DTCSM
#4	记忆选择	米	0x00 - 0xff	梅米斯

表 265 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

n) 表 265- 请求消息定义 -sub

```
function=reportUserDefMemoryDTCSnapshotRecordByDTCNumber
```

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能= [RePoType=RePueServer Debug	米	0x18	Lev _ rudmdtcssbdtc
#3 #4 #5	DTCMASKILTECT[0]= dtchighbyte dtcmiddlebyt	米 米 米	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcmrec _ 二氯乙烷 二甲基氯化铵 DTCLB
#6	dtcsnapshotrecordnumber	米	0x00 -	二次谐波合成
#7	记忆选择	米	0x00 - 0xff	梅米斯

表 266 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

o) 表 266- 请求消息定义 -sub

```
function=reportUserDefMemoryDTCExtDataRecordByDTCNu  
mber
```

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能 reportuserdefmemorydtcextdatarecordbydtcnub	米	0x19	Lev _ rudmdtcedrbdn
#3 #4 #5	DTCMASKILTECT[0]= dtchighbyte dtcmiddlebyt	米 米 米	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcmrec _ 二氯乙烷 二甲基氯化铵 DTCLB
#6	dtcextdatarecordnumber	米	0x00 - 0xff	德特雷恩
#7	记忆选择	米	0x00 -	梅米斯

表 267 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

p) 表 267-请求消息定义-子功能=reportWWOBDDTCByMaskRecord

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI

#2	子功能[reporttype == 报告记录]	米	0x42	Lev _ robddtcbmr
#3	功能群标识符	米	0x00 - 0xff	FGID
#4 #5	dccStReMySaskField[]= DTCSstatusMask DTCSseverityMas	米	0x00 - 0xff 0x00 - 0xff	dtcsvmrec _ dtcsm dtcsvm

表 268 基于使用的子功能参数定义了 ReadDTCInformation 请求消息的结构。

q) 表 268-请求消息定义-sub function=reportWWOBDDTCWithPermanent.

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读取请求信息	米	0x19	RDTCI
#2	子功能[reporttype == 永久性状态	米	0x55	Lev _ rwwobddtcwps
#3	功能群标识符	米	0x00 - 0xff	FGID

11.3.2.2 请求消息子功能参数\$Lead (LeVig) 定义

该服务使用子功能参数来选择表 269 中指定的 DTC 报告类型之一。可能的级别的解释和使用在下面详细说明（未示出禁止 PosRspMsgIndicationBit（位 7）。

r) 表 269 请求消息子功能定义

6 - 0 位	描述	Cvt	助记符
0x00	等值线 此值由本文档保留以供将来定义。	米	等值线
0x01	reportnumberofdtcbystatusmask 此参数指定服务端应向客户端发送与客户端定义的状态掩码匹配的 DTC 等值线。	U	RNDOTCBSM
0x02	reportdtcbystatusmask 此参数指定服务端应向客户端发送与客户端定义的状态掩码匹配的 DTC 和相应状态的列表。	U	RDTCBSM
0x03	报告识别 此参数指定服务端应向客户端发送所有 DTCSnapshot 数据记录标识（DTC 编号和 DTCSnapshot 记录编号）。	U	RDTCSSI
0x04	reportdtcsnapshotrecordbydtcnumber 此参数指定服务端应向客户端发送与客户端定义的 DTC 号和 DTCSnapshot 记录号（所有记录的 0xFF）相关联的 DTCSnapshot 记录。	U	RDTCSSBDTC
0x05	reportdtcstoreddatabyrecordnumber 此参数指定服务端应向客户端发送与客户端定义的 DTCSStoredData 记录编号（所有记录的 0xFF）相关联的 DTCSStoredData 记录。	U	RDTCSDBRN
0x06	reportdtcextdatarecordbydtcnumber 此参数指定服务端应向客户端发送与客户端定义的 DTC 号和 DTCExtendedData 记录号相关联的 DTCExtendedData 记录（对于所有记录 0xFF，对于所有 OBD 记录 0xFE）。	U	RDTECEDRBDN

0x07	报告数字记录 此参数指定服务端应向客户端发送与客户端定义的严重性屏蔽记录匹配的 DTC 错误	U	RNDTCBSMR
------	--	---	-----------

表 269 – (续)

6 - 0 位	描述	Cvt	助记符
0x08	报告全文记录 此参数指定服务端应向客户端发送与客户端定义的严重性屏蔽记录匹配的 DTC 列表和相应状态。	U	RDTCBSMR
0x09	reportseverityinformationofdtc 此参数指定服务端应向客户端发送客户端请求消息中指定的特定 DTC 的严重性信息。	U	RSIODTC
0x0A	报告支持 此参数指定服务端应向客户端发送服务端内支持的所有 DTC 和相 应的列表。	U	RSUPDTC
0x0B	reportfirsttestfaileddtc 此参数指定服务端应向客户端发送自上次清除诊断信息以来由服 务端检测到的第一个故障 DTC。注意，通过该子功能参数报告的信 息应独立于 DTC 是否被确认或老化。	U	射频热释光晶 体管
0x0C	reportfirstconfirmeddtc 此参数指定服务端应向客户端发送自上次清除诊断信息以来由服 务端检测到的第一个确认的 DTC。 通过该子功能参数报告的信息应独立于经确认的 DTC 的老化过程 (例如，如果 DTC 老化，从而允许其状态被复位，则第一经确认的 DTC 记录应继续由服务端保存，而不管成为事后确认)。	U	射频 CDTC
0x0D	reportmostrecenttestfaileddtc 此参数指定服务端应向客户端发送自上次清除诊断信息以来由服 务端检测到的最近失败的 DTC。注意，通过该子功能参数报告的信 息应独立于 DTC 是否被确认或老化。	U	RMRTFDTC
0x0e	reportmostrecentconfirmeddtc 此参数指定，服务端应向客户端发送自上次清除诊断信息以来由服 务端检测到的最新确认的 DTC。 注意，通过该子功能参数报告的信息应独立于经确认的 DTC 的老化 过程(例如，如果 DTC 老化，从而允许其状态被复位，则假定没有 其他 DTC 变为 c，则服务端应继续保存第一经确认的 DTC 记录)。 后来痊愈了。	U	RMRCDC
0x0F	报告镜像存储器 此参数指定服务端应从 DTC 镜像存储器向客户端发送 DTC 列表以及 与客户端定义的状态掩码匹配的相应状态。	U	RMDT-TCBSM

表 269 – (续)

6 - 0 位	描述	Cvt	助记符
0x10	<p>报告</p> <p>此参数指定服务端应将 DTCExtendedData 记录(从 DTC 镜像存储器中)与客户机定义的 DTC 号和 DTCExtendedData 记录号(所有记录的 0xFF, 所有 OBD 记录的 0xFE) DTC 相关联的 DTCExtendedData</p>	U	RMDEDRBDN
0x11	<p>【中文解释】：报告</p> <p>此参数指定服务端应将镜像内存中与客户端定义的状态掩码匹配的 DTC 数量发送到客户端。</p>	U	RNMODTCTBSM
0x12	<p>reportnumberofemissionsobddtcbystatusmask</p> <p>此参数指定服务端应向客户端发送与客户端定义的状态掩码匹配的与排放相关的 OBDDTC 的数量。报告的 OBDDTC 的数量应仅限于要求符合排放相关法律要求的那些。</p>	U	ReNoeBoDDTCBSM
0x13	<p>reportemissionsobddtcbystatusmask</p> <p>此参数指定服务端应向客户端发送与排放相关的 OBDDTC 列表和与客户端定义的状态掩码匹配的相应状态。所报告的 OBDDTC 清单应仅是符合与排放有关的法律要求的那些。</p>	U	罗布 TCSBSM
0x14	<p>reportdtcfaultdetectioncounter</p> <p>此参数指定服务端应向客户端发送当前已检测到或尚未检测到“挂起”或“确认”的“预置”DTC 的列表。</p> <p>DTCFault.on. 的用意是识别一个不断增长的或间歇性的问题的简单方法, 它不能被特定 DTC 的 statusOfDTC 字节识别/读取。DTCFault.on. 的内部实现应是车辆制造商特定的(例如, 字节数, 签名与未签名)。但是, 所报告的值应是按比例缩放的 1 字节有符号值, 以便+127(0x7F)表示“失败”的测试结果, 并且任何其他非零正值表示“预置”的测试结果。但是, 具有+127 值的 DTCFault.on. 的 DTC 不应根据以下规定规则报告。每次测试逻辑运行时, DTCFault.on. 应增加车辆制造商的特定数量, 并指示该测试运行的失败。</p> <p>报告的 DTCFault.on. 值大于零且小于+127(即, 0x01–0x7E)表明 DTC 启用标准得到满足, 并且未完成的测试结果至少在一个条件或阈值中预先出现。</p> <p>仅报告具有非零正值小于+127(0x7F)的 DTC。</p> <p>每次测试逻辑运行时, DTCFault.on. 应按汽车制造商的具体数量递减, 并指示该测试运行的通过。如果 DTCFault.on. 降低到零或低于 DTC, 则不再在肯定响应消息中报告。DTCFault.on. 的值不应在操作周期之间保持。</p> <p>如果收到清除诊断信息服务请求, 则所有 DTC 的 DTCFault.on. 值应重置为零。附加复位条件应由车辆制造商定义。参考 D.5, 例如实现细节。</p>	U	RDTCFDC

表 269 – (续)

6 – 0 位	描述	Cvt	助记符
0x15	永久状态报告 此参数指定服务端应向客户端发送具有“永久 DTC”状态的 DTC 列表。如 2.1 所述。	U	RDTCWPS
0x16	reportdtcextdatarecordbyrecordnumber 此参数指定服务端应向客户端发送与客户端定义的 DTCExtendedData 记录编号小于 0xF0 的 DTCExtendedData 记录相关。	U	RDTCEBR
0x17	reportuserdefmemorydtcbystatusmask 此参数指定服务端应从用户定义的 DTC 存储器中向客户端发送 DTC 列表以及与客户端定义的状态掩码匹配的对应状态。	U	RADMTCCTCM
0x18	回波测距仪 此参数指定服务端应向客户端发送与客户端定义的 DTC 号码和 DTCSnapshot 记录号(所有记录的 0xFF)相关联的 DTCSnapshot 记录——在用户定义的 DTC 内存中。	U	RUDDMTCSSBDTC
0x19	reportuserdefmemorydtcextdatarecordbydtcnumber 此参数指定服务端应向客户端发送与客户端定义的 DTC 号和 DTCExtendedData 记录号(所有记录的 0xFF)相关联的 DTCExtendedData 记录(在用户定义的 DTC 内存中)。	U	鲁德迈克德尔德布恩
0x1a – 0x41	等值线 此值由本文档保留以供将来定义。	米	等值线
0x42	报告记录 此参数指定服务端应向客户端发送 WWH OBD DTC 的列表以及与客户端定义的状态掩码和严重性掩码记录匹配的相应状态和严重性信息。	U	RWWHBDDTCBMR
0x43 – 0x54	等值线 此值由本文档保留以供将来定义。	米	等值线
0x55	永久状态 此参数指定服务端应向客户端发送具有“永久 DTC”状态的 WWH OBD DTC 的列表。如 2.1 所述。	U	RWWHBDOTCP
0x56 – 0x7f	等值线 此值由本文档保留以供将来定义。	米	等值线

11.3.2.3 请求消息数据参数定义

表 270 指定请求消息的数据参数。

s) 表 270 请求数据参数定义

定义
DTCSTATUS 掩膜 DTCSTATUS 掩码包含八个（8）DTC 状态位。八个比特中的每一个的定义可以在 D.2 中找到。该字节用于请求消息中，以允许客户端请求状态与 DTCStatusMask 匹配的 DTC 的 DTC 信息。如果 DTC 实际状态位中的任何一个被设置为“1”，并且 DTCStatusMask 中的对应状态位也被设置为“1”（即，如果 DTCStatusMask 是按位逻辑的并且与 DTC 实际状态相关且结果非零），则 DTC 状态与 DTCStatusMask 匹配。然后发生了一场比赛。如果客户端指定了包含服务端不支持的位的状态掩码，则服务端应仅使用其支持的位来处理 DTC 信息。 dtcmaskrecord dtchighbyte dtcmiddlebyte, dtclowbyte [,] DTCMaskRecord 是包含 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的 3 字节值，它们一起表示服务端支持的特定诊断故障代码的唯一标识号。 3 字节 DTC 编号的定义允许几种编码 DTC 信息的方法。也可以这样做 ↓ 根据 ISO 15031-6[12] 规范，使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。这种格式由 DTCFormat.er=SAE_J2012-DA_DTCFormat_00 标识，或者 ↓ 通过根据 ISO 14229 的这个部分使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码，其中没有指定任何解码方法，因此允许车辆制造商定义的解码方法。此格式由 DTCFormat.er=ISO_14229-1_DTCFormat 标识，或者 ↓ 根据 SAE J1939-73[19] 规范使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。这种格式由 DTCFormat.er=SAE_J1939-73_DTCFormat 标识，或者 ↓ 根据 ISO 11992-4[5] 规范使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。这种格式由 DTCFormat.er=ISO_11992-4_DTCFormat 标识。 ↓ 根据 ISO 27145-2[16] 规范使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。该格式由 DTCFormat.er=SAE_J2012-DA_WWH-OBD_DTCFormat 标识。 dtcsnapshotrecordnumber DTCSnapshotRecordNumber 是一个 1 字节的值，指示通过 reportDTCSnapshotByDTCNumber 子功能为客户端定义的 DTCMaskRecord 请求的特定 DTCSnapshot 数据记录的数量。DTCSnapshot 数据记录号 0x00 应保留用于立法目的（例如，WWH-OBD）。在 0x01 到 0xFE 范围内的 DTCSnapshot 记录应可供车辆制造商特定使用。0xFF 的值请求服务端一次报告所有存储的 DTCSnapshot 数据记录。 dtcstoreddatarecordnumber DTCStoredDataRecordNumber 是一个 1 字节的值，指示通过 reportDTCStoredDataByRecordNumber 子功能请求的特定 DTCStoredDataRecord 的数量。DTCStReDeDATA 记录号 0x00 应保留用于立法目的。在 0x01 到 0xFE 的范围内的 DTCStored 数据记录应可用于车辆制造商的特定用途。0xFF 的值请求服务端一次报告所有存储的 DTCStoredData 数据记录。

表 270 – (续)

定义
dtcextdatarecordnumber
DTCExtDataRecordNumber 是一个 1 字节的值，指示通过 reportDTCExtDataRecordByDTCNumber 和 reportDTCExtDataRecordByRecordNumber 子功能为客户端定义的 DTCMaskRecord 请求的特定 DTCExtendedData 记录的数量。对于与排放相关的服务端（OBD 兼容的 ECU），DTCExtDataRecordNumber 0x00 应保留以备将来 OBD 使用。
保留以下 dtExtDATAcReordNordNeX 范围：
↓ 0x00 的值由 ISO/SAE 保留。
↓ 值 0x01 – 0x8F 请求服务端向车辆制造商报告特定存储的 DTCExtendedData 记录。
↓ 0x90 – 0xEF 的值请求服务端报告存储的 DTCExtendedData 记录的立法 OBD。
↓ ISO/SAE 保留 0xF0 – 0xFD 的值，用于将来在单个响应消息中报告组。
↓ 0xFE 的值请求服务端在单个响应消息中报告所有合法的 OBD 存储 DTCExtendedData 记录。
↓ 0xFF 的值请求服务端在单个响应消息中报告所有存储的 DTCExtendedData 记录。
DcStReStudiMaskList [DcStutyMyScript, DTCStatusMask]
DTCSeverityMaskRecord 是一个 2 字节的值，包含 DTCSeverityMask 和 DTCStatusMask(参见 D. 3 和 D. 2)。
德克萨斯要塞
DTCSENTITYMASK 包含三个 DTC 严重比特。三个比特中的每一个的定义可以在 D. 3 中找到。该字节用于请求消息中，以允许客户端请求严重性定义与 DTCSeverityMask 匹配的 DTC 的 DTC 信息。如果 DTC 实际严重性位中的任何一个被设置为“1”，并且 DTCSeverityMask 中的相应严重性位也被设置为“1”（即，1），则 DTC 严重性定义与 DTCSeverityMask 匹配。如果 DTCSeverityMask 在逻辑上与 DTC 的实际严重性成位 ANDed，并且结果非零，则发生匹配。
功能群标识符
FunctionalGroup. er 用于区分测试设备在由许多不同的 ECU 组成的电气架构中的不同功能系统组之间发送的命令。如果 ECU 已经实现了排放系统软件以及在 I/M 测试期间可以检查的其他系统，则仅报告所请求的功能系统组的 DTC 信息是重要的。I/M 测试不应失败，因为另一个功能系统组存储了 DTC 信息。
函数组标识符在 D. 5 中指定。
记忆选择
当检索 DTC 时，此参数应用于寻址各个用户定义的 DTC 存储器。

11.3.3 肯定响应信息

11.3.3.1 肯定响应消息定义

对服务 ReadDTCInformation 请求的肯定响应取决于服务请求中的子功能。

表 271 定义了子功能参数的肯定响应消息格式。

t) 表 271-响应消息定义-子功能=reportNumberOfDTCByStatusMask、
 reportNumberOfDTCBy.ityMaskRecord、reportNumberOfMirrorMemoryDTCByStatusMask、
 reportNumberOfEmissionsOBDDTCByStatusMask

A_Data 字	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCPRI
#2	reporttype = [reportnumberofdtcbystatusmask reportnumberofdtcbyservicemaskrecord reportnumberofmirrormemorydtcbystatusma	米	0x01 0x07 0x11	Lev_rnodtcbsm rnodtcbsmr rnommtdtcbsm ReNoeBoDDTCBSM
#3	DTCSTATUS 可用性要求	米	0x00 - 0xff	二氯化锡
#4	DTC 公式= 在这一点上，我要说的是，我的意思是，这是一个很好的选择。	米	0x00 0x01 0x02 0x03	dtdcid _ J2012-DADTCF00 14229—1DTCF J1939—73DTCF 11992~4DTCF
#5 #6	DTCCOUNT[0]= 一点	米	0x00 - 0xff 0x00 - 0xff	_DTCC dtcchb dtcclb

表 272 定义了子功能参数的肯定响应消息格式。

u) 表 272-响应消息定义-子功能=reportDTCByStatusMask、reportSupportedDTC、
 reportFirstTestFailedDTC、reportFirstCon.edDTC、reportMostRecentTestFailedDTC、
 reportMostRecentCon.edDTC、reportMirrorMemoryDTCByStatusMask、
 reportEmissionsOBDDTCByStatusMask。永久状态报告

A_Data 字	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCPRI
#2	reporttype = [reportDTCByStatusMaskreport SupportedDTCreportFirstTestFai ledDTCreportFirstCon.edDTCrepo rtMostRecentTestFailedDTCrepor tMostRecentCon.edDTCreportMost RecrrorMemoryDTCByStatusMaskrep ortEmissionsOBDDTCByStatusMask	米	0x02 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F	Lev_rdtcbsm rsupdtc rftfdtc rfcdtc rmrtfdtc rmrcdtc rmmtdtcbsm robddtcbsm
#3	DTCSTATUS 可用性要求	米	0x00 -	二氯化锡
#4	DTCANSTATUS 记录[]= 	C 1	0x00 -	dtdcasr _ 二氯乙烷

#5	dtchighbyte #	C 1	0x00 -	二甲基氯化铵
#6	dtcmiddlebyte	C 1	0x00 -	DTCLB
#7	# 1 1 1 1	C 1	0x00 -	SODTC
#8	dtclowbyte #	C 2	0x00 -	二氯乙烷
#9	statusofdtc #	C 2	0x00 -	二甲基氯化铵
#10	dtchighbyte #	C 2	0x00 -	DTCLB
#11	dtcmiddlebyte	C 2	0x00 -	SODTC
:		:	:	:
的 n - 3 #	# 2 2 2 2	C 2	0x00 -	二氯乙烷
# N 2	dtclowbyte #	C 2	0x00 -	二甲基氯化铵
的 N - 1 #		C 2	0x00 -	DTCLB
#n		C 2	0x00 -	SODTC

表 272 – (续)

A_Data 字节	参数名	Cvt	字节值	助记符
C 1:	这个参数只有在 DTC 信息可以被报告的情况下才存在。			
C 2:	此参数仅当 reportType=reportSupportedDTC、reportDTCByStatusMask、reportMirrorMemoryDTCByStatusMask、reportEmissionsOBDDTCByStatusMask、reportDTCWithPermanentStatusStak 和多于一个 DTC 信息可用于报告时才存在。			

表 273 定义了子功能参数的肯定响应消息格式。

v) 表 273–响应消息定义–sub function=reportSnapshot.cation

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCPRI
#2	reporttype = [报告识别	米	0x03	Lev _ rdtcssi
#3 #4 #5	(1) = (# dtcrecord dtchighbyte # 1 1 1 dtomidddlobyte)	C 1 C 1 C 1	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcasr _ 二氯乙烷 二甲基氯化 DTCLB
#6	dtcsnapshotrecordnumber # 1	C 1	0x00 - 0xff	二次谐波合
:	:	:	:	:
的 n - 3 # # N 2 的 N - 1 #	dtcrecord # m=[[] dtchighbyte # m m m dtomidddlobyte #]	C 2 C 2 C 2	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcasr _ 二氯乙烷 二甲基氯化 DTCLB
#n	dtcsnapshotrecordnumber # m	C 2	0x00 - 0xff	二次谐波合
C 1: DTCTRecord 和 DTCSnapshotRecordNumber 参数仅在至少一个 DTCSnapshot 记录可用时才存在。				
C 2: DTCTRecord 和 DTCSnapshotRecordNumber 参数仅当可以报告多个 DTCSnapshot 记录时才存在。				

表 274 定义了子功能参数的肯定响应消息格式。

w) 表 274–响应消息定义–sub function=reportDTCSnapshotRecordByDTCNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCPRI
#2	reporttype = [reportdtcsnapshotrecordbydtcnumber]	米	0x04	Lev _ rdtcssbdtc
#3	DTCANSTATUS 记录[] =	米	0x00 - 0xff	dtcasr _ 二氯乙烷

#4	dtchighbyte	米	0x00 - 0xff	二甲基氯化
#5	dtcmiddlebyt	米	0x00 - 0xff	DTCLB
#6	dtclowbyte	米	0x00 - 0xff	SODTC
#7	dtcsnapshotrecordnumber # 1	C 1	0x00 - 0xff	二次谐波合
#8	dtcsnapshotrecordnumberofidentifiers # 1	C 1	0x00 - 0xff	DTCSRNI

表 274 – (续)

A_Data 字节	参数名	Cvt	字节值	助记符
#9 #10 #11 : # 11 + (P 1) : # R (M - 1 - # R (M - 1) # R (M - 1) : 阿尔	DTCSNAPNOPTRTRECORE [...] dataidentifier # 字节 (MSB) # 1 1 1 2 # dataidentifier # 字节 1 字节 (LSB) snapshotdata # # 1 : snapshotdata # 1 字节 # P : <u>Dat aidfier detailfier</u>	C 1 C 1 C 1 C 1 C 1 C 1 C 2 C 2 C 2 C 2 C 2 C 2 C 2	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff : 0x00 - 0xff : 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	DTCSSRI didb11 didb12 SSD11 : SSD1P : didb21 didb22 SSD21 : SSD2M
:	:	:	:	:
γ T	dtcsnapshotrecordnumber # x	C 3	0x00 - 0xff	二次谐波合成
# T + 1	dtcsnapshotrecordnumberofidentifiers # x	C 3	0x00 - 0xff	DTCSRNI
# T + 2 # T + 3 # T 5 : # T + 5 + (P : # N- (U - 2 - # N- (U - 1) # N- (U - 1) : #n	我的意思是，你知道吗? dataidentifier # 字节 (MSB) # 1 1 1 2 # dataidentifier # 字节 1 字节 (LSB) snapshotdata # # 1 : snapshotdata # 1 字节 # P : <u>Dat aidfier detailfier</u>	C 3 C 3 C 3 C 3 C 3 C 3 C 4 C 4 C 4 C 4 C 4 C 4	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff : 0x00 - 0xff : 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	DTCSSRI didb11 didb12 SSD11 : SSD1P : didb21 didb22 SSD21 : SSD2U
C 1: DTCSnapshotRecordNumber 和 DTCSnapshotRecord 参数中的第一个 data.er/snapshotData 组合只有在至少一个 DTCSnapshot 记录可用时才存在。				
C 2/C 4 允许在单个 DTCSnapshotRecord 中存在多个数据标识符/快照数据组合。例如，对于单个数据标识符仅引用数据的完整部分的情况，情况就是这样。当 data.er 引用数据块时，可以使用单个 data.er/snapshotData 组合。				
. C 3: DTCSnapshotRecordNumber 和 DTCSnapshotRecord 参数中的第一个 data.er/snapshotData 组合仅在请求报告所有记录（在请求中将 DTCSnapshotRecordNumber 设置为 0xFF）并且可以报告多个记录时才存在。				

表 275 定义了子功能参数的肯定响应消息格式。

x) 表 275–响应消息定义–sub function=reportDTCStoredDataByRecordNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCIPR
#2	reporttype = [报告数据库记录号]	米	0x05	Lev _ rdtcsdbrn
#3	dtcstoreddatarecordnumber # 1	米	0x00 - 0xff	二磷酸胞苷
#4	DTCANSTATUS 记录[]* 1 =	C 1	0x00 - 0xff	dtcasr _ 二氯乙烷

#5	dtchighbyte	C 1	0x00 - 0xff	二甲基氯化
#6	dtcmiddlebyt	C 1	0x00 - 0xff	DTCLB
#7	dtclowbyte	C 1	0x00 - 0xff	SODTC

表 275 – (续)

A_Data 字	参数名	Cvt	字节值	助记符
#8	dtcstoredadatarecordnumberofidentifiers # 1	C 1	0x00 - 0xff	德克萨斯
#9 #10 #11 : # 11 +(P 1) : # R (M - 1) # R (M - 1) # R (M - 1) : 阿尔	dtcstoredadatarecord [1] = [# dataidentifier # #字节 (MSB) 1 1 1 2 # dataidentifier #字节 1 字节(LSB) dtcstoredata # # 1 : dtcstoredata # # P 1 字节 : data.er#w	C 1 C 1 C 1 C 1 C 1 C 2 C 2 C 2 C 2 C 2	0x00 - 0xff 0x00 - 0xff	dtcsdr _ didb11 didb12 dsd11 : dsd1p : didb21 didb22 dsd21 : dsd2m
:	:	:	:	:
γ T	dtcstoredadatarecordnumber # X	C 3	0x00 - 0xff	二磷酸胞苷
# T + 1 # T+2 # T+3 4 + T #	DTCANSTATUS 记录[]XX= dtchighbyte dtcmiddlebyt e dtclowbyte	C 3 C 3 C 3 C 3	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcasr _ 二氯乙烷 二甲基氯化 DTCLB SODTC
# T 5	dtcstoredadatarecordnumberofidentifiers # X	C 3	0x00 - 0xff	德克萨斯
# T 6 # T+7 8 + T # : # T + 8 + : # N- (U - 2) # N-(U - 1) # N- (U - : #n	dtcstoredadatarecord [] [# x = dataidentifier # #字节 (MSB) 1 1 1 2 # dataidentifier #字节 1 字节(LSB) dtcstoredata # # 1 : dtcstoredataa # # P 1 字节 : data.er#w	C 3 C 3 C 3 C 3 C 3 C 4 C 4 C 4 C 4	0x00 - 0xff 0x00 - 0xff	dtcsdr _ didb11 didb12 dsd11 : dsd1p : didb21 didb22 dsd21 : dsd2u
C 1: DTCStoredDataRecord 参数中的 DTCAndStatusRecord 和第一个 data.er/DTCStoredData 组合只有在至少一个 DTCStoredData 记录可用时才存在。				
C 2/C 4 允许在单个 DTCStoredDataRecord 中存在多个数据标识符/DTCStoredData 组合。例如，对于单个数据标识符仅引用数据的完整部分的情况，情况就是这样。当 data.er 引用数据块时，可以使用单个 data.er/DTCStoredData 组合。				
C 3: DTCStReDATA 记录号，DTCAndStatusRecord。并且 DTCStoredDataRecord 参数中的第一数据标识符 /DTCStoredData 组合仅当请求报告所有记录（请求中将 DTCStoredDataRecordNumber 设置为 0xFF）并且可以报告多个记录时才存在。				

表 276 定义了子功能参数的肯定响应消息格式。

y) 表 276- 响应消息定义 -sub function=reportDTCExtDataRecordByDTCNumber 和 reportMirrorMemoryDTCExtDataRecordByDTCNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCIPR
#2	reporttype = [reportDTCExtDataRecordByDTCNumber reportMirrorMemoryDTCExtDataRecordByDTCNumber]	米	0x06 0x10	Lev _ rdtcedrbd rmdedrbdn
#3 #4 #5 ... #7	DTCANSTATUS 记录[]= dtchighbyte dtcmiddlebyte e dtclowbyte dtcextdatarecordnumber # 1	米 米 米 C 1	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0xfd 0x00	HB DTCMB DTTCB SODTC 德特雷恩
#8 : # T + Q (D 1)	(1) = (# dtcextdatarecord extendeddata # # 1 字节 1 : extendeddata # # D 1 字节 1)	C C C	0x00 - 0xff : 0x00 - 0xff	dtcssr _ edd11 edd1p
:# γ T	dtcextdatarecordnumber # x	C 2	0x00 - 0xfd	德特雷恩
# T + 1 : # T + 1 (0)	dtcextdatarecord # X = [[] extendeddata # 字节 # x 1 : extendeddata # # 0 v 字节 1]	C 2 C	0x00 - 0xff : 0x00 - 0xff	dtcssr _ eddx1 EDDV0
C 1: DTCExtDataRecord 参数中的 DTCExtDataRecordNumber 和 extendedData 仅在至少一个 DTCExtDataRecord 可用时才存在。				
C 2: DTCExtDataRecord 参数中的 DTCExtDataRecordNumber 和 extendedData 仅在请求报告所有记录（请求中将 DTCExtDataRecordNumber 设置为 0xFE 或 0xFF）并且可以报告多个记录的情况下存在。				

表 277 定义了子功能参数的肯定响应消息格式。

z) 表 277- 响应消息定义 -sub
function=reportDTCExtDataRecordByDTCNumber, reportDTCInformation
ofDTC

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCIPR
#2	reporttype = [报告 DTC 的详细信息]	米	0x08 0x09	Lev _ rdtcbsmr rsiodtc
#3	DTCSTATUS 可用性要求	米	0x00 - 0xff	三氯化锡

表 277 – (续)

A_Data 字节	参数名	Cvt	字节值	助记符
#4 #5 #6 #7 #8 #9 ： # N 5 # n 4 的 n - 3 # # N 2 的 N - 1 # #n	dtand Dead StimeReal[][]= dtcseverity # dtcfunctionaluni t # 1 1 1 1 dtchighbyte # dtcmiddlebyte # dtclowbyte # statusofdtc # 1 1 ： dtcseverity # m m m dtcfunctionalunit	C 1 C 1 C 1 C 1 C 1 C 1 C 1 C 2 C 2 C 2 C 2 C 2 C 2 C 2 C 2	0x00 - 0xff 0x00 - 0xff	dtcasr 直接转矩 二氯四氟 二氯乙烷 二甲基氯 DTCLB SODTC ： 直接转矩 二氯四氟 二氯乙烷 二甲基氯 DTCLB SODTC
C 1: 在 reportDTCBy.ityMaskRecord 的情况下, 如果至少一个 DTC 与客户端定义的 DTC 严重性掩码匹配, 则必须存在该参数。在 report.ityInformationOfDTC 的情况下, 如果服务端支持请求消息中指定的 DTC, 则必须存在此参数。				
C 2: 此参数记录仅当 reportType=reportDTCBy.ityMaskRecord 时才存在。如果多个 DTC 与客户端定义的 DTC 严重性掩码匹配, 则必须存在它。				

表 278 定义了子功能参数的肯定响应消息格式。

aa) 表 278–响应消息定义–sub function=reportDTCFault.on.

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCPRI
#2	reporttype = [reportdtcfaultdetectioncounter]	米	0x14	Lev _ rdtcfdc
#3 #4 #5 #6 #7 #8 #9 #10 的 n - 3 # # N 2	dtcfaultdetectioncounterrecord [] = [dtchighbyte # dtcmiddlebyte # dtclowbyte # 1 1 1 2 1 dtcfaultdetectioncount er # dtchighbyte # dtcmiddlebyte # dtclowbyte # dtcfaultdetectioncount er # 2 2 2 ：	C 1 1 1 1 C C C C C	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x01 - 0xff 0x00 - 0xff C	DTCFDCRY-DT CHB DTTCB DTCLB DTCFDC DTTCB DTTCB DTCFT DTCFDC DTCHB DTCMB DTCLB 双馈直流传
C 1: 该参数仅当至少一个 DTC 具有正值小于 0x7F 的 DTCFault.on. 时才存在。				
C 2: 这个参数记录只在多于一个 DTC 的 DTCFault.on. 的正值小于 0x7F 的情况下出现。				

表 279 定义了子功能参数的肯定响应消息格式。

bb) 表 279- 响应消息定义 -sub

function=reportDTCExtDataRecordByRecordNumber

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCIPR
#2	reporttype = [reportdtcextdatarecordbyrecordnumber]	米	0x16	Lev _ rdtcedrbr
#3	dtcextdatarecordnumber	米	0x00 - 0xef	德特雷恩
#4	DTCANSTATUS 记录[]* 1= dtchighbyte	C	0x00 - 0xff	DTCASRY-DT CHB DTCMB
#5	dtcmiddlebyt	C	0x00 - 0xff	DTTCB SODTC
#6	e dtclowbyte	C	0x00 - 0xff	
#8	(1) = (# dtcextdatarecord extendeddata # # 1 字节 1 : # T + 1 # T + 2 # T + 3	C 1 C : # ... # D 1 为共 1 # ... # D 1 为共 1	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcedr _ edd11 edd1p
:	:	:	:	:
# T + 4 + (P 1)	DTCANSTATUS 记录[]XX= dtchighbyte dtcmiddlebyt e dtclowbyte statusofdtc	C 2 C C C	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	二氯乙酸
4 + T # : # T + 4 + (P 1)	dtcextdatarecord # X = [[] extendeddata # 字节 # x 1 : # ... # D V 字节 1	C 2 C : # ... # D V	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	dtcedr _ eddx1 :

C 1: DTCAndStatusRecord 和 DTCExtDataRecord 参数只有在至少一个 DTCExtDataRecord 可用时才存在。
C 2: DTCAndStatusRecord 和 DTCExtDataRecord 参数只在多于一个 DTCExtDataRecord 可供报告时才存在。
注意，实现者需要指定响应不会超过使用过的诊断通信所能达到的长度。

表 280 定义了子功能参数的肯定响应消息格式。

cc) 表 280-响应消息定义-sub function=reportUserDefMemoryDTCByStatusMask

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	59	RDTCIPR
#2	reporttype = [reportuserdefmemorydtcbystatusmask]	米	17	Lev _ rudmdtcbsm

#3	记忆选择	米	00 至 FF	梅米斯
#4	DTCSTATUS 可用性要求	米	00 至 FF	二氯化锡

表 280 – (续)

A_Data 字节	参数名	Cvt	字节值	助记符
#5 #6 #7 #8 #9 #10 #11 #12 ： 的 n - 3 # # N 2 的 N - 1 # #n	DTCANSTATUS 记录[]= dtchighbyte # dtcmiddlebyte # 1 1 1 1 dtclowbyte # statusofdtc # dtchighbyte # dtcmiddlebyte # 2 2 2 2 dtclowbyte # statusofdtc #	C 1 C 1 C 1 C 1 C 2 C 2	00 至 FF 00 至 FF	dtcasr 二氯乙烷 二甲基氯化铵 DTCLB SODTC 二氯乙烷 二甲基氯化铵 DTCLB SODTC ： 二氯乙烷 二甲基氯化铵 DTCLB SODTC
C 1:这个参数只有在 DTC 信息可以被报告的情况下才存在。				
C 2:这个参数只在多于一个 DTC 信息可以被报告的情况下存在。				

表 281 定义了子功能参数的肯定响应消息格式。

dd) 表 281– 响应消息定义 -sub

function=reportUserDefMemoryDTCSSnapshotRecordByDTCNu

mber

A_Data 字	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCPRI
#2	RePosiType = [RePurtServer Debug]	米	0x18	Lev _ rudmdtcssbdtc
#3	记忆选择	米	0x00 –	梅米斯
#4 #5 #6 #7	DTCANSTATUS 记录[]= dtchighbyte dtcmiddlebyt e dtclowbyte	米 米 米 米	0x00 – 0xff 0x00 – 0xff 0x00 – 0xff 0x00 – 0xff	dtcasr 二氯乙烷 二甲基氯化铵 DTCLB SODTC
#8	dtcsnapshotrecordnumber # 1	C 1	0x00 – 0xff	二次谐波合成孔
#9	dtcsnapshotrecordnumberofidentifiers # 1	C 1	0x00 – 0xff	DTCRNI
#10 #11 #12 ： # 12 +(P 1) ： # R (M – 1) # R (M – 1) # R (M – 1)	DTCSNAPNOPTRTRECORE [...] dataidentifier # 字节 (MSB) # 1 1 1 2 # dataidentifier #字节 1 字节 (LSB) snapshotdata # # 1 ： C 2 C 2 C 2	C 1 C 1 C 1 C 1 C 1 C 1 C 1 C 2 C 2 C 2	0x00 – 0xff 0x00 – 0xff	DTCSSRI didb11 didb12 SSD11 ： SSD1P ： didb21 didb22 SSD21

:		snapshotdata # 1 字节# P	C 2	:	SSD2M
:	:	.	C 2	0x00 - 0xff	:

表 281 - (续)

A_Data 字	参数名	Cvt	字节值	助记符
A_Data 字节	参数名	Cvt	字节值	助记符
γ T	dtcsnapshotrecordnumber # x	C 3	0x00 - 0xFF	二次谐波合
# T + 1	dtcsnapshotrecordnumberofidentifiers # x	C 3	0x00 - 0xFF	DTCSRNI
# T + 2 # T + 3 # T 5 : # T + 5 + (P 1) : # N- (U - 2 - 1)	我的意思是，你知道吗? dataidentifier # 字节 (MSB) # 1 1 1 2 # dataidentifier # 字节 1 字节 (LSB) snapshotdata # # 1 : snapshotdata # 1 字节 # P : Dataidfier detailfier	3 C C C C C C 3 3 3	0x00 - 0xFF 0x00 - 0xFF	dtcssr - didb11 didb12 ssd11 ssd1p : didb21 : didb22 ssd21
C 1: DTCSnapshotRecord 参数中的 DTCSnapshotRecordNumber 和第一个 data.er/snapshotData 组合仅当至少一个 DTCSnapshot 记录可被报告时才存在。				
C 2/C 4 允许在单个 DTCSnapshotRecord 中存在多个数据标识符/快照数据组合。例如，对于单个数据标识符仅引用数据的完整部分的情况，情况就是这样。当 data.er 引用数据块时，可以使用单个数据.er/snapshotData 组合。				
. C 3: DTCSnapshotRecord 参数中的 DTCSnapshotRecordNumber 和第一个 data.er/snapshotData 组合仅在请求报告所有记录（请求中将 DTCSnapshotRecordNumber 设置为 0xFF）并且可以报告多个记录时才存在。				

表 282 定义了子功能参数的肯定响应消息格式。

ee) 表 282- 响应消息定义 -sub

function=reportUserDefMemoryDTCExtDataRecordByDTCNum

ber

A_Data 字	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCIPR
#2	reportuserdefmemorydtceextdatarecordbydtcnrumber reporttype = []	米	0x19	Lev _ rudmdtcedrbdn
#3	记忆选择	米	0x00 - 0xFF	梅米斯
#4 #5 #6 #7	DTCANSTATUS 记录 []= dtchighbyte dtcmiddlebyte e dtclowbyte	米 米 米 米	0x00 - 0xFF 0x00 - 0xFF 0x00 - 0xFF 0x00 - 0xFF	dtcasr - 二氯乙烷 二甲基氯化铵 DTCLB SODTC
#8	dtceextdatarecordnumber # 1	C 1	0x00 - 0xFE	德特雷恩
#9	(1) = (#	C 1	0x00 - 0xFF	DTCSSRI edd11

# + 9 (P 1)	dtcextdatarecord	extendeddata # # 1 字节 1	C 1	0x00 - 0xff	: edd1p
:	:	:	:	:	:

表 282 – (续)

A_Data 字	参数名	Cvt	字节值	助记符
# T + 1 ⋮ # T + 1 +	dtcextdatarecord # X = [[]] extendeddata #字节# x 1 ⋮ [[]]	C 2 C 2 C 2	0x00 - 0xff ⋮ 0x00 - 0xff	DTCSSRI eddx1 ⋮ EDDXQ
C 1: DTCExtDataRecord 参数中的 DTCExtDataRecordNumber 和 extendedData 仅在至少一个 DTCExtDataRecord 可用时才存在。				
C 2: DTCExtDataRecord 参数中的 DTCExtDataRecordNumber 和 extendedData 仅在请求报告所有记录（请求中将 DTCExtDataRecordNumber 设置为 0xFE 或 0xFF）并且可以报告多个记录的情况下存在。				

表 283 定义了子功能参数的肯定响应消息格式。

ff) 表 283–响应消息定义–sub function=reportWWHOBDTCByMaskRecord

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCIPR
#2	reporttype = [报告记录]	米	0x42	Lev _ rwwhobddtcbsmr
#3	功能群标识符	米	0x00 - 0xff	FGID
#4	DTCSTATUS 可用性要求	米	0x00 - 0xff	二氯化锡
#5	dTabService 可用性要求	米	0x00 - 0xff	差热扫描隧道显微镜
#6	DTC 公式= Say To J20-da DTE DTCForce LE_04_Se_I1939-73_St .	米	0x04 0x02	dtdcfid _ j2012 - dadtdcf04_I1939
#7 #8 #9 #10 #11 ⋮ # n 4 的 n - 3 # # n 2	dtand Dead StimeReal[][]= dtcseverity # dtchighbyte # (MSB) 1 1 1 1 dtcmiddlebyte # dtclowbyte # statusofdtc # 1 ⋮ dtcseverity # dtchighbyte (MSB) # m 1 m m m dtcmiddlebyte # 1	1 1 C C C ⋮ C C 1	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff ⋮ 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff	DTCASRY-DT CS DTCHB DTTCB DTCLB SODTC dtcs ⋮ DTTCB DTCMB DTTCB
C 1: 这个参数只有在 DTC 信息可以被报告的情况下才存在。				
C 2: 这个参数只在多于一个 DTC 信息可以被报告的情况下存在。				

表 284 定义了子功能参数的肯定响应消息格式。

gg) 表 284-响应消息定义-sub function=reportWWHOBDDTCWithPermanent.

A_Data 字节	参数名	Cvt	字节值	助记符
#1	读写信息响应	米	0x59	RDTCIPR
#2	reporttype = [永久性状态]	米	0x55	Lev _ rwwhobddtcwps
#3	功能群标识符	米	0x00 - 0xff	FGID
#4	DTCSTATUS 可用性要求	米	0x00 - 0xff	二氯化锡
#5	DTC 公式= Say To J20-da DTE DTCForce LE 04 Se I1939-73 St .	米	0x04 0x02	dtdcfid _ j2012 _ dadtdcf04_I1939
#6 #7 #8 #9 : 的 n - 3 # # N 2	DTCANSTATUS 记录[]= dtchighbyte # (MSB) dtemiddlebyte # 1 1 1 1 dtclowbyte # statusofdtc # : dtchighbyte # (MSB) m m m dtemiddlebyte #	C C C C 1 1 1 1	0x00 - 0xff 0x00 - 0xff 0x00 - 0xff 0x00 - 0xff : 0x00 - 0xff 0x00 - 0xff	DTCASRY-DTC HB DTCMB DTTCB SODTC DTCHB DTCMB DTTCB SODTC
C 1:这个参数只有在 DTC 信息可以被报告的情况下才存在。				
C 2:这个参数只在多于一个 DTC 信息可以被报告的情况下存在。				

11.3.3.2 肯定响应消息数据参数定义

表 285 指定了肯定响应消息的数据参数。

hh) 表 285 响应数据参数定义

定义
报告类型
这个参数是客户机请求消息中提供的子功能参数的位 6-0 的回波。

DTC 记录

此参数记录包含一个或多个 DTCSeve.、DTCFunclonalUnit、DTCHighByte、DTCMiddleByte、DTCLowByte 和 statusOfDTC_if 分组，这些分组是 SAE_J2012-DA_DTCFormat_00、ISO_14229-1_DTCFormat、SAE_J1939-73_DTCFormat(参见下文以了解更多细节)。ISO1192-4DTC 格式或 SAEYJ2012-DAUDTCFATAT04。

DTCSeverity 识别故障对于车辆操作和/或系统功能的重要性，并允许向驾驶员显示推荐的动作。DTC-严重度的定义可以在 D.3 中找到。DTCFunctionalUnit 是一个 1 字节的值，用于标识报告 DTC 的相应的基本车辆/系统功能。DTCFunctionalUnit 的定义是特定于实现的，应在相应的实现标准中指定。

DTCHighByte、DTCMiddleByte 和 DTCLowByte 一起表示服务端支持的特定诊断故障代码的唯一标识号。DTCHighByte 和 DTCMiddleByte 表示正在被诊断的电路或系统。DTCLowByte 表示电路或系统中的故障类型（例如，传感器开路、传感器短路到地、基于算法的故障等）。该定义可以在 ISO 1531-6[12] 规范中找到。

此参数记录包含一个或多个 DTCSeve.、DTCFunclonalUnit、SPN(可疑参数号)、FMI(故障模式标识符) 和 OC(Occurrence.)，如果 SAE_J1939-73_DTCFormat.SPN、FMI 和 OC 在 SAE J1939 (18) 中定义。

表 285 – (续)

<h3>定义</h3>
<p>DTCANSTATUS 记录</p> <p>该参数记录包含一个或多个 DTCHighByte、DTCMiddleByte、DTCLowByte 和 statusOfDTC 分组，如果为 ISO_14229-1_DTCFormat 、 SAE_J2012-DA_DTCFormat_00 、 SAE_J1939-73_DTCFormat 、 SAE_J2012-DA_DTCFormat_04 或 ISO_11992-4_DTCFormat。SAE_J1939-73_DTCFormat 支持 SPN (可疑参数编号)、FMI (故障模式标识符) 和 OC (发生计数器) 参数。SPE、FMI 和 OC 在 SAE J1939 中定义。</p> <p>DTCHighByte、DTCMiddleByte 和 DTCLowByte 一起表示服务端支持的特定诊断故障代码的唯一标识号。3 字节 DTC 数的编码也可以完成：</p> <ul style="list-style-type: none"> ↓ 根据 ISO 15031-6[12] 规范，使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。这种格式由 DTCFormat.er=SAE_J2012-DA_DTCFormat_00 标识，或者 ↓ 通过根据 ISO 14229-1 规范使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码，该规范没有指定任何解码方法，因此允许车辆制造商定义的解码方法。此格式由 DTCFormat.er=ISO_14229-1_DTCFormat 标识，或者 ↓ 根据 SAE J1939-73[19] 规范使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。这种格式由 DTCFormat.er=SAE_J1939-73_DTCFormat 标识，或者 ↓ 根据 ISO 11992-4[5] 规范使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。这种格式由 DTCFormat.er=ISO_11992-4_DTCFormat 标识。 ↓ 根据 ISO 27145-2[16] 规范使用 DTCHighByte、DTCMiddleByte 和 DTCLowByte 的解码。这种格式由 DTCFormat.er=SAE_2012-DA_WWHOBD_DTCFormat 标识。
<p>DTC 记录</p> <p>此参数记录包含一个或多个 DTCHighByte、DTCMiddleByte 和 DTCLowByte 分组。DTCRecord 的解释取决于这个表中定义的 DTCFormat.er 参数中包含的值。</p>
<p>状态反馈</p> <p>特定 DTC 的状态（例如，测试此操作周期失败等）。statusOfDTC 字节中包含的比特的定义可以在本规范的 D.2 中找到。服务端不支持的位应被报告为“0”。</p>
<p>DTCSTATUS 可用性要求</p> <p>一个字节，其位定义为与 statusOfDTC 相同，并且表示服务端支持的状态位。服务端不支持的位应设置为“0”。对于服务端支持的每个 DTC，应实现每个支持的位（由值“1”表示）。</p>
<p>牙科医生</p> <p>这个 1 字节的参数值定义了服务端报告的 DTC 的格式。</p> <ul style="list-style-type: none"> ↓ SAE_J2012-DA_DTCFormat_00：该参数值标识服务端报告的 DTC 格式，如 ISO 15031-6[12] 规范中所定义。 ↓ ISO_14229-1_DTCFormat：该参数值标识服务端报告的 DTC 格式，如该表中由参数 DTCAndStatusRecord 定义的。 ↓ SAE_J1939-73_DTCFormat：该参数值标识由服务端报告的 DTC 格式，如 SAE J1939-73[19] 中所定义。 ↓ ISO_11992-4_DTCFormat：此参数值标识服务端报告的 DTC 格式，如 ISO 11992-4[5] 规范中所定义。 ↓ SAE_J2012-DA_DTCFormat_04：该参数值标识服务端报告的 DTC 格式，如 ISO 27145-2[16] 规范中所定义。 <p>DTCFormat.er 字节中包含的字节值的定义可以在本规范的 D.4 中找到。给定的服务端只支持一个 DTC 形式。</p>

表 285 – (续)

定义
DTCCOUNT 这个 2 字节参数共同引用响应 reportNumberOfDTCByStatusMask 或 reportNumberOfMirrorMemoryDTC 请求而发送的 DTCCCountHighByte 和 DTCCCountLowByte 参数。 DTCCOUNT 提供与客户机请求中定义的 DTCStatusMask 匹配的 DTC 数量的计数。
dtcsnapshotrecordnumber 或者是客户端在 reportDTCSnapshotRecordByDTCNumber 请求中指定的 DTCSnapshotRecordNumber 参数的回声，或者是存储的 DTCSnapshot 记录的实际 DTCSnapshotRecordNumber。
dtcsnapshotrecordnumberofidentifiers 这个单字节参数显示了紧跟在后面的 DTCSnapshotRecord 中的数据标识符的数量。0x00 的值应用于指示在对应的 DTCSnapshotRecord 中包括未定义数量的数据标识符（例如，主要用例是当 DTCSnapshotRecord 包含超过 255 个数据标识符时）。
DTCSNAP 记录 DTCSnapshotRecord 包含系统故障发生时的数据值的快照。
DTCSturdDATA 记录 DTCStoredDataRecord 包含系统故障发生时的数据值的冻结帧。
dtcstoredadatarecordnumber 或者是客户端在 reportDTCStoredDataByRecordNumber 请求中指定的 DTCStoredDataRecordNumber 参数的回声，或者是存储的 DTCStoredDataRecordRecord 的实际 DTCStoredDataRecordNumber。
dtcstoredadatarecordnumberofidentifiers 这个单字节参数显示了紧跟在后面的 DTCStoredDataRecord 中的数据标识符的数量。
dtcextdatarecordnumber 或者是客户端在 reportDTCExtDataRecordByDTCNumber 或 reportCExtDataRecordByRecordNumber 请求中指定的 DTCExtDataRecordNumber 参数的回声，或者是存储的 DTCExtendedData 记录的实际 DTCExtDataRecordNumber。
dtExtDATA 记录 DTCExtDataRecord 是服务端特定的信息块，可以包含与 DTC 相关联的扩展状态信息。 DTCExtendedData 包含 DTC 参数值，这些值在请求时已经标识。
dtcfaultdetectioncounterrecord DTCFault.on.Record 是一个记录，包括一个或多个 DTC 编号以及 DTC 特定的 DTCFault.on. 参数值。
dtcfaultdetectioncounter DTCFault.on. 报告 DTC 的故障检测计数的数目。
功能群标识符 包含 DTC(s) 的功能系统组的一个字节标识符，例如制动器、排放、乘员限制、轮胎充气、前向/外部照明等。这些值在 D.5 中定义。
dTabService 可用性要求 一个字节，其位定义与 DTCSeverity 相同，并且表示服务端支持的严重性位。服务端不支持的位应设置为

“0”。

记忆选择

此参数是客户机请求消息中提供的 MemorySelection 参数的回声。

11.3.4 支持的否定响应码 (NRCI)

该服务应实施下列否定响应代码。每个响应代码发生的环境记录在表 286 中。如果错误场景应用于服务端，则应使用列出的否定响应。

ii) 表 286 支持的否定响应码

NRC	描述	助记符
0x12	子功能不支持 如果不支持子功能参数，则应发送此 NRC。	SFNS
0x13	不正确消息格式无效格式 如果消息的长度错误，则应发送此 NRC。	英洛夫
0x31	请求范围 该 NRC 应发送如下： ↓ 客户端指定了服务端未识别的 DTCMASKEnter; ↓ 客户端指定了一个无效的 DTCSnapshotRecordNumber/DTCExtDataRecordNumber。注意，这与服务端支持 DTCSnapshotRecordNumber 和 DTCMaskRecord 组合或 DTCExtDataRecordNumber 和 DTCMaskRecord 组合但是当前没有数据与之关联的情况不同（即，无需数据的肯定响应）； ↓ 客户端指定服务端未识别的 FunctionalGroup. er;	罗尔

11.3.5 消息流示例-Read DTC 信息

11.3.5.1 一般假设

对于所有示例，客户端通过将 PosRspMsgIndicationBit (子功能参数的位 7) 设置为“FALSE” (0) 来请求具有响应消息。

11.3.5.2 示例#1-ReadDTCInformation, 子功能=reportNumberOfDTCByStatusMask

11.3.5.2.1 实例#1 概述

这个例子演示了 reportNumberOfDTCByStatusMask 子功能参数对于已确认的 DTC (DTC 状态掩码 0x08) 的使用以及各种掩码原理。这个服务端的 DTCSTATUS 可用性 = 0x2F。

11.3.5.2.2 Example of the Environment

服务端支持总共三个 DTCs (为了简单起见!)，它在客户端请求时具有以下状态：

以下假设适用于 DTC P0805-11 离合器位置传感器-接地短路 (0x080511)，状态 OfDTC 0x24 (0010 0100b)。

表 287 定义了 DTC P0805-11 的 StUSTOFDTC=0x24。

jj) 表 287——DTC P0805-11 的 StustoSFDT=0x24

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	在请求时 DTC 不再失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P0A9B-17 混合电池温度传感器-高于阈值(0xA9B17)的电路电压, 状态 OfDTC 为 0x26(0010 0110b)。

表 288 定义了 DTC P0A9B-17 的 StUSTOFDT=0x26。

kk) 表 288 -DTC P0A9B-17 的 STATISOFDT=0x26

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	在请求时 DTC 不再失败。
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P2522-1F A/C 请求 “B” -电路间歇(0x25221F), 状态 OfDTC 0x2F(0010 1111b)。

表 289 定义了 DTC P2522-1F 的 STATUSOFTC=0x2F。

11) 表 289——DTC P2522-1F 的 STATISOFDT=0x2F

statusofdtc 位字段:	比特	位状态	描述
------------------	----	-----	----

测试失败	0	1	DTC 在请求时失败
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

11.3.5.2.3 实例#1 消息流

在下面的示例中，由于只有 DTC P2522-1F A/C 请求“B”，所以向客户端返回一个计数
- 电路间歇(0x25221F)，0x2F(0010111b) 的状态 OfDTC 匹配客户端定义的 0x08(000000000b) 的状态掩码。

表 290 定义了 ReadDTCInformation, sub function=reportNumberOfDTCByStatusMask, 请求消息流示例#1。

mm) 表 290–ReadDTCInformation, sub function=reportNumberOfDTCByStatusMask, 请求消息
流示例#1

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 reportnumberofdtcbystatusmask suppressposrspmsgindicationbit	0x01	RNDOTCBSM
#3	DTCSTATUS 掩膜	0x08	DTCSM

表 291 定义了 ReadDTCInformation, sub function=reportNumberOfDTCByStatusMask, 肯定响应，示例#1。

nn) 表 291–ReadDTCInformation, sub function=reportNumberOfDTCByStatusMask, 肯定响应，示
例#1

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	reporttype = reportnumberofdtcbystatusmask	0x01	RNDOTCBSM
#3	DTCSTATUS 可用性要求	0x2f	二氯化锡
#4	DTCM=ISO14142~1Y-DTC 格式	0x01	14229 – 1dtcf
#5	DTCount [DTCount] DTCount	0x00	DTCCHB
#6	[dtcountwiByte]	0x01	DTCLB

11.3.5.3 示例#2–ReadDTCInformation, sub function=reportDTCByStatusMask, 匹配返回的 DTC

11.3.5.3.1 实例#2 概述

这个例子演示了 reportDTCByStatusMask 子功能参数的使用，以及结合不支持的屏蔽位的各种屏蔽原理。这个例子也适用于子功能参数 reportMirrorMemoryDTCByStatusMask 和子功能参数 reportUserDefMemoryDTCByStatusMask。除了使用存储在 DTC 镜像存储器或用户定义存储器中的 DTC 执行状态掩码检查之外。

11.3.5.3.2 Example of the Environment

除了第 7 位 “warningIndicatorRequested” 之外，服务端支持用于屏蔽目的的所有状态位。

服务端支持总共三个 DTCs（为了简单起见！），它在客户端请求时具有以下状态：

以下假设适用于 DTC P0A9B-17 混合电池温度传感器-高于阈值(0x0A9B17)的电路电压，状态 OfDTC 0x24(0010 0100b)。

表 292 定义了 DTC P0A9B-17 的 STATUSOFTC=0x24。

oo) 表 292 -DTC P0A9B-17 的 StutoSFDTC=0x24

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	在请求时 DTC 不再失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P2522-1F A/C 请求 “B” - 电路间歇 (0x25221F)，状态 OfDTC 0x00 (0000000b)。

表 293 定义了 DTC P2522-1F 的 StutoSFTC=0x00。

pp) 表 293——DTC P2522-1F 的 StutoSFDTC=0x00

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	0	DTC 在当前或以前的操作周期中没有失败。
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	0	DTC 测试自上次代码清除以来从未失败
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P0805-11 离合器位置传感器-接地短路(0x080511), 0x2F(0010 1111b)状态 OfDTC。

表 294 定义了 DTC P0805-11 的 StUSTOFCDTC=0x2F。

qq) 表 294——DTC P0805-11 的 StutoSFDTc=0x2F

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	1	DTC 在请求时失败
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

11.3.5.3.3 实例#2 消息流

在下面的示例中, DTC P0A9B-17(0xA9B17) 和 P0805-11(0x080511) 被返回到客户端的请求。不返回 DTC P2522-1F(0x25221F), 因为它的状态 0x00 与 0x84 的 DTCStatusMask 不匹配(如下面的示例中的客户端请求消息中指定的)。服务端应绕过对它不支持的状态位进行掩蔽。

表 295 定义了 ReadDTCInformation, sub function=reportDTCByStatusMask, 请求消息流示例#2。

rr) 表 295—ReadDTCInformation, sub function=reportDTCByStatusMask, 请求消息流示例#2

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能=RePtRtcByStasuScript, 删除	0x02	RDTCSM
#3	DTCSTATUS 掩膜	0x84	DTCSM

表 296 定义了 ReadDTCInformation, sub-function=reportDTCByStatusMask, 肯定响应, 示例#2。

ss) 表 296–ReadDTCInformation, sub function=reportDTCByStatusMask, 肯定响应, 示例#2

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCPRI
#2	RePosiType=RePrdTcByStasuScript	0x02	RDTCBM
#3	DTCSTATUS 可用性要求	0x7f	二氯化锡
#4	DTCANSTATUS 记录 1	0x0A	二氯乙烷
#5	DTCANSTATUS 记录 1	0x9b	二甲基氯化铵
#6	DTCANSTATUS 记录 1	0x17	DTCLB
#7	DTCANSTATUS 记录 1	0x24	SODTC
#8	DTCANSTATUS 记录 2	0x08	二氯乙烷
#9	DTCANSTATUS 记录 2	0x05	二甲基氯化铵
#10	DTCANSTATUS 记录 2	0x11	DTCLB
#11	DTCANSTATUS 记录 2	0x2f	SODTC

11.3.5.4 示例#3–ReadDTCInformation, 子功能=reportDTCByStatusMask, 没有返回匹配的 DTC

11.3.5.4.1 实例#3 概述

这个示例演示了在没有 DTC 与客户机定义的 DTCStatusMask 匹配的情况下, reportDTCByStatusMask 子功能参数的使用。

11.3.5.4.2 Example of the Environment

除了第 7 位 “warningIndicatorRequested” 之外, 服务端支持用于屏蔽目的的所有状态位。服务端支持总共两个 DTC (为了简单起见!), 它在客户端请求时具有以下状态:

以下假设适用于 DTC P2522-1F A/C 请求 “B” - 电路间歇(0x25221F), 状态 OfDTC 0x24(0010 0100b)。

表 297 定义了 DTC P2522-1F 的 STATUSOFTC=0x24。

tt) 表 297——DTC P2522-1F 的 StutoSFDTc=0x24

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	在请求时 DTC 不再失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败

彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P0A9B-17 混合电池温度传感器-高于阈值(0x0A9B17)的电路电压, 0x00(0000 0000b)的状态 OfDTC。

表 298 定义了 DTC P0A9B-17 的 StUSTOFDTC=0x00。

uu) 表 298 -DTC P0A9B-17 的 StUSTOFCDTC=0x00

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	0	DTC 在当前或以前的操作周期中没有失败。
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	0	DTC 测试自上次代码清除以来从未失败
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

客户端请求服务端将所有位 0 (TestFailed) 设置为逻辑“1”的 DTC 报告为 reportByStatusMask。

11.3.5.4.3 实例#3 消息流

在下面的示例中, 上述 DTC 中没有一个返回到客户端的请求, 因为在请求时, DTC 中没有一个测试失败。

表 299 定义了 ReadDTCInformation, sub function=reportDTCByStatusMask, 请求消息流示例#3。

vv) 表 299-ReadDTCInformation, sub function=reportDTCByStatusMask, 请求消息流示例#3

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能=RePtRtcByStasuScript, 删除	0x02	RDTCBSM
#3	DTCSTATUS 掩膜	0x01	DTCSM

表 300 定义了 ReadDTCInformation、sub function=reportDTCByStatusMask、肯定响应、示例#3。

ww) 表 300–ReadDTCInformation, sub function=reportDTCByStatusMask, 肯定响应, 示例#3

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	RePosiType=RePrdTcByStasuScript	0x02	RDTCSBM
#3	DTCSTATUS 可用性要求	0x7f	二氯化锡

11.3.5.5 示例#4–ReadDTCInformation, 子功能=reportDTCSnapshot.cation

11.3.5.5.1 实例#4 概述

这个示例演示了 reportDTCSnapshot.cation 子功能参数的使用。

11.3.5.5.2 Example of the Environment

下列假设适用：

- ↓ 服务端支持存储给定 DTC 的两个 DTCSnapshot 记录的能力。
- ↓ 服务端应指示当前为 DTC 编号 0x123456 存储了两个 DTCSnapshot 记录。为了这个示例的目的, 假设这个 DTC 已经发生过三次 (这样, 由于服务端内缺乏存储空间, 所以仅存储第一和最近的 DTCSnapshot 记录)。
- ↓ 服务端应指示当前为 DTC 编号 0x789ABC 存储了一个 DTCSnapshot 记录。
- ↓ 所有 DTCSNAPPEX 记录都按升序存储。

11.3.5.5.3 实例#4 消息流

在下面的示例中, 三个 DTCSnapshot 记录被返回到客户端的请求。

表 301 定义了 ReadDTCInformation、sub function=reportDTCSnapshot.cation、请求消息流示例#4。

xx) 表 301–ReadDTCInformation, sub function=reportDTCSnapshot.cation, 请求消息流示例#4

消息方向	客户机服务端
消息类型	请求

A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能=报告 删除	0x03	RDTCSI

表 302 定义了 ReadDTCInformation, sub function=reportDTCSnapshot.cation, 肯定响应, 示例#4。

yy) 表 302–ReadDTCInformation, sub function=reportDTCSnapshot.cation, 肯定响应, 示例#4

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	报告类型	0x03	RDTCSI
#3	DTCANSTATUS 记录 1	0x12	二氯乙烷
#4	DTCANSTATUS 记录 1	0x34	二甲基氯化铵
#5	DTCANSTATUS 记录 1	0x56	DTCLB
#6	dtcsnapshotrecordnumber # 1	0x01	二乙基氯化锡
#7	DTCANSTATUS 记录 2	0x12	二氯乙烷
#8	DTCANSTATUS 记录 2	0x34	二甲基氯化铵
#9	DTCANSTATUS 记录 2	0x56	DTCLB
#10	dtcsnapshotrecordnumber # 2	0x02	二乙基氯化锡
#11	DTCANSTATUS 记录 3	0x78	二氯乙烷
#12	DTCANSTATUS 记录 3	0x9a	二甲基氯化铵
#13	DTCANSTATUS 记录 3	0xBC	DTCLB
#14	dtcsnapshotrecordnumber # 3	0x01	二乙基氯化锡

11.3.5.6 示例#5–ReadDTCInformation, 子功能= reportDTCSnapshotRecord–ByDTCNumber

11.3.5.6.1 实例#5 概述

这个例子演示了 reportDTCSnapshotRecordByDTCNumber 子功能参数的使用。这个示例也适用于子功能参数 reportUserDefMemory–DTCSnapshotRecordByDTCNumber, 只是检查是用存储在用户定义内存中的 DTC 执行的。

11.3.5.6.2 Example of the Environment

下列假设适用：

↓ 服务端支持存储给定 DTC 的两个 DTCSnapshot 记录的能力。

↓ 此示例假定前面示例的延续。

↓ 假设服务端向服务端存储的两个 DTCSnapshot 记录中的第二个请求 DTC 编号 0x123456 (参见前面的示例, 其中向客户端返回 0x02 的 DTSnapshotRecordCount)。

- ↓ 假设 DTC 0x123456 的 statusOfDTC 为 0x24，并且每次发生 DTC 时捕获以下环境数据。
- ↓ DTCSNAPPAST 记录数据通过数据标识符 0x47 11 引用。

表 303 定义了 DTCSNAPPEX 记录内容。

zz) 表 303 -DTCSNAPPASS 记录内容

数据字节	DTCSNAPPASS 记录内容	字节值
#1	DTCSNAPHOSTERATION[数据 1] = ECT (发动机冷却剂温度)	0xA6
#2	DTCSNAPHOSTERATION[数据 2] = TP (节气门位置)	0x66
#3	DTCSNAPHOSTERATION[数据 3]=RPM (发动机速度)	0x07
#4	DTCSNAPHOSTERATION[数据 4]=RPM (发动机速度)	0x50
#5	DTCSNAPHOSTERATION[数据 5] = MAP (歧管绝对压力)	0x20

11.3.5.6.3 实例#5 消息流

在下面的示例中，根据客户端的 reportDTCSnapshotRecordByDTCNumber 请求返回一条 DTCSnapshot 记录。

表 304 定义了 ReadDTCInformation，子功能=reportDTCSnapshotRecordByDTCNumber，请求消息流示例#5。

aaa) 表 304-ReadDTCInformation，子功能=reportDTCSnapshotRecordByDTCNumber，请求消息

流示例#5

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 =reportDTCSnapshotRecordByDTCNumb	0x04	RDTCSRBDN
#3	(DTCHGBBER)	0x12	二氯乙烷
#4	DTCMaskField[DTCMIDLE 字节]	0x34	二甲基氯化铵
#5	dtclowbyte dtcmaskrecord []	0x56	DTCLB
#6	dtcsnapshotrecordnumber	0x02	二次谐波合成孔

表 305 定义了 ReadDTCInformation，sub function=reportDTCSnapshotRecordByDTCNumber，肯定响应，示例#5。

bbb) 表 305–ReadDTCInformation, sub function=reportDTCsnapshotRecordByDTCNumber, 肯定响应, 示例#5

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCPRI
#2	RePoType=RePrdTcSnApHoStReCordByDTC 编号	0x04	RDTCSRBDN
#3	DTCAndStatusRecord[DTCHighByte]	0x12	DTTCB
#4	DTCAndStatusRecord[DTCMiddleByte]	0x34	DTCMB
#5	e] DTCAndStatusRecord[DTCLowByte]	0x56	DTTCB 0x24
#6			
#7	dtcsnapshotrecordnumber	0x02	德特雷恩
#8	dtcsnapshotrecordnumberofidentifiers	0x01	DTCSRNI
#9	[1] # dataidentifier	0x47	didb1
#10	字节的字节 (MSB)	0x11	didb2
#11	dtcsnapshotrecord [日期] = 1 #	0xA6	EDY1
#12	等 dtcsnapshotrecord [日期] = 2	0x66	EDY2
#13	# dtcsnapshotrecord [日期] # TP	0x07	EDY3
#14	= 3 dtcsnapshotrecord # RPN RPN	0x50	EDY4

11.3.5.7 示例#6–ReadDTCInformation, 子功能= reportDTCStoredDataByRecordNumber

11.3.5.7.1 实例#6 概述

此示例演示 reportDTCStoredDataByRecordNumber 子功能参数的使用。

11.3.5.7.2 Example of the Environment

下列假设适用：

- ↓ 服务端支持为给定的 DTC 存储两个 DTCStoredDataRecords 的能力。
- ↓ 此示例假定前面示例的延续。
- ↓ 假设服务端请求服务端存储的两个 DTCStoredDataRecord 中的第二个 DTC 编号 0x123456(参见前面的示例, 其中两个 DTCStoredDataRecordCount 返回给客户端)。
- ↓ 假设 DTC 0x123456 的 statusOfDTC 为 0x24, 并且每次发生 DTC 时捕获以下环境数据。
- ↓ DTCStReDATA 数据记录数据通过数据标识符 0x47 11 引用。

表 306 定义了 DTCSnAPPASS 数据记录内容。

ccc) 表 306 -dtSturdDATA 记录内容

数据字节	DTCSnAPPASS 记录内容	字节值
#1	DTCSnTRADEDATATESTATION[数据 1] = ECT (发动机冷却剂温度)	0xA6
#2	DTCSnReDATA 记录[数据 2]=TP (节气门位置)	0x66
#3	DTCSnReDATA 记录[数据 3]=RPM (发动机速度)	0x07
#4	DTCSnReDATA 记录[数据 4]=RPM (发动机速度)	0x50
#5	DTCSnReDATA 记录[DATA 5]=MAP (歧管绝对压力)	0x20

11.3.5.7.3 实例#6 消息流

在下面的示例中，请求 DTCSnAPPASS 记录号 2，服务端返回 DTC 和 DTCSnAPPASS 记录内容。

表 307 定义了 ReadDTCInformation，sub function=reportDTCSnAPPASSByRecordNumber，请求消息流示例#6。

ddd) 表 307-ReadDTCInformation，sub function=reportDTCSnAPPASSByRecordNumber，请求消息流示例#6

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	sub function=reportDTCSnAPPASSByRec	0x05	RDTCSnDRBRN
#3	dtcsnappassrecordnumber	0x02	二磷酸胞苷脱氢

表 308 定义了 ReadDTCInformation，sub function=reportDTCSnAPPASSByRecordNumber，肯定响应，示例#6。

eee) 表 308-ReadDTCInformation，sub function=reportDTCSnAPPASSByRecordNumber，肯定响应，示例#6

消息方向	客户端服务端		
消息类型	响应		

A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读写信息响应	0x59	RDT CIPR
#2	RePosiType=RePrdTcStordDebug 记录编号	0x05	RDT CS DRBRN
#3	dtcstoreddatarecordnumber	0x02	二磷酸胞苷脱氢
#4	DTCANSTATUS 记录	0x12	二氯乙烷
#5	DTCANSTATUS 记录[DTCMIDLE 字节]	0x34	二甲基氯化铵
#6	dtclowbyte dtcandstatusrecord []	0x56	DTCLB
#7	DTCANSTATUS 记录[STATISOFDTC]	0x24	SODTC

表 308 – (续)

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#8	dtcstoredatarecordnumberofidentifiers	0x01	德克萨斯
#9	(名) 大同 (字节 1 (MSB))	0x47	DI dB1
#10	具有追溯性 [字典 2] (LSB)	0x11	DI dB2
#11	数据# dtcstoredatarecord [1] = 等	0xA6	EDY1
#12	数据# dtcstoredatarecord [2] = AP	0x66	EDY2
#13	数据# dtcstoredatarecord [3] = 转速	0x07	EDY3
#14	数据# dtcstoredatarecord [4] = 转速	0x50	EDY4
#15	数据# dtcstoredatarecord [5] = map	0x20	EDY5

11.3.5.8 示例#7–ReadDTCInformation, 子功能=reportDTCExtDataRecordByDTCNumber

11.3.5.8.1 实例#7 概述

这个例子演示了 reportDTCExtDataRecordByDTCNumber 子功能参数的使用。这个示例也适用于子功能参数 reportUserDefMemory–DTCExtDataRecordByDTCNumber，只是检查是用存储在用户定义内存中的 DTC 执行的。

11.3.5.8.2 Example of the Environment

下列假设适用：

- ↓ 服务端支持为给定的 DTC 存储两个 DTCExtendedData 记录的能力。
- ↓ 假设服务端请求服务端存储的所有可用 DTCExtendedData 记录以获得 DTC 编号 0x123456。
- ↓ 假设 DTC 0x123456 的 statusOfDTC 为 0x24，并且以下扩展数据可用于 DTC。
- ↓ DTCExtendedData 是通过 DTCExtDataRecordNumbers 0x05 和 0x10 引用的

fff) 表的内容 dtcextdatarecordnumber 0x05 309

数据字节	dtcextdatarecord contents for dtcextdatarecordnumber 0x05	字节值
#1	预热循环计数器——自 DTC 命令 MIL 关闭以来的预热循环次数	0x17

表的内容 dtcextdatarecordnumber 0x10 310

数据字节	dtcextdatarecord contents for dtcextdatarecordnumber 0x10	字节值
------	---	-----

#1	DTC 故障检测计数器-每次 DTC 测试检测到故障时递增，每次测试报告无故障时递减。	0x79
----	---	------

11.3.5.8.3 实例#7 消息流

在下面的示例中，客户端请求一个 DTCMaskRecord，该 DTCMaskRecord 包括 DTC 编号和值为 0xFF 的 DTCExtDataRecordNumber（报告所有 DTCExtDataRecords）。服务端返回两个 DTCExtDataRecords，这两个 DTCExtDataRecords 已经记录了客户机提交的 DTC 编号。

表 311 定义了 ReadDTCInformation，子功能=reportDTCExtDataRecordByDTCNumber，请求消息流示例#7。

ggg) 表 311–ReadDTCInformation, sub function=reportDTCExtDataRecordByDTCNumber, 请求消息流示例#7

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 =reportDTCExtDataRecordByDTCNumbe	0x06	RDTECEDRBDN
#3	(DTCHGBBER)	0x12	二氯乙烷
#4	DTCMaskField[DTCMIDLE 字节]	0x34	二甲基氯化铵
#5	dtclowbyte dtcmaskrecord []	0x56	DTCLB
#6	dtcextdatarecordnumber	0xff	德特雷恩

表 312 定义了 ReadDTCInformation，sub function=reportDTCExtDataRecordByDTCNumber，肯定响应，示例#7。

hhh) 表 312–ReadDTCInformation, sub function=reportDTCExtDataRecordByDTCNumber, 肯定响应，示例#7

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	reporttype = reportdtcextdatarecordbydtcnumber	0x06	RDTECEDRBDN
#3	DTCANSTATUS 记录	0x12	二氯乙烷
#4	DTCANSTATUS 记录[DTCMIDLE 字节]	0x34	二甲基氯化铵
#5	dtclowbyte dtcandstatusrecord []	0x56	DTCLB
#6	DTCANSTATUS 记录[STATISOFDTC]	0x24	SODTC
#7	dtcextdatarecordnumber	0x05	德特雷恩
#8	1 # dtcextdatarecord [字节]	0x17	EDY1

#9	dtcextdatarecordnumber	0x10	德特雷恩
#10	1 # dtcextdatarecord [字节]	0x79	EDY1

11.3.5.9 示例#8–ReadDTCInformation, sub function=reportNumberOfDTC–By. ityMaskRecord

11.3.5.9.1 实例#8 概述

此示例演示 reportNumberOfDTCBy. ityMaskRecord 子功能参数的使用。

11.3.5.9.2 例子

服务端总共支持三个 DTC，它们在客户端请求时具有以下状态：

以下假设适用于 DTC P0A9B-17 混合电池温度传感器–高于阈值的电路电压 (0x0A9B17)、状态 0fDTC 0x24(0010 0100b)、DTCF 功能单元=0x10、DTCSSeverity=0x20:

注意，只有严重字节的位 7 到 5 是有效的。

表 313 定义了 DTC P0A9B-17 的 STATUSOFTC=0x24。

iii) 表 313 –DTC P0A9B-17 的 STATUSOFTC=0x24

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	在请求时 DTC 不再失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P2522-1F A/C 请求 “B” –电路间歇 (0x25221F)、0x00(0000 0000 二进制) 状态 0fDTC、DTCFunctionalUnit=0x10、DTCSSeverity=0x20:

注意，只有严重字节的位 7 到 5 是有效的。

表 314 定义了 DTC P2522-1F 的 STATUSOFTC=0x00。

jjj) 表 314——DTC P2522-1F 的 STATUSOFTC=0x00

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败

彭定克	2	0	DTC 在当前或以前的操作周期中没有失败。
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	0	DTC 测试自上次代码清除以来从未失败
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P0805-11 离合器位置传感器-接地短路(0x080511)、0x2F(0010 1111b)状态 0fDTC、DTCFunctionalUnit= 0x10、DTCSeverity= 0x40:

注意，只有严重字节的位 7 到 5 是有效的。

表 315 定义了 DTC P0805-11 的 StUSTOFCDTC=0x2F。

kkk) 表 315——DTC P0805-11 的 StutoSFDTc=0x2F

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	1	DTC 在请求时失败
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

为了屏蔽目的，服务端支持 testFailed 和. edDTC 状态位。

11.3.5.9.3 实例#8 消息流

在下面的示例中，由于 DTC P0805-11 (0x080511) 与客户端定义的 0xC001 (DTCSeverityMask=110xxxxxb=0xC0, DTCSStatusMask=0000 0001b) 的严重性掩码记录匹配，所以向客户端返回一个计数。

表 316 定义了 ReadDTCInformation, sub function=reportNumberOfDTCBy. ityMaskRecord, 请求消息流示例#8。

111) 表 316—ReadDTCInformation, sub function=reportNumberOfDTCBy. ityMaskRecord, 请求消息流示例#8

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	sub function=reportNumberOfDTCBy. ityMa	0x07	RNDTCBSMR
#3	DTCSSeverityMaskRecord (德克萨斯)	0xC0	离散余弦变换支
#4	DTCSSeverityMaskRecord (DTCSTATUS)	0x01	DTCSM

表 317 定义了 ReadDTCInformation、sub function=reportNumberOfDTCBy. ityMaskRecord、肯定响应、肯定响应，示例#8。

mmm) 表 317–ReadDTCInformation, sub function=reportNumberOfDTCBy.ityMaskRecord, 肯定响应, 示例#8

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCPYR
#2	报告类型	0x07	RNDTCBSMR
#3	DTCSTATUS 可用性要求	0x09	二氯化锡
#4	DTCM=ISO14142^1Y-DTC 格式	0x01	14229 – 1dtcf
#5	达特县	0x00	二氯化锡
#6	达特县	0x01	DTCLB

11.3.5.10 示例#9–ReadDTCInformation, sub function= reportDTCBy.ityMaskRecord

11.3.5.10.1 示例#9 概述

此示例演示 reportDTCBy.ityMaskRecord 子功能参数的使用。

11.3.5.10.2 例子

11.3.5.9.2 中定义的假设和本子句中定义的假设都适用。

在下面的示例中, DTC P0805-11 (0x080511) 匹配客户端定义的 0xC001 (DTCSeverityMask=0xC0=110x XXXX b, DTCStatusMask=0x01, 0000 0001b) 的严重性掩码记录, 并向客户端报告。DTC P0805-11 (0x080511) 的严重性为 0x40 (010xxxxx b)。除了第 7 位 “warningIndicatorRequested” 之外, 服务端支持用于屏蔽目的的所有状态位。

仅注意严重性掩码字节的位 7 至 5 是有效的。

11.3.5.10.3 实例#9 消息流

在下面的示例中, 一个 DTCSeverityRecord 返回到客户端的请求。

表 318 定义了 ReadDTCInformation, sub function=reportDTCBy.ityMaskRecord, 请求消息流示例#9。

nnn) 表 318–ReadDTCInformation, sub function=reportDTCBy.ityMaskRecord, 请求消息流示例#9

消息方向	客户机服务端
消息类型	请求

A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能= 删除	0x08	RDTCBSMR
#3	DTCSeverityMaskRecord (德克萨斯)	0xC0	离散余弦变换支
#4	DTCSeverityMaskRecord (DTCSTATUS)	0x01	DTCSM

ooo) 表 319–ReadDTCInformation, sub function=reportDTCBy. ityMaskRecord, 肯定响应,

示例#9

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTcipR
#2	报告类型	0x08	RDTcbsmr
#3	DTCSTATUS 可用性要求	0x7f	二氯化锡
#4	dtCuffyTrimeA* 1	0x40	直接转矩控制系
#5	dtCuffyTrimeAs** 1 [DTC-功能单元]	0x10	二氯四氟化钾
#6	dtcseverityrecord # dtchighbyte [1]	0x08	二氯乙烷
#7	dtCuffyTrimeAs** 1 [dTCMIDLE 字节]	0x05	二甲基氯化铵
#8	dtCuffyTrimeAs** 1 [dTCLoByLy]	0x11	DTCLB
#9	dtCuffyTrimeSea* 1	0x2f	SODTC

11.3.5.11 示例#10–ReadDTCInformation, sub function=report. ityInformationOfDTC

11.3.5.11.1 示例#10 概述

此示例演示 report. ityInformationOfDTC 子功能参数的使用。

11.3.5.11.2 Example Proceedings

在 11.3.5.1.2 中定义的假设适用。

11.3.5.11.3 实例#10 消息流

在下面的示例中, 与客户端定义的 DTC 掩码记录匹配的 DTC P0805–11 (0x080511) 被报告给客户端。

ooo) 表 320–ReadDTCInformation, sub function=report. ityInformationOfDTC, 请

求消息流示例#10

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTci
#2	子功能 reportseverityinformationofdtc =, 删除	0x09	RSIODTC
#3	(DTCHBBER)	0x08	二氯乙烷

#4	DTCMaskField[DTCMIDLE 字节]	0x05	二甲基氯化铵
#5	dtclowbyte dtcmaskrecord []	0x11	DTCLB

表 321–ReadDTCInformation, sub function=report.ityInformationOfDTC, 肯定响应, 示例#10

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCPRI
#2	报告类型	0x09	RSIODTC
#3	DTCSTATUS 可用性要求	0x7f	二氯化锡
#4	dcService 记录[dcTrace]	0x40	直接转矩控
#5	DTC 记录[DTC 功能单元]	0x10	二氯四氟化
#6	dTraceRealEdvest[dtCHigEythy]	0x08	二氯乙烷
#7	dcService 记录[dTCMIDLE 字节]	0x05	二甲基氯化
#8	dtcseverityrecord dtclowbyte []	0x11	DTCLB
#9	dTraceRead 记录[STATISOFDTC]	0x2f	SODTC

11.3.5.12 示例#11–ReadDTCInformation–sub function=reportSupportedDTC

11.3.5.12.1 示例#11 概述

此示例演示 reportSupportedDTC 子功能参数的使用。

11.3.5.12.2 Example Proceedings

在 113.5.1.2 中定义的假设适用。此外，适用下列假设：

- ↓ 服务端支持总共三个 DTCs (为了简单起见!)，它在客户端请求时具有以下状态。
- ↓ 下面的两个假设适用于直接转矩控制 0x123456, statusofdtc 0x24 (0100 0010 (b))

qqq) 表 322 – StutoSoFTC=0x24

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	0	DTC 从未被证实
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	1	DTC 自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC 0x234505，状态 OfDTC 为 0x00 (0000000b)

rrr) 表 323 – StutoSoFTC=0x00

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	0	DTC 在当前或以前的操作周期中没有失败。
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	0	DTC 测试自上次代码清除以来从未失败
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC 0xABCD01, 状态 OfDTC 0x2F(0010 1111b)

sss) 表 324 –STATUSOFDTC=0x2F

statusofdtc 位字段:	比特 γ	位状态	描述
测试失败	0	1	DTC 在请求时失败
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

11.3.5.12.3 实例#11 消息流

在下面的示例中, 上述三个 DTC 都被返回到客户端的请求, 因为所有 DTC 都受到支持。

ttt) 表 325–ReadDTCInformation, sub function=reportSupportedDTC,
请求消息流示例#11

消息方向	客户机服务端
消息类型	请求

A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能=ReaveServices DDTCS, 删除	0x0A	RSUPDTC

表 326–ReadDTCInformation, sub function=readSupportedDTC, 肯定响应, 示例#11

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	RePixType = Read PosithDDTCs	0x0A	RSUPDTC
#3	DTCSTATUS 可用性要求	0x7f	二氯化锡
#4	DTCANSTATUS 记录 1	0x12	二氯乙烷
#5	DTCANSTATUS 记录 1	0x34	二甲基氯化铵
#6	DTCANSTATUS 记录 1	0x56	DTCLB
#7	DTCANSTATUS 记录 1	0x24	SODTC
#8	DTCANSTATUS 记录 2	0x23	二氯乙烷
#9	DTCANSTATUS 记录 2	0x45	二甲基氯化铵
#10	DTCANSTATUS 记录 2	0x05	DTCLB
#11	DTCANSTATUS 记录 2	0x00	SODTC
#12	DTCANSTATUS 记录 3	0xAB	二氯乙烷
#13	DTCANSTATUS 记录 3	0xcd	二甲基氯化铵
#14	DTCANSTATUS 记录 3	0x01	DTCLB
#15	DTCANSTATUS 记录 3	0x2f	SODTC

11.3.5.13 示例#12–ReadDTCInformation, sub function=reportFirstTestFailedDTC, 可用信息

11.3.5.13.1 实例#12 概述

此示例演示了 reportFirstTestFailedDTC 子功能参数的使用，其中假定自服务端上次清除诊断信息请求以来至少发生了一个失败的 DTC。

如果自服务端上次发出清除诊断信息请求以来，服务端中恰好有一个 DTC 失败，那么服务端将响应来自客户端的 reportMostRecentTestFailedDTC 请求返回相同的信息。

在这个示例中，响应 reportFirstTestFailedDTC 返回的 DTC 的状态在请求时不再是当前的（当请求服务端报告最近失败/确认的 DTC 时可能出现相同的现象）。

以下示例中的请求 / 响应消息的一般格式也适用于子功能参数 reportFirstCon.edDTC、reportMostRecentTestFailedDTC 和 reportMostRecent-Con.edDTC (对于适当的 DTC 状态和类似的假设)。

11.3.5.13.2 Example Proceedings

下列假设适用：

- ↓ 自上次从服务端发出清除诊断信息请求以来，至少有一个 DTC 失败。

↓ 服务端支持所有用于掩蔽目的的状态位。

↓ DTC 编号 0x123456=自上次代码清除以来首次检测到失败的 DTC。

↓ 以下假设适用于 DTC 0x123456, statusOfDTC 0x26(0010 0110b):

uuu) 表 327 -STATUSOFDTC=0x26

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	1	DTC 在当前操作周期中从未失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	0	DTC 从未被证实
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	1	DTC 自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

11.3.5.13.3 实例#12 消息流

在下面的示例中, DTC 0x12345 6 被返回给客户端的请求。

vvv) 表 328-ReadDTCInformation , sub
function=reportFirstTestFailedDTC, 请求消息流示例
#12

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 reportfirsttestfaileddtc=, 删除	0x0B	射频 CDTC

表 329-ReadDTCInformation, sub function=reportFirstTestFailedDTC, 肯定响应, 示例#12

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符

#1	读写信息响应	0x59	RDTCIPR
#2	reporttype = reportfirsttestfaileddtc	0x0B	射频 CDTC
#3	DTCSTATUS 可用性要求	0xff	二氯化锡
#4	DTCANSTATUS 记录	0x12	二氯乙烷
#5	DTCANSTATUS 记录[DTCMIDLE 字节]	0x34	二甲基氯化铵
#6	dtclowbyte dtcandstatusrecord []	0x56	DTCLB
#7	DTCANSTATUS 记录[STATISOFDTC]	0x26	SODTC

11.3.5.14 示例#13-ReadDTCInformation, sub function=reportFirstTestFailedDTC, 没有可用信息

11.3.5.14.1 实例#13 概述

此示例演示了 reportFirstTestFailedDTC 子功能参数的使用，其中假定自上次从服务端发出清除诊断信息请求以来没有出现失败的 DTC。

以下示例中的请求/响应消息的一般格式也适用于子功能参数 reportFirstCon. edDTC、reportMostRecentTestFailedDTC。以及 reportMostRecentCon. edDTC（用于适当的 DTC 状态并在类似的假设下）。

11.3.5.14.2 Example Proceedings

下列假设适用：

自上次从服务端发出清除诊断信息请求以来，没有发生过失败的 DTC。服务端支持所有用于掩蔽目的的状态位。

11.3.5.14.3 实例#13 消息流

在下面的示例中，没有向客户机请求返回 DTC。

www) 表 330-ReadDTCInformation , sub
function=reportFirstTestFailedDTC, 请求消息流示例
#13

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 reportfirsttestfaileddtc=, 删除	0x0B	射频 CDTC

表 331-ReadDTCInformation, sub function=reportFirstTestFailedDTC, 肯定响应，示例#13

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	reporttype = reportfirsttestfaileddtc	0x0B	射频 CDTC

#3	DTCSTATUS 可用性要求	0xff	二氯化锡
----	-----------------	------	------

11.3.5.15 示例#14-ReadDTCInformation, sub
function=reportNumberOfEmissionsOBD-DTCByStatusMask

11.3.5.15.1 实例#14 概述

这个例子演示了 reportNumberOfEmissionsOBD-DTCByStatusMask 子功能参数的使用，以及各种屏蔽原理。

11.3.15.2 例子

服务端支持所有用于掩蔽目的的状态位。此外，服务端总共支持三个与排放相关的 OBDDTC(为了简单起见!)，它在客户端请求时具有以下状态：

以下假设适用于与排放相关的 OBD DTC P0005-00 燃料切断阀“A”控制电路/打开(0x000500)，状态 OfDTC 0xAE(1010 1110b)。

xxx) 表 332——DTC P000 05-00 的 StutoSFDTC=0xAE

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	1	DTC 自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	1	服务端请求警告指示器激活 (OBD DTC)

以下假设适用于与排放相关的 OBD DTC P022F-00 中冷器旁路控制“B”电路高(0x022F00)，状态 OfDTC 0xAC(1010 1100b)。

yyy) 表 333 —DTC P022F-00 的 StutoStftc= 0xac

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	DTC 在请求时没有失败。
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	DTC 测试完成，因为最后一个代码清除
测试失败	5	1	DTC 自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期，
warningindicatorrequested	7	1	服务端请求警告指示器激活 (OBD DTC)

以下假设适用于与排放相关的 OBD DTC P0A09-00 DC/DC 转换器状态电路低输入(0x0A0900)，状态 OfDTC 0xAF(1010 1111b)。

zzz) 表 statusofdtc = 334 (DTC p0a09 00

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	1	DTC 在请求时失败
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	DTC 测试完成, 因为最后一个代码清除
测试失败	5	1	DTC 测试自上次代码清除以来至少失败一次
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	1	服务端请求警告指示器激活 (OBD DTC)

11.3.5.15.3 实例#14 消息流

在下面的示例中, 由于假设中定义的所有 DTC 都与客户机定义的状态掩码 0x08 - .edDTC(0000 1000b) 匹配, 因此向客户机返回计数 3。

aaaa) 表 335–ReadDTCInformation, sub function=reportNumberOfEmissionsOBD=DTCByStatusMask, 请求消息流示例#14

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 =reportNumberOfEmissionsOBDTCByStatus	0x12	ReNoeBoDDTCBS M
#3	DTCSTATUS 掩膜	0x08	DTCSM

表 336–ReadDTCInformation, sub function=reportNumberOfEmissionsOBD=DTCByStatusMask, 肯定响应, 示例#14

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	reporttype =	0x12	ReNoeBoDDTCBSM
#3	DTCSTATUS 可用性要求	0xff	二氯化锡
#4	DTCFrimaToistFor=SAEYJ2012-DAYDTCFATAT00	0x00	J2012-DADTCF00

#5	DTCount [DTCount] DTCount	0x00	DTCCHB
#6	[dtcountwiByte]	0x03	DTCLB

11.3.5.16 示例#15–ReadDTCInformation, sub function=reportEmissionsOBDDTC–ByStatusMask, 返回所有匹配的 OBDDTC

11.3.5.16.1 实例#15 概述

这个示例演示了 reportEmissionsOBDDTCByStatusMask 子功能参数的使用，以及结合不支持的屏蔽位的各种屏蔽原则。

11.3.5.16.2 Example Proceedings

服务端支持所有用于掩蔽目的的状态位。服务端支持总共三个 DTCs（为了简单起见！）定义在 113.5.7.2。

11.3.5.16.3 实例#15 消息流

在下面的示例中，与排放相关的 OBD DTC P0005–AE 燃料切断阀“A”控制电路/打开(0x000500)。P022F–00 中间冷却器旁路控制“B”电路高(0x022F00)和 P0A09–00DC/DC 转换器状态电路低输入(0x0A0900)被返回到客户端的请求，因为假设中定义的所有 DTC 都与客户端定义的 0x80 状态掩码匹配——警告 Indica 被请求的 (1000 0000 B)。

注意：服务端应绕过对它不支持的状态位的屏蔽。

bbbb) 表 337–ReadDTCInformation, sub function=reportEmissionsOBDDTCByStatusMask, 请求消息流示例#15

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 =reportEmissionsOBDDTCByStatusMas	0x13	罗布 TCSBSM
#3	DTCSTATUS 掩膜	0x80	DTCSTM

表 338–ReadDTCInformation, sub function=reportDTCByStatusMask, 肯定响应,
示例#15

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	reporttype =	0x13	罗布 TCSBSM
#3	DTCSTATUS 可用性要求	0xff	二氯化锡
#4	DTCANSTATUS 记录 1	0x00	二氯乙烷
#5	DTCANSTATUS 记录 1	0x05	二甲基氯化铵
#6	DTCANSTATUS 记录 1	0x00	DTCLB
#7	DTCANSTATUS 记录 1	0xAe	SODTC
#8	DTCANSTATUS 记录 2	0x02	二氯乙烷
#9	DTCANSTATUS 记录 2	0x2f	二甲基氯化铵
#10	DTCANSTATUS 记录 2	0x00	DTCLB
#11	DTCANSTATUS 记录 2	0xac	SODTC
#12	DTCANSTATUS 记录 3	0x0A	二氯乙烷
#13	DTCANSTATUS 记录 3	0x09	二甲基氯化铵
#14	DTCANSTATUS 记录 3	0x00	DTCLB
#15	DTCANSTATUS 记录 3	0xAF	SODTC

**11.3.5.17 示例#16–ReadDTCInformation, sub
function=reportEmissionsOBDDTCByStatusMask (.edDTC 和
warningIndicatorRequested), 匹配返回的 DTC**

11.3.5.17.1 实例#16 概述

这个例子演示了 reportEmissionsOBDDTCByStatusMask 子功能参数的使用。以及请求服务端报告与“确认 DTC”和“warningIndicatorRequested(MIL=ON)”状态相关的 OBDDTC 以及不支持的屏蔽位的屏蔽原则。此示例显示了与排放相关的 OBDDTC 的典型 OBD 扫描工具类型请求，该请求导致 MIL 打开，因此没有通过 I/M（检查和维护）测试。

11.3.5.17.2 Example Proceedings

为了屏蔽目的，服务端不支持位 0 (testFailed)、位 4 (testNotCompletedSinceLastClear) 和位 5 (testFailedSinceLastClear)。这就导致了 0xCE (1100×1110 b) 的 DTCSTATUS 可用性。

客户机使用值 0x88 (100010001000b) 的 DTC 状态掩码，因为只有状态为“.edDTC=1”和“warningIndicatorRequested=1”的 DTC 才会显示给技术人员。服务端支持总共三个 DTCs (为了简单起见！)，它在客户端请求时具有以下状态：

以下假设适用于 DTC P010A-14 质量或体积空气流“A”-短路接地或断开(0x010A14)，状态 OfDTC 0x00(0000 0000b)：

cccc) 表 339——DTC P010A—14 的 StutoSFDTc=0x00

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	不适用
测试失败的操作周期	1	0	DTC 在当前操作周期中从未失败
彭定克	2	0	DTC 在当前或以前的操作周期中没有失败。
确认 DTC	3	0	在请求时未确认 DTC。
核试验	4	0	不适用
测试失败	5	0	不适用
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	0	服务端不要求警告指示器激活

以下假设适用于 DTC P0180—17 燃料温度传感器 A—高于阈值的电路电压(0x018017), 0x8E 的状态 OfDTC(1000 1110b):

dddd) 表 statusofdtc = 340 0x8e (DTC p0180 17)

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	不适用
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败
确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	不适用
测试失败	5	0	不适用
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	1	服务端请求警告指示器激活 (OBD DTC)

以下假设适用于 DTC P0190—1D 燃料轨压力传感器 “A” - 超出范围的电路电流(0x01901D), 0x8E(1000 1110b) 的状态 OfDTC:

eeee) 表 341——DTC P0190—1D 的 StutoSFDTc=0x8E

statusofdtc 位字段:	比特	位状态	描述
测试失败	0	0	不适用
测试失败的操作周期	1	1	DTC 在当前操作周期失败
彭定克	2	1	DTC 在当前或以前的操作周期中失败

确认 DTC	3	1	DTC 在请求时被确认
核试验	4	0	不适用
测试失败	5	0	不适用
TestNoT 完成了这个操作循环	6	0	DTC 测试完成了这个操作周期
warningindicatorrequested	7	1	服务端请求警告指示器激活 (OBD DTC)

11.3.5.17.3 实例#16 消息流

在下面的示例中，P0180-17(0x018017) 和 P0190-1D(0x01901D) 被返回到客户端的请求。

注意：服务端应绕过对它不支持的状态位的屏蔽。

ffff) 表 342-ReadDTCInformation, sub function=reportEmissionsOBDDTCByStatusMask, 请求消息

流示例#16

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 =reportEmissionsOBDDTCByStatusMas	0x13	罗布 TCSBSM
#3	DTCSTATUS 掩膜	0x88	DTCSM

表 343-ReadDTCInformation, sub function=reportDTCTByStatusMask, 肯定响应, 示
例#16

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读写信息响应	0x59	RDTCTPR
#2	reporttype = reportemissionsobddtcbystatusmask	0x13	罗布 TCSBSM
#3	DTCSTATUS 可用性要求	0xCE	二氯化锡
#8	DTCANSTATUS 记录 1	0x01	二氯乙烷
#9	DTCANSTATUS 记录 1	0x80	二甲基氯化铵
#10	DTCANSTATUS 记录 1	0x17	DTCLB
#11	DTCANSTATUS 记录 1	0x8e	SODTC
#12	DTCANSTATUS 记录 2	0x01	二氯乙烷
#13	DTCANSTATUS 记录 2	0x90	二甲基氯化铵
#14	DTCANSTATUS 记录 2	0x1d	DTCLB
#15	DTCANSTATUS 记录 2	0x8e	SODTC

11.3.5.18 示例#17-ReadDTCInformation, 子 功能=reportDTCExtDataRecordByRecordNumber

11.3.5.18.1 实例#17 概述

此示例演示 reportDTCExtDataRecordByRecordNumber 子功能参数的使用。

11.3.5.18.2 Example Proceedings

下列假设适用：

- a) 服务端支持存储所有 DTC 的两个 DTCExtendedData 记录的能力。
- b) 假设服务端请求服务端存储的所有可用 DTCExtendedData 记录以获得记录号 0x05。
- c) 假设 DTC 0x123456 的 statusOfDTC 为 0x24，并且以下扩展数据可用于 DTC。
- d) DTCExtendedData 是通过 DTCExtDataRecordNumber 0x05 引用的
- e) 假设 DTC 0x234561 具有状态 OfDTC 0x24，并且以下扩展数据可用于 DTC。
- f) DTCExtendedData 是通过 DTCExtDataRecordNumber 0x05 引用的。

gggg) 表的内容为 0x05 344 dtcextdatarecordnumber DTC 0x123456

数据字节	dtcextdatarecord contents for dtcextdatarecordnumber 0x05	字节值
#1	预热循环计数器——自 DTC 命令 MIL 关闭以来的预热循环次数	0x17

表的内容为 0x05 345 dtcextdatarecordnumber DTC 0x234561

数据字节	dtcextdatarecord contents for dtcextdatarecordnumber 0x05	字节值
#1	预热循环计数器——自 DTC 命令 MIL 关闭以来的预热循环次数	0x79

11.3.5.18.3 实例#17 消息流

在下面的示例中，客户机请求一个 DTCMaskRecord，该 DTCMaskRecord 包括 DTC 编号和值为 0x05 的 DTCExtDataRecordNumber（报告所有 DTCExtDataRecords）。服务端返回两个 DTC，这两个 DTC 已经记录了客户机提交的 DCExtDataRecordNumber。

hhhh) 表 346–ReadDTCInformation, sub function=reportDTCExtDataRecordByRecordNumber,

请求消息流示例#7

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 reportdtcextdatarecordbyrecordnumber suppressposrspmsgindicationbit =,	0x16	RDTECEDRBDN
#3	dtcextdatarecordnumber	0x05	德特雷恩

表 347–ReadDTCInformation, sub function=reportDTCExtDataRecordByRecordNumber, 肯定响应,
示例#7

消息方向	客户端服务端		
消息类型	响应		
A_Data 字节	描述 (所有值均为十六进制)	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	reporttype =	0x16	RDTECEDRBDN
#3	DTCAndStatusRecord[DTCHighByte]	0x12	DTTCB
#4	DTCAndStatusRecord[DTCMiddleByte]	0x34	DTcmb
#5	DTCAndStatusRecord[DTCLowByte]	0x56	DTTCB
#6		0x24	
#7	dtcextdatarecordnumber	0x05	德特雷恩
#8	1 # dtcextdatarecord [字节]	0x17	EDY1
#9	DTCANSTATUS 记录	0x23	二氯乙烷
#10	DTCANSTATUS 记录[DTCMIDLE 字节]	0x45	二甲基氯化
#11	dtclowbyte dtcandstatusrecord []	0x61	DTCLB
#12	DTCANSTATUS 记录[STATISOFDTC]	0x24	SODTC
#13	dtcextdatarecordnumber	0x05	德特雷恩
#14	1 # dtcextdatarecord [字节]	0x79	EDY1

11.3.5.19 示例#18–ReadDTCInformation, 子功能= reportWHOBDTCByMaskRecord

11.3.5.19.1 示例#18 概述

这个例子演示了 reportWHOBDTCByMaskRecord 子功能参数用于已确认的 DTC (DTC 状态掩码 0x08) 的用法。车辆使用 CAN 总线连接两个与排放相关的服务端。

客户端使用以下请求参数设置:

- ↓ 功能组标识符=0x33 (排放系统组),
- ↓ DTCSeverityMaskRecord. DTCSeverityMask=0xFF (报告具有任何严重性和类状态的 DTC),
- ↓ DTCSeverityMaskRecord. DTCStatusMask=0x08 (报告 DTC 状态为“1”)。服务端支持

以下设置:

- ↓ 功能组标识符=0x33 (排放系统组),
- ↓ DTCSTATUS 可用性:
- ↓ dcStRealEdababyMaskase= 0xFF,
- ↓ DTCFrimaToistFor=SAEYJ2012-DAYDTCFATAT04= 0x04。

11.3.5.19.2 Example Proceedings

例 1 的所有假设都适用。

11.3.5.19.3 实例#18 消息流

在下面的示例中，服务端#1 仅报告 DTC P2522-1F A/C 请求“B”-电路间歇(0x25221F)，因为 0x2F(0010 1111 二进制) 的状态 OfDTC 匹配客户端定义的状态掩码 0x08(0000 1000b)。服务端#2 报告 DTC P0235-12 涡轮增压器/增压器增压传感器 “A” ——电池短路，因为 0x2E(0010 1110 b) 的状态 OfDTC 与客户端定义的状态掩码 0x08(0000 1000b) 匹配。

iiii) 表 348–ReadDTCInformation 请求，子功能=reportNumberOfDTCByStatusMask

消息方向	客户机服务端		
消息类型	请求		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读取请求信息	0x19	RDTCI
#2	子功能 =reportWWHOBDDTCByMaskRecord, .PosRspMs	0x42	RWWHBODTCBM R
#3	函数组标识符（函数组标识符=发射=0x33）	0x33	FGID
#4	dccStReMyaskField[] = [DTCSTATUS 屏蔽]	0x08	DTCSM
#5	dcStReStudiMaskField[] =	0xff	离散余弦变换支

表 349–ReadDTCInformation 响应，子功能=reportWWHOBDDTCByStatusMask

消息方向	1 客户端服务端#		
消息类型	响应		
A_Data 字节	描述（所有值均为十六进制）	字节值	助记符
#1	读写信息响应	0x59	RDTCIPR
#2	报告类型	0x42	RWWHBODTCBM R
#3	函数组标识符（函数组标识符=发射=0x33）	0x33	FGID
#4	DTCSTATUS 可用性要求	0xff	二氯化锡
#5	dTabService 可用性要求	0xff	差热扫描隧道显
#6	DTCFrimaToistFor = [SAEYJ2012–DAYDTCFATAT04]	0x04	j2012 – dadtcf04
#7	DTCANDATIORITY 记录[DTCRealthsI] 1	0x20	dtcasr _ dtcs

#8	dtand Dead StimeReal[dtChigBythy* 1]	0x25	DTCAR DTCHR DTCHB
#9	DTCAND Sead TrimeRead [DTCMIDLBYEYONEXY 1]	0x22	dtcasr _ dtcmb
#10	dtand Dead StimeReal[dtCulWByth] (1)	0x1f	dtcasr _ dtclb
#11	dtand Dead StimeRealSt[StutoSftcT] (1)	0x2f	dtcasr _ sodtc

12 输入输出控制功能单元

12.1 概述

表 351 定义了输入输出控制功能单元。

表 351——输入输出控制功能单元

服务	描述
通过标识符控制输入输出	客户端请求控制服务端特定的输入输出

12.2 通过 ID 控制输入输出（0x2F）服务

12.2.1 服务描述

客户端使用通过 **ID** 控制输入输出服务为输入信号，内部服务端功能提供替代值，和/或强制控制电气系统中的输出（执行器）值。总而言之，该服务用于相对简单（例如，静态）的输入替代/输出控制，对于更多复杂输入替代/输出控制的情况需要使用例程控制。

客户端的请求消息包括一个引用服务端输入信号，内部服务端功能，和/或输出信号（执行器）（设备控制访问可能包含一组信号）的 **ID**。控制选项记录参数应该包括所有服务端输入信号，内部功能和/或输出信号要求的信息。如果 **ID** 涉及的控制参数不止一个（即 **ID** 是打包的或者位映射的），车辆制造商可以要求请求消息中包括一个控制使能掩码。如果车辆制造商选择支持使能掩码概念，则该服务所有类型通过 **ID** 控制输入输出的请求必须有控制使能掩码参数。如果通过 **ID** 控制输入输出所请求的 **ID** 涉及一个被测量的输出值或者反馈值，则服务端应反馈一个服务端控制策略范围内的正确目标替代值，以使正常服务端控制策略尝试达到客户端请求消息期望的状态。

如果被请求的控制成功启动或者已经达到它期望的状态，则服务端应该发送一帧肯定响应消息。即使当前 **ID** 不受测试设备控制，服务端也应发送带有将控制权返回 **ECU** 的输入输出控制参数的肯定响应消息。另外，当收到将控制权返回 **ECU** 的请求后，服务端应该为客户端提供将控制掩码（如果支持）位全部置‘1’的能力，以将打包或者位映射 **ID** 的控制权完全返回给 **ECU**。请求消息 **ID** 的控制选项记录参数中输入输出控制参数之后的控制状态字节的格式和长度应该与被请求 **ID** 数据记录的长度和格式完全匹配。该方式应确保使用带有相同 **DID** 的通过 **DID** 读取数据服务能够获取实际的输出或输入状态。

当使用通过 **ID** 控制输入输出服务执行输入替代或者输出控制时，**ECU** 收到请求后会有两个基本要求。第一个是断开 **ID** 参数上游控制策略的相应数据对象，否则将更新数据对象值。第二个是替代控制策略所有下游活动所使用的相应数据对象值。例如，测试设备请求直接强制点亮头灯，需要阻止头灯开关位置影响头灯的输出，并且将需求状态为“ON”的替代值赋给最终决定头灯状态期望输出功能所使用的数据对象。

该服务允许单个请求消息中单个 **ID** 和它相应的参数的控制。如果这样做，服务端应响应一帧响应消息，该消息包含请求消息的 **DID** 和控制状态信息。

重要提示——服务端和客户端的请求和响应消息行为应该遵循 7.5 中的定义。

12.2.2 请求消息

12.2.2.1 请求消息定义

表 352——请求消息定义

A_Data 类型	参数名称	Cvt	字节值	助记符
#1	通过 ID 控制输入输出请求 SID	M	0x2F	IOCBI

#2 #3	DID[]=[Byte#1(MSB) Byte#2(LSB)]	M M	0x00-0xFF	IOI_ B1 B2
#4 . . #4+(m-1)	控制选项记录[]=[输入输出参数 控制状态#1 . . 控制状态#m]	M_1 C_1 . . C_1	0x00-0xFF 0x00-0xFF . . 0x00-0xFF	CSR_ IOCP_ CS_ . . CS_
#4+m . #4+m+(r-1)	控制使能掩码记录#1[]=[控制掩码#1 . . 控制掩码#r]	C_2 . . C_2	0x00-0xFF . . 0x00-0xFF	CEM_ CM_ . . CM_
M_1 输入输出控制参数详细定义见 E.1。 C_1 : 该参数的存在与否取决于 DID 和输入输出控制参数 (见 E.1)。 C_2 : 如果车辆制造商支持控制使能掩码的概念, 如果 DID 有多个参数组成则应该包含该参数 (见控制使能掩码记录定义)				

12.2.2.2 请求消息子功能参数\$Level(LEV_)定义

该服务不使用子功能参数。

12.2.2.3 请求消息数据参数定义

该服务的数据参数定义如下:

表 353——请求消息数据参数定义

定义
DID 该参数确定一个服务端本地输入信号, 内部参数和/或输出信号。该参数值的可用范围参见附录 C.1 定义的 DID 表。
控制选项记录 控制选项记录包含一个或多个字节 (输入输出控制参数和控制状态#1 至控制状态#m)。控制选项记录参数的细节定义在附录 E.1 中。
控制使能掩码记录 控制选项记录有一个或多个字节 (控制掩码#1 至控制掩码#r) 组成。仅当被控制的 DID 包含一个以上参数时, 支持控制使能掩码记录 (即 DID 定义为位映射或打包的)。DID 中定义的每个参数都要在控制使能掩码记录中有对应位。被控制的 DID 仅包含一个参数时, 不应支持控制使能掩码记录。 注 DID 中的每个参数可以是任意数目的位。 控制使能掩码记录中的每一位应该决定 DID 中的相应参数是否会被请求影响。控制使能掩码记录位值为 '0' 代表对应参数不受该请求的影响, 一个位的值为 '1' 应该代表对应参数受该请求影响。控制掩码#1 的最高有效位对应控制状态最高有效位控制状态#1 的第一个参数, 控制掩码#1 的次高有效位应该对应控制状态的第二个参数, 通过这种方式使用必要数量的控制掩码字节以标记所有参数。例如, 控制掩码#2 的最低有效位应该对应控制状态中的第 16 个参数。对于位映射 DIDs, 不支持的位也应在控制使能掩码记录中有对应的位, 以便控制使能掩码记录中的每个参数掩码位的位置都能与控制状态中相应参数的位置正确对应。

12.2.3 肯定响应消息

12.2.3.1 肯定响应消息定义

表 354——肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
-----------	------	-----	-----	-----

#1	通过 ID 控制输入输出的响应 SID	M	0x6F	IOCBIPR
#2 #3	DID[]=[字节#1 (MSB) 字节#2 (LSB)]	M M	0x00-0xFF 0x00-0xFF	IOI_ B1 B2
#4 #5 . . #5+(m-1)	控制状态记录[]=[输入输出控制参数 控制状态#1 . . 控制状态#m]	M <i>C₁</i> . <i>C₁</i>	0x00-0xFF 0x00-0xFF . 0x00-0xFF	CSR_ IOCP_ CS_ . CS_
<i>C₁</i> : 该参数的存在与否取决于 DID 和输入输出控制参数 (见 E.1)。				

12.2.3.2 肯定响应消息数据参数定义

表 355——响应消息数据参数定义

定义
DID 该参数与请求消息中的 DID 相同。
控制状态记录 控制状态参数由多个字节 (输入输出控制参数和控制状态#1 至控制状态#m) 组成, 例如反馈数据。控制状态记录参数细节定义在附录 E.1。

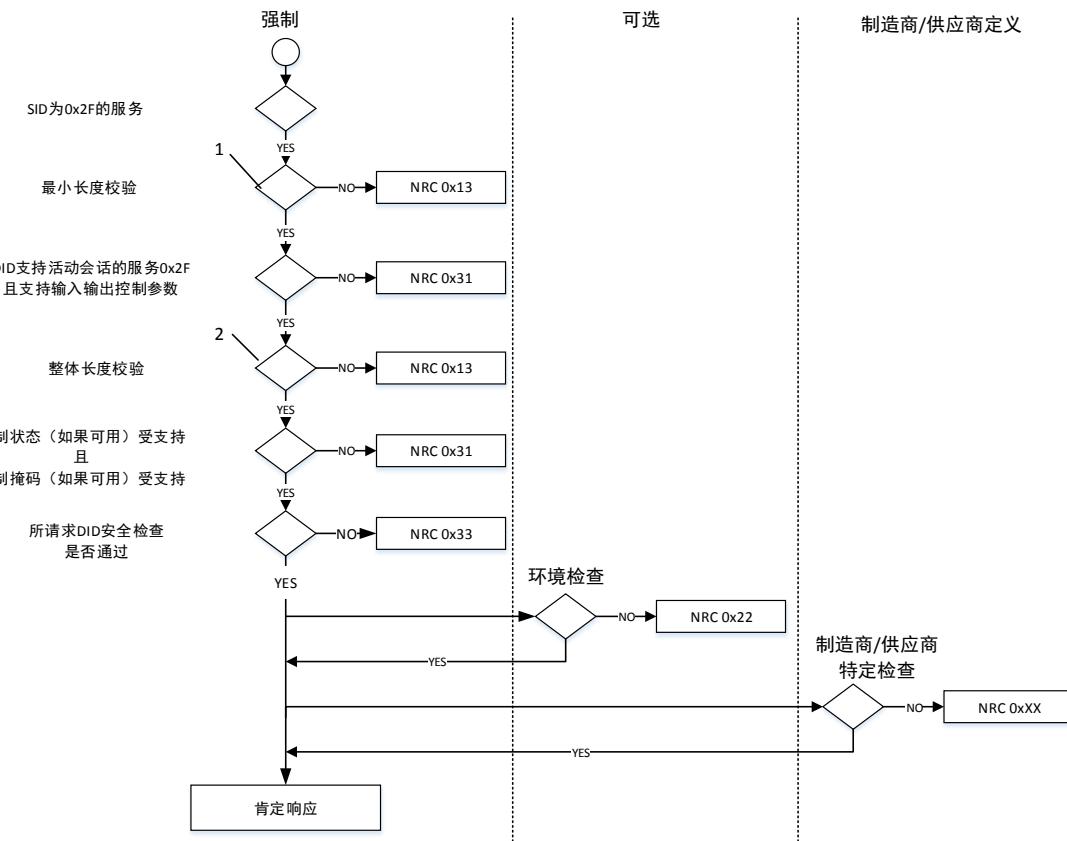
12.2.4 受支持的否定响应码 (NRC_)

该服务应执行以下否定响应码。表 356 给出了每个响应码发生的场景。如果故障方案适用于服务端, 则使用所列的否定响应。

表 356——支持的否定响应码

NRC	描述	助记符
0x13	消息长度错误或者无效格式 消息长度错误时发送该 NRC。	IMLOIF
0x22	条件不正确 请求输入输出控制的条件未满足时返回该 NRC。	CNC
0x31	请求超出范围 该 NRC 在以下情况发送: —设备不支持所请求的 DID 值; —输入输出控制参数中的值无效 (见输入输出控制参数定义); —控制选项记录中的一个或多个可用控制状态值无效; —设备不支持控制使能掩码记录的使能控制位组合。	ROOR
0x33	安全访问被拒绝 客户端发送一个有效安全的 DID 且服务端安全功能当前为激活状态则返回该 NRC。	SAD

评估序列的编档见图 24。



图例

- 2 至少 4 (SI+DID+IOCP)
- 3 If IOCP=shortTermAdjustment, 1 字节 SI+2 字节 DID+1 字节 IOCP+第 n 个字节控制状态+第 n 个字节控制掩码 (如果适用),
If IOCP<>shortTermAdjustment, 1 字节 SI+2 字节 DID+1 字节 IOCP+第 n 字节控制掩码 (如果适用)。

图 24——通过 ID 控制输入输出服务的 NRC 处理

12.2.5 通过 ID 控制输入输出消息流示例

12.2.5.1 假设

下个示例显示了 HVAC 控制模块如何使用通过 ID 控制输入输出服务，假设在单个服务端内执行物理通信。

12.2.5.2 示例#1-“进气门位置”短期调整

受控制的参数是 DID (0x9B00) 相关的“进气门位置”。

换算：进气门位置[%]=十进制 (Hex) *1[%]

12.2.5.2.1 步骤#1：通过 ID 读数据

该示例使用通过 ID 读取数据服务读取进气门位置的当前状态。

表 357——通过 ID 读取数据请求消息流示例#1-步骤#1

消息方向	客户端 ————— 服务端		
消息类型	请求		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x22	RDBI

#2	DID[字节#1]=0x9B	0x9B	DID_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	DID_B2

表 358——通过 ID 读取数据肯定响应消息流示例#1-步骤#1

消息方向	服务端 —— 客户端		
消息类型	响应		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	通过 ID 请求数据的响应 SID	0x62	RDBIPR
#2	DID[字节#1]=0x9B	0x9B	DID_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	DID_B2
#4	数据记录[数据#1]=10%	0x0A	DREC_DAT1

12.2.5.2.2 步骤#2：短期调整

表 359——通过 ID 控制输入输出的请求消息流示例#1-步骤#2

消息方向	客户端 —— 服务端		
消息类型	请求		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x2F	IOCBPI
#2	DID[字节#1]=0x9B	0x9B	IOI_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	IOI_B2
#4	控制选项记录[控制输入输出参数]=短期调整	0x03	IOCP_STA
#5	控制选项记录[控制状态#1]=60%	0x3C	CS_1

表 360——通过 ID 控制输入输出肯定响应消息流示例#1-步骤#2

消息方向	服务端 —— 客户端		
消息类型	响应		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x6F	IOCBIPR
#2	DID[字节#1]=0x9B	0x9B	IOI_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	IOI_B2
#4	控制选项记录[控制输入输出参数]=短期调整	0x03	IOCP_STA
#5	控制选项记录[控制状态#1]=12%	0x0C	CS_1

注 客户端已经按照上述规定发送一帧通过 ID 请求输入输出请求消息。服务端立即完成肯定响应的发送，肯定响应消息中带有“进气门状态”控制状态参数为 12%。进气门需要一定的时间调整到被请求的 60%。

12.2.5.2.3 步骤#3：通过 ID 读数据

该示例使用通过 ID 读取数据服务读取进气门位置的当前状态。

表 361——通过 ID 读取数据请求消息流示例#1-步骤#3

消息方向	客户端 —— 服务端		
消息类型	请求		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x22	RDBI
#2	DID[字节#1]=0x9B	0x9B	DID_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	DID_B2

表 362——通过 ID 请求数据肯定响应消息流示例#1-步骤#3

消息方向	服务端 —— 客户端
------	------------

消息类型		响应	
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	通过 ID 请求数据的响应 SID	0x62	RDBIPR
#2	DID[字节#1]=0x9B	0x9B	DIID_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	DIID_B2
#4	数据记录[数据#1]=60%	0x3C	DREC_DAT1

注 当通过 ID 控制输入输出激活时，客户端按照上述规定完成通过 ID 读取数据的请求消息的发送。它将给服务端的控制策略一定时间，以最终达到期望值。上述示例出现在服务端最终到达期望目标值时。

12.2.5.2.4 步骤#4：将控制权返回 ECU

表 363——通过 ID 控制输入输出的请求消息流示例#1-步骤#4

消息方向		客户端 ————— 服务端	
消息类型		请求	
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x2F	IOCBP
#2	DID[字节#1]=0x9B	0x9B	IOI_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	IOI_B2
#4	控制选项记录[控制输入输出参数]=将控制权返回 ECU	0x00	RCTECU

表 364——通过 ID 控制输入输出肯定响应消息流示例#1-步骤#4

消息方向		服务端 ————— 客户端	
消息类型		响应	
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x6F	IOCBP
#2	DID[字节#1]=0x9B	0x9B	IOI_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	IOI_B2
#4	控制选项记录[控制输入输出参数]=将控制权返回 ECU	0x00	RCTECU
#5	控制状态记录[控制状态#1]=58%	0x3A	CS_1

12.2.5.2.5 步骤#5：冻结当前状态

表 365——通过 ID 控制输入输出的请求消息流示例#1-步骤#5

消息方向		客户端 ————— 服务端	
消息类型		请求	
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x2F	IOCBP
#2	DID[字节#1]=0x9B	0x9B	IOI_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	IOI_B2
#4	控制选项记录[控制输入输出参数]=冻结当前状态	0x02	IOCP_FCS

表 366——通过 ID 控制输入输出的肯定响应消息流示例#1-步骤#5

消息方向		服务端 ————— 客户端	
消息类型		响应	
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x6F	IOCBP
#2	DID[字节#1]=0x9B	0x9B	IOI_B1
#3	DID[字节#2]=0x00 (“进气门位置”)	0x00	IOI_B2

#4	控制选项记录[控制输入输出参数]=冻结当前状态	0x02	IOCP_FCS
#5	控制状态记录[控制状态#1]=50%	0x32	CS_1

12.2.5.3 示例#2——EGR 和 IAC 短期调整

12.2.5.3.1 假定

该示例使用打包的 DID 0x0155 展示对一个请求中单个参数或者多个参数进行控制。

该子条款规定了一个短期调整功能的测试环境和与示例 DID 0x0155 相关的消息流。DID 支持下表 367 中描述的五个单独参数。

表 367——混合数据块-DID 定义——示例#2

DID	数据字节	参数		数据记录内容
		编 号	大小	
0x0155	#1 (所有位)	#1	8 位	数据记录[数据#1]=IAC 枢轴位置 (n=计数)
	#2-#3(所 有 位)	#2	16 位	数据记录[数据#2-#3]=RPM (0=0U/min, 65535U/min)
	#4 (bits7-4)	#3	4 位	数据记录[数据#4 (bits 7-4)]=踏板位置 A: 线性比例, 0=0%, 15=120%
	#4 (bits3-0)	#4	4 位	数据记录[数据#4 (bits 7-4)]=踏板位置 B: 线性比例, 0=0%, 15=120%
	#5 (所有位)	#5	8 位	数据记录[数据#5]=EGR 工作周期: 线性比例, 0 计数=0%, 255 计数=100%

DID 0x0155 由定义打包并且由五个基本参数组成。对于单个控制目的，这些基本参数每一个都是可以通过控制使能掩码记录中的单个位进行选择。如果 DID 未定义成打包或者位映射，则请求消息中不显示控制使能掩码记录。控制掩码#1 的最高有效位通常与控制状态#1 最高有效位的 DID 首个参数对应。具体展示见表 368.

表 368——控制使能掩码记录——示例#2

DID 0x0155 控制使能掩码记录。	
总大小=1 字节 (即, 仅包含控制使能掩码#1)	
位排布	控制使能掩码#1-位含义 (1=受影响, 0=不受影响)
7 (最高有效位)	决定参数#1 (IAC 枢轴位置) 是否被请求影响
6	决定参数#2 (RPM) 是否被请求影响
5	决定参数#3 (踏板位置 A) 是否被请求影响
4	决定参数#4 (踏板位置 B) 是否被请求影响
3	决定参数#5 (EGR 工作周期) 是否被请求影响
2	由于无对应参数故不影响
1	由于无对应参数故不影响
0 (最低有效位)	由于无对应参数故不影响

12.2.5.3.2 示例#1：仅控制 IAC 枢轴位置

表 369 定义了同故宫 ID 控制输入输出的请求消息流示例#2-案例 1。

表 369——通过 ID 控制输入输出的请求消息流示例#2-案例#1

消息方向	客户端 ————— 服务端		
消息类型	请求		
A_Data 类型	描述 (所有值都为十六进制)	字节值	助记符
#1	通过 ID 控制输入输出的请求 SID	0x2F	IOCBI
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3		0x55	IOI_B2

	DID[字节#2]=0x055 (IAC/RPM/PPA/PPB/EGR)		
#4	控制选项记录[输入输出控制参数]=短期调整	0x03	IOCP_STA
#5	控制选项记录[控制状态#1]=IAC 枢轴位置 (7 计数)	0x07	CS_1
#6	控制选项记录[控制状态#2]=RPM (XX)	0xXX	CS_2
#7	控制选项记录[控制状态#3]=RPM (XX)	0xXX	CS_3
#8	控制选项记录[控制状态#4]=踏板位置 A (Y) 和 B(Z)	0xYZ	CS_4
#9	控制选项记录[控制状态#5]=EGR 工作周期 (XX)	0xXX	CS_5
#10	控制使能掩码[控制掩码#1]=仅控制 IAC 枢轴位置	80	CM_1

注 由于状态掩码#1 参数定义了只有 DID 中第一个参数被请求影响，所以控制状态#2-#5 中传输的 RPM，踏板位置 A，踏板位置 B，和 EGR 工作周期值是无效的。

表 370——通过 ID 控制输入输出的肯定响应消息流示例#2-案例#1

消息指向	服务端 ——> 客户端		
消息类型	响应		
A_Data 类型	描述 (所有值都是十六进制)	字节值	助记符
#1	通过 ID 控制输入输出的响应 SID	0x6F	IOCBLIPR
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3	DID[字节#2]=0x55 (IAC/RPM/PPA/PPB/EGR)	0x55	IOI_B2
#4	控制状态记录[输入输出控制参数]=短期调整	0x03	IOCP_STA
#5	控制状态记录[控制状态#1]=IAC 枢轴位置 (7 计数)	0x07	CS_1
#6	控制状态记录[控制状态#2]=RPM(750 U/min)	0x02	CS_2
#7	控制状态记录[控制状态#3]=RPM	0xEE	CS_3
#8	控制状态记录[控制状态#4]=踏板位置 A(8%)B(16%)	0x12	CS_4
#9	控制状态记录[控制状态#5]=EGR 工作周期 (35%)	0x59	CS_5

注 控制状态#1-控制状态#5 所有参数值都应该反映系统当前状态。

12.2.5.3.3 案例#2：仅控制 RPM

表 371 定义了通过 ID 控制输入输出请求消息流示例#2-案例#2。

表 371——通过 ID 控制输入输出请求消息流示例#2-案例#2

消息指向	客户端 ——> 服务端		
消息类型	请求		
A_Data 类型	描述 (所有值都是十六进制)	字节值	助记符
#1	通过 ID 控制输入输出的请求 SID	0x2F	IOCBP
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3	DID[字节#2]=0x55 (IAC/RPM/PPA/PPB/EGR)	0x55	IOI_B2
#4	控制状态记录[输入输出控制参数]=短期调整	0x03	IOCP_STA
#5	控制状态记录[控制状态#1]=IAC 枢轴位置 (XX 计数)	0xXX	CS_1
#6	控制状态记录[控制状态#2]=RPM(0x3E8=1000 U/min)	0x03	CS_2
#7	控制状态记录[控制状态#3]=RPM	0xE8	CS_3
#8	控制状态记录[控制状态#4]=踏板位置 A(Y)B(Z)	0xYZ	CS_4
#9	控制状态记录[控制状态#5]=EGR 工作周期 (XX)	0xXX	CS_5
#10	控制使能掩码[控制掩码#1]=仅控制 RPM	0x40	CM_1

注 控制状态#1 和控制状态#4-#5 中传输的 IAC 枢轴位置，踏板位置 A，踏板位置 B 和 EGR 工作周期值无效，因为控制掩码#1 参数定义了请求仅影响 DID 中第二个参数。

表 372 定义了通过 ID 控制输入输出肯定响应消息流示例#2-案例#2。

表 372——通过 ID 控制输入输出肯定响应消息流示例#2-案例#2

消息指向	服务端 ——> 客户端
------	-------------

消息类型		响应	
A_Data 类型	描述 (所有值都是十六进制)	字节值	助记符
#1	通过 ID 控制输入输出的请求 SID	0x6F	IOCBLIPR
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3	DID[字节#2]=0x55 (IAC/RPM/PPA/PPB/EGR)	0x55	IOI_B2
#4	控制状态记录[输入输出控制参数]=短期调整	0x03	IOCP_STA
#5	控制状态记录[控制状态#1]=IAC 枢轴位置 (7 计数)	0x09	CS_1
#6	控制状态记录[控制状态#2]=RPM(950 U/min)	0x03	CS_2
#7	控制状态记录[控制状态#3]=RPM	0Xb6	CS_3
#8	控制状态记录[控制状态#4]=踏板位置 A(8%)B(16%)	0x12	CS_4
#9	控制状态记录[控制状态#5]=EGR 工作周期 (35%)	0x59	CS_5

注 控制状态#1-控制状态#5 传输的所有参数都要反映系统当前状态。

12.2.5.3.4 案例#3：同时控制踏板位置 A 和 EGR 工作周期

表 373 定义了通过 ID 控制输入输出请求消息流示例#2-案例#3

表 373——通过 ID 控制输入输出请求消息流示例#2-案例#3

消息指向		客户端 —— 服务端	
消息类型		请求	
A_Data 类型	描述 (所有值都是十六进制)	字节值	助记符
#1	通过 ID 控制输入输出的请求 SID	0x2F	IOCBI
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3	DID[字节#2]=0x55 (IAC/RPM/PPA/PPB/EGR)	0x55	IOI_B2
#4	控制状态记录[输入输出控制参数]=短期调整	0x03	IOCP_STA
#5	控制状态记录[控制状态#1]=IAC 枢轴位置 (XX)	0xXX	CS_1
#6	控制状态记录[控制状态#2]=RPM(XX)	0xXX	CS_2
#7	控制状态记录[控制状态#3]=RPM(XX)	0xXX	CS_3
#8	控制状态记录[控制状态#4]=踏板位置 A(0x3=24%), B(Z)	0x3Z	CS_4
#9	控制状态记录[控制状态#5]=EGR 工作周期 (45%)	0x72	CS_5
#10	控制使能掩码[控制掩码#1]=控制踏板位置 A 和 EGR	28	CM_1

注 控制状态#1-#3 和控制状态#4 (位 3-0) 中传输的 IAC 枢轴位置, RPM 和踏板位置 B 值无效, 因为控制掩码#1 参数定义了请求仅影响 DID 中的第三个和第四个参数。

表 374 定义了通过 ID 控制输入输出肯定响应消息流示例#2-案例#3。

表 374——通过 ID 控制输入输出肯定响应消息流示例#2-案例#3

消息指向		服务端 —— 客户端	
消息类型		响应	
A_Data 类型	描述 (所有值都是十六进制)	字节值	助记符
#1	通过 ID 控制输入输出的响应 SID	0x6F	IOCBI
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3	DID[字节#2]=0x55 (IAC/RPM/PPA/PPB/EGR)	0x55	IOI_B2
#4	控制状态记录[输入输出控制参数]=短期调整	0x03	IOCP_STA
#5	控制状态记录[控制状态#1]=IAC 枢轴位置 (7 计数)	0x07	CS_1
#6	控制状态记录[控制状态#2]=RPM(850 U/min)	0x03	CS_2
#7	控制状态记录[控制状态#3]=RPM	0x52	CS_3
#8	控制状态记录[控制状态#4]=踏板位置 A(24%)B(16%)	0x32	CS_4
#9	控制状态记录[控制状态#5]=EGR 工作周期 (41%)	0x69	CS_5

注 控制状态#1-控制状态#5 所有参数传输值应该反映系统的当前状态。

12.2.5.3.5 案例#4：将所有参数的控制权返回 ECU

表 375 定义了通过 ID 控制输入输出请求消息流的示例#2-案例#4。

表 375——通过 ID 控制输入输出请求消息流示例#2-案例#4

消息方向	服务端 ——> 客户端		
消息类型	请求		
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x2F	IOCB1
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3	DID[字节#2]=0x55 (IAC/RPM/PPA/PPB/EGR)	0x55	IOI_B2
#4	控制选项记录[控制输入输出参数]=将控制权返回 ECU	0x00	RCTECU
#5	控制使能掩码[控制掩码#1]=所有基本参数	0xFF	CM_1

表 376 定义了通过 ID 控制输入输出肯定响应消息流示例#2-案例#4。

表 376——通过 ID 控制输入输出肯定响应消息流示例#2-案例#4

消息方向	服务端 ——> 客户端		
消息类型	响应		
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	通过 ID 请求数据的请求 SID	0x6F	IOCBIPR
#2	DID[字节#1]=0x01	0x01	IOI_B1
#3	DID[字节#2]=0x55 (IAC/RPM/PPA/PPB/EGR)	0x55	IOI_B2
#4	控制状态记录[输入输出控制参数]=将控制权返回 ECU 控制状态记录[控制状态#1]=IAC 枢轴位置 (9 计数)	0x00	IOCP_STA
#5	控制状态记录[控制状态#2]=RPM(850 U/min)	0x09	CS_1
#6	控制状态记录[控制状态#3]=RPM	0x03	CS_2
#7	控制状态记录[控制状态#4]=踏板位置 A(8%)B(16%)	0x52	CS_3
#8	控制状态记录[控制状态#5]=EGR 工作周期 (35%)	0x12	CS_4
#9		0x59	CS_5

注 控制状态#1-控制状态#5 所有参数的传输值应反映系统的当前状态。

13 例程功能性单元

13.1 概述

表 377 定义了例程功能性单元

表 377——例程功能性单元

服务	描述
例程控制	客户端请求开启，停止服务端中的例程或者请求例程结果。

该功能性单元指定了需要在服务端和客户端中执行的远程激活例程的服务。以下子条款描述了两种不同的实现方法（方法“A”和“B”）。也可能有其他实现方法。方法“A”和“B”应该被用作实现例程服务的指南。

注 每个方法都有在例程结束之后请求例程结果服务的功能。方法的选择和实现由车辆制造商和系统供应商决定。

以下是对方法“A”和“B”的简短介绍：

——方法“A”：

- 该方法基于假定客户端已经启动服务端内存中的例程后，客户端须负责停止例程的情况。
- 在完成例程控制请求消息和第一帧响应消息（如果“肯定”基于服务端环境）之间的某个时间启动服务端内存中的例程。
- 在完成停止例程请求消息和第一帧响应消息（如果“肯定”基于服务端环境）之间的某个时间停止服务端内存中的例程。
- 客户端可在例程停止之后请求例程结果。

——方法“B”：

- 该方法基于假定客户端已经启动服务端内存中的例程，客户端须负责停止例程的情况。
- 在完成例程控制请求消息和第一帧响应消息（如果“肯定”基于服务端环境）之间的某个时间启动服务端内存中的例程。
- 服务端应按内存中的编程或者之前初始化随时停止例程。

13.2 例程控制（0x31）服务

13.2.1 服务描述

客户端使用例程控制服务执行特定步骤序列并且获得任何相关结果。该服务具有较大灵活性，但一般用途包括擦除内存，重置或学习自适应数据，运行自测试，覆盖正常服务端控制策略，控制服务端值随时间变化以及预定义序列（比如关闭敞篷车顶）等。一般情况下，使用该服务控制输出可以进行更复杂类型的控制，而通过 ID 控制输入输出用于相对简单的控制（例如，静态控制）。

13.2.1.1 概述

客户端使用例程控制服务进行如下操作：

- 开启例程
- 停止例程，
- 请求例程结果

使用 2 字节例程 ID 引用例程。

以下子条款规定了通过例程 ID 引用开启例程，停止例程和请求例程结果。

重要提示——服务端和客户端应该满足 7.5 中定义的请求和响应消息行为。

13.2.1.2 启动例程 ID 引用的例程

在完成启动例程请求消息和第一帧响应消息之间的某个时间启动服务端内存中的例程，如果响应消息是肯定或否定，则指示请求已经执行或者正在执行中。

例程可以是代替正常操作代码运行的测试，也可以是使能和执行正常操作代码运行的例程。特别第一种案例，可能需要使用诊断会话控制服务将服务端切换到特定的诊断会话或者使用安全访问服务解锁服务端以使用启动例程服务。

13.2.1.3 停止例程 ID 引用的例程

在完成停止例程请求消息和第一帧响应消息之间的某个时间停止服务端内存中的例程，如果响应消息是肯定或否定，则指示请求停止例程已经执行或者正在执行。

13.2.1.4 请求例程 ID 引用的例程结果

客户端使用该子功能请求 RID 引用的结果（例如，退出状态信息），该结果由服务端内存中执行的例程产生。若停止例程子功能的肯定响应消息收到例程结果（例如正常/异常退出结果），则须使用请求例程结果子功能。举个例子，例程结果可能是服务端收集的数据，而该数据由于服务端性能限制无法在例程执行过程中发送。

13.2.2 请求消息

13.2.2.1 请求消息定义

表 378——请求消息定义

A_Data 类型	参数名称	Cvt	字节值	助记符
#1	例程控制请求 SID	M	0x31	RC
#2	子功能=[例程控制类型]	M	0x00-0xFF	LEV_ RCTP_
#3 #4	例程 ID[]=[字节#1（MSB） 字节#2（LSB）]	M M	0x00-0xFF 0x00-0xFF	RI_ B1 B2
#5	例程控制选项记录[]=[例程控制选项#1]	C/U	0x00-0xFF	RCEOR_ RCO_

#n	例程控制选项#m]	C/U	0x00-0xFF	RCO_
C: 用户可以在子功能为启动例程和停止例程时使用该参数。				

13.2.2.2 请求消息子功能参数\$Level (LEV_) 定义

该服务使用子功能参数选择对例程的控制。有关可能级别的说明和用途，参见表 379 (抑制肯定响应位 (bit7) 没有显示)。

表 379——请求消息子功能定义

Bits6-0	描述	Cvt	助记符
0x00	ISOSAEReserved 该值为本文件为未来定义预留。	M	ISOSAERESRVD
0x01	启动例程 该参数定义了服务端应启动 RID 定义的例程。	M	STR
0x02	停止例程 该参数定义了服务端应停止 RID 定义的例程。	U	STPR
0x03	请求例程结果 该参数定义了服务端应返回 RID 定义的例程结果值。	U	RRR
0x04-0x7F	ISOSAEReserved 该值为本未见为未来定义预留。	M	ISOSAERESRVD

13.2.2.3 请求消息数据-参数定义

表 380 定义了请求消息中的数据参数。

表 380——请求消息中数据参数的定义

定义
例程 ID 该参数确定一个服务端的本地例程，并且在数据标识符范围之外（见附录 F.1.）。
例程控制选项记录 该参数包含下述两者之一： —例程进入参数，可选择性地定义例程的启动条件（例如，启动时间，变量启动等）， —例程退出参数，可选择性地定义例程的停止条件（例如例程停止前的超时时间，参数等）。

13.2.3 肯定响应消息

13.2.3.1 肯定响应消息定义

表 381 定义了肯定响应消息。

表 381——肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	例程控制响应 SID	M	0x71	RCPR
#2	例程控制类型	M	00-7F	RCTP_
#3	例程标识符[]=[字节#1 (MSB) 字节#2 (LSB)]	M	0x00-0xFF	RI_ B1
#4		M	0x00-0xFF	B2
#5	例程信息	C ₁	0x00-0xFF	RINF_
#6	例程状态记录[]=[例程状态#1 . . . 例程状态#m]	U	0x00-0xFF	RSR_ RS_
#n		U	0x00-0xFF	RS_

C₁ 例程信息字节定义了一个计划表（例如，启动例程，停止例程，请求例程结果），允许一般外部测试

设备对任何例程按此进行处理。在 ISO/SAE 规范（例如 ISO 27145-3, SAE J1979-DA, ISO 26021）定义了例程状态记录的情况下，必须有该参数（即使 ISO/SAE 定义的例程状态记录为“0”字节）。如果例程状态记录完全由车辆制造商定义，则可以选择性地支持该参数。该字节应该由车辆制造商定义。
U 仅在车辆制造商为例程标识符 (RID) 进行定义的情况下，例程状态记录[]才包含例程状态字节。

表 382——响应消息数据参数定义

定义
例程控制类型 该参数是与请求消息中子功能参数 bits6-0 位相同。
例程标识符 该参数与请求消息中例程标识符相同。
例程信息 例程信息字节编码由车辆制造商定义，为车辆制造商支持一般外部测试设备基于该返回值处理所有完成的例程（例如，如果要求停止例程或者请求例程结果）提供一种方法。
例程状态记录 该参数记录用于为客户端提供以下三者之一： ——启动例程之后服务端状态的额外信息； ——停止例程之后服务端状态的额外信息（例如，总运行时间，例程停止前产生的结果）； ——服务端之前停止例程的结果（退出状态信息）。

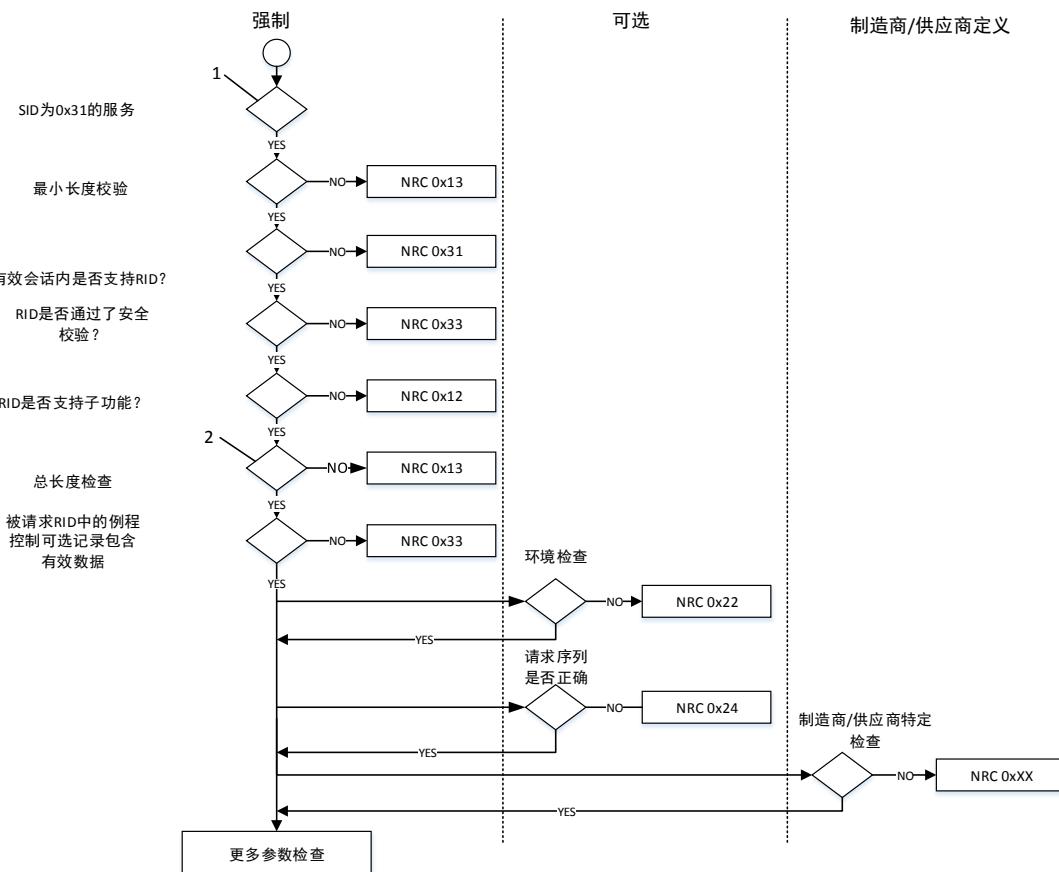
13.2.4 支持的否定响应码 (NRC_)

该服务应执行以下否定响应码。每个响应码发生的情形都定义在表 383 中。如果错误场景适用服务端则使用列表中的否定响应。

表 383——受支持的否定响应码

NRC	描述	助记符
0x12	子功能不支持 若被请求的子功能不被支持或者对于所请求的 RID 不支持子功能，则发送该 NRC。	SFNS
0x13	错误消息长度或无效格式 如果消息长度错误则发送该 NRC。	IMLOIF
0x22	条件不正确 如果请求例程控制的条件不满足则回复该 NRC。	CNC
0x24	请求序列错误 在以下情况返回该 NRC： •当收到‘开始例程’子功能，当前例程激活且不能重启（这取决于车辆制造商是否能够重启正在激活的例程）， •收到‘停止例程’子功能时，当前例程未激活。 •收到‘请求例程结果’子功能时，例程结果不可得（例如请求例程 ID 没有开启）	RSE
0x31	请求超出范围 在以下情况发送该 NRC： ——服务端不支持被请求的例程 ID， ——被请求例程 ID 的用户可选例程控制选项记录包含无效数据。	ROOR
0x33	安全访问被拒绝 当客户端发送了一个带有有效安全例程 ID 的请求，并且服务端安全功能当前为激活状态时，则发送该 NRC。	SAD
0x72	一般编程故障 当服务端执行一个访问服务端内部内存的例程时检测到错误，则会返回该 NRC。例如，例程擦除或者编程一个永久内存设备（例如，Flash 内存）中特定内存区域时，访问内存区域失败。	GPF

诊断序列编档如图 25。



图例

- 1 至少 4 个字节 (SI+子功能+RID 参数)
- 2 特定 RID 所需的 1 字节 SI+1 字节 SF+2 字节 RID+第 n 字节例程控制选项记录

图 25——例程控制服务的 NRC 处理

13.2.5 例程控制消息流示例

13.2.5.1 示例#1：子功能=启动例程

该子条款说明了服务端中启动例程的测试条件，在测试员“扭动”所有受测试系统线束接头时，持续间歇性地（尽可能快）测试所有输入输出信号。使用例程 ID 0x0201 引用该例程。

测试环境：点火状态=on，发动机状态=off，车速=0[kph]

客户端通过设置抑制肯定响应位（子功能参数的 bit7）为“假”（‘0’）请求响应消息。

表 384 定义了例程控制请求消息流-示例#1。

表 384——例程控制请求消息流-示例#1

消息方向		客户端 ————— 服务端	
消息类型		请求	
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	例程控制的请求 SID	0x31	RC
#2	子功能=开启例程，	0x01	LEV_STR

	抑制肯定响应位=假		
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x01	RI_B2

表 385 定义了肯定响应消息流-示例#1。

表 385——肯定响应消息流-示例#1

消息方向	服务端 ——> 客户端		
消息类型	响应		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	例程控制的响应 SID	0x71	RCPR
#2	子功能=开启例程	0x01	STR
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x01	RI_B2
#5	例程状态记录[例程状态#1]=车辆制造商定义	0x32	RRS_

13.2.5.2 示例#2：子功能=停止例程

该子条款规定了停止服务端内例程的测试环境，当技术人员“扭动”所有受测试系统线束接头时，持续间歇性地（尽可能快）测试所有输入输出信号。通过例程 ID 0x0201 引用该例程。

测试环境：点火状态=on，发动机状态=off，车速=0[kph]

客户端通过设置抑制肯定响应位（子功能参数的 bit7）为“假”（‘0’）请求响应消息。

表 386 定义了例程控制请求消息流-示例#2。

表 386——例程控制请求消息流示例#2

消息方向	客户端 ——> 服务端		
消息类型	请求		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	例程控制的响应 SID	0x31	RC
#2	子功能=停止例程， 抑制肯定响应位=假	0x02	SPR
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x01	RI_B2

表 387 定义了例程控制肯定响应消息流-示例#2。

表 387——例程控制肯定响应消息流-示例#2

消息方向	服务端 ——> 客户端		
消息类型	响应		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	停止例程的响应 SID	0x71	RCPR
#2	子功能=停止例程	0x02	SPR
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x01	RI_B2
#5	例程状态记录[例程状态#1]=车辆制造商定义	0x30	RRS_

13.2.5.3 示例#3：子功能=请求例程结果

该示例展示了如何在例程完成之后回读结果。当技术人员“扭动”受测试系统所有线束接头时，例程应该持续

间断性地（尽可能快）测试所有输入输出信号。通过例程 ID 0x0201 引用该例程。

测试环境：点火状态=on；发动机状态=off；车速=0[kph]。

客户端通过设置抑制肯定响应位（子功能参数 bit7）为“假”（‘0’）请求响应消息。

表 388 定义了请求例程结果的请求消息流-示例#3。

表 388——请求例程结果的请求消息流-示例#3

消息方向	客户端 ————— 服务端		
消息类型	请求		
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	例程控制的响应 SID	0x31	RC
#2	子功能=请求例程结果， 抑制肯定响应位=假	0x03	RRR
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x01	RI_B2

表 389——请求例程结果的肯定响应消息流-示例#3

消息方向	服务端 ————— 客户端		
消息类型	响应		
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	例程控制的响应 SID	0x71	RCPR
#2	子功能=请求例程结果	0x03	SPR
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x01	RI_B2
#5	例程状态记录[例程状态#1]=车辆制造商定义	0x30	RRS_
#6	例程状态记录[例程状态#2]=输入信号#1	0x33	RRS_
.	.	.	.
#n	例程状态记录[例程状态#m]=输入信号#m	0x8F	RRS_

13.2.5.4 示例#4：子功能=带例程控制选项的开启例程

该子条款规定了传输控制单元中开启例程的测试环境，在特定模式对特定档位进行档位校准。档位范围可以是#1-#20，模式可以是工作台，台架和整车。通过例程 ID 0x0202 引用该例程。

测试环境：点火状态=on；发动机状态=off；车速=0[kph]。

客户端通过设置抑制肯定响应位（子功能参数 bit7）为“假”（‘0’）请求响应消息。

表 390 定义了例程控制请求消息流-示例#4。

表 390——例程控制请求消息流-示例#4

消息方向	客户端 ————— 服务端		
消息类型	请求		
A_Data 类型	描述（所有值为十六进制）	字节值	助记符
#1	例程控制的响应 SID	0x31	RC
#2	子功能=开启例程， 抑制肯定响应位=假	0x01	RRR
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x02	RI_B2
#5	例程控制选项#1[选择档位]=车辆制造商定义	0x06	RCO_
#6	例程控制选项#2[测试环境]	0x01	RCO_

表 391 定义了例程控制肯定响应消息流-示例#4。

表 391——例程控制肯定响应消息流-示例#4

消息方向	服务端 ——> 客户端		
消息类型	响应		
A_Data 类型	描述 (所有值为十六进制)	字节值	助记符
#1	例程控制的响应 SID	0x71	RCPR
#2	子功能-开启例程	0x01	STR
#3	例程标识符[字节#1] (MSB)	0x02	RI_B1
#4	例程标识符[字节#2] (LSB)	0x02	RI_B2
#5	例程状态记录[例程状态#1]=车辆制造商定义	0x30	RRS_
#6	例程状态记录[例程状态#2]=响应时间	0x33	RRS_
.	.	.	.
#n	例程状态记录[例程状态#m]=输入信号#m	0x8F	RRS_

14 Upload Download functional unit

14 上传下载功能单元

14.1 Overview

14.1 综述

Table 392 defines the Upload Download functional unit.

表392定义了上传下载功能单元。

Table 392 — Upload Download functional unit

表 392 上传下载功能单元

服务	描述
请求下载	客户端请求从协商客户端到服务端的数据传输。
请求上传	客户端请求协商从服务器到客户端的数据传输。
传输数据	客户端向服务端发送数据（下载）或请求数据（上传）
请求传输终止	服务端请求终止数据传输。
请求文件传输	客户机请求在服务端和客户端之间协商文件传输。

14.2 请求下载 (0x34) 服务

14.2.1 服务描述

客户端使用“请求下载”服务启动从客户端到服务端的数据传输(下载)。

在服务端收到“请求下载”请求消息后，服务器应在发送肯定响应消息之前采取所有必要的操作来接收数据。

重要 —— 服务端和客户端应满足7.5中规定的请求和响应消息行为。

14.2.2 请求消息

14.2.2.1 请求消息定义

表393定义了请求消息。

表 393 — 请求消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	请求下载请求 SID	M	0x34	RD

#2	数据格式标识符	M	0x00 - 0xFF	DFI_
#3	地址和长度格式标识符	M	0x00 - 0xFF	ALFID
#4 : #(m-1)+4	内存地址[]=[字节#1 (MSB) : 字节#m]	M : C1	0x00 - 0xFF : 0x00 - 0xFF	MA_ B1 : Bm
#n-(k-1) : #n	内存大小[]=[字节#1 (MSB) : 字节#k]	M : C2	0x00 - 0xFF : 0x00 - 0xFF	MS_ B1 : Bk
C1:该参数的存在取决于地址长度信息参数的地址长度和长度。				
C2:该参数的存在取决于地址和长度标识符的内存大小信息。				

14.2.2.2 请求消息子功能参数 \$Level (LEV_) 定义

该服务不使用子功能参数。

14.2.2.3 请求消息数据参数定义

表394定义了请求消息的数据参数。

Table 394 — 请求消息数据参数定义

定义
数据格式标识符 该数据参数是一个字节值，每个半字节分别编码。高半字节指定“压缩方法”，低半字节指定“加密方法”。值 0x00 值指未压缩方法或加密方法。除 0x00 以外的值是特定于汽车制造商的。
地址和长度格式标识符 该参数是一个字节值，每个字节单独编码(示例值见附录 H.1)： -位 7 - 4: 长度(字节数)memorySize 参数； -位 3 - 0: 长度(字节数)memoryAddress 参数。
内存地址 该参数是要写入数据的服务端内存的起始地址。此地址使用的字节数由地址和长度格式标识符的低字节(位 3 - 0)定义。该参数中的字节#m 始终是在服务器中引用的地址中最低有效字节。地址中最高有效字节可以用作内存标识符。 一个使用内存标识符的例子是一个具有 16 位寻址和内存地址重叠的双处理器服务端(当一个给定地址对任何一个处理器都有效，但是会指向不同的物理内存设备或内部和外部的闪存)。在本例中，内存地址参数中另一个未使用的字节可以指定为用于选择所需内存设备的内存标识符。此功能的使用应由车辆制造商/系统供应商定义。
内存大小 服务端使用该参数比较内存大小与传输数据服务中传输的数据总量。这增加了编程的安全性。用于此大小的字节数由地址和长度格式标识符的高半字节(位 7 - 4)定义。如果使用数据压缩，则是汽车制造商指定内存大小是否表示压缩或未压缩的大小。

14.2.3 肯定响应消息

14.2.3.1 肯定响应消息定义

表395定义了肯定响应消息。

Table 395 — 肯定响应消息定义

A_Data 字节	参数名称	Cvt	字节值	助记符
#1	请求下载请求响应 SID	M	0x74	RDPR
#2	长度格式标识符	M	0x00 - 0xF0	LFID

#3 : #n	块长度的最大数量[]=[字节#1 (MSB) : 字节#m]	M : C1	0x00 - 0xFF : 0x00 - 0xFF	MNNROB_ B1 : Bm
---------------	--	--------------	---------------------------------	--------------------------

14.2.3.2 肯定响应消息数据参数定义

表396定义了肯定响应消息数据参数。

表396定义了肯定响应消息数据参数。

定义
长度格式标识符 该参数是一个字节值，每个半字节分别编码： -位 7 – 4: 块长度的最大数量参数的长度(字节数)； -位 3 – 0: 保留的文档，设置为 0x0。 该参数的格式与请求消息中所包含的地址长度格式标识符参数的格式兼容，但低半字节必须设置为 0x0。
块长度的最大数量 该参数由请求上传肯定响应消息使用，以通知客户端来自服务端的每个传输数据肯定响应消息中应该包含多少数据字节。这个长度反映了完整的消息长度，包括服务标识符和传输数据肯定响应消息中的数据参数。该参数允许客户端在服务端开始向客户端传输数据之前适应服务器的发送缓冲区大小。客户端需要接受与所报告的块长度的最大数量长度相等的传输数据响应。传输数据响应长度小于发送块长度的最大数量长度(如果有的话)是由服务端指定的。 注：给定块中的最后一个传输数据响应可能需要小于块长度的最大数量。

14.2.4 支持的否定响应码 (NRC_)

本服务应执行以下否定响应码。表397记录了每个响应代码出现的情况。如果错误场景适用于服务端，则应使用列出的否定响应。

表397 支持的否定响应码

否定响应码	描述	助记符
0x13	不正确的消息长度或无效格式 如果消息长度错误，则应发送此否定响应码	IMLOIF
0x22	条件不正确 如果服务端在接收软件或校准模块的下载过程中收到此服务的请求，则应返回此 NRC。如果在下载模块期间服务端和客户端之间的数据大小不匹配，则可能出现这种情况。则应返回此 NRC。	CNC
0x31	请求超出范围 该否定响应码应被返回，如果： ——指定的数据格式标识符无效。 ——指定的地址和长度格式标识符无效。 ——指定的内存地址/内存大小无效。	ROOR

0x33	安全访问拒绝 如果服务端是安全的(对于支持安全访问服务的服务端), 则在收到对该服务的请求时返回该 NRC。	SAD
0x70	上传下载不被接受 该 NRC 指出, 由于某些故障条件, 无法完成下载到服务端内存的尝试。	UDNA

6 会话层服务

6.1 文件概述

下图所示为基于不同协议的第6章的实施。

6.2 会话层服务

6.2.1 概述

服务接口定义了一系列有助于使用会话层所提供功能的必要服务, 如数据传输/接收和协议参数设置。

所有会话层服务均具有相同的一般结构。服务原语定义了服务用户(如诊断应用程序)与服务提供者(如会话层)的合作方式。为定义服务, 特规定了如下三类服务原语:

- 服务请求原语: **S_Data.request**, 上级应用层使用此原语将会话层所需控制信息或数据传递至会话层(即服务用户请求服务提供者处理控制信息或传输数据);
- 服务指示原语: **S_Data.indication**, 会话层使用此原语将状态信息和所接收的数据传递至上级应用层(即服务提供者将会话层内部事件或同级通信层实体服务用户的服务请求告知服务用户);
- 服务确认原语: **S_Data.confirm**, 会话层使用此原语将状态信息传递至应用层(即服务提供者将服务用户先前服务请求的结果告知服务用户)。

所有会话层服务均具有相同的一般格式。服务原语编写格式如下:

```
service_name.type(
    parameterA,
    parameter B,
    parameter C
    [,parameter X,...]
)
```

其中:

- “service_name (服务名称)”指服务名称(如 **S_Data**) ;
- “type (类型)”指服务原语的类型(如请求、指示、确认) ;
- “parameter A,... (参数 A...)”指服务原语以数值列表方式传递的 **S_PDU** (会话层协议数据单元)(如寻址信息、数据、长度、结果) ;
- “parameter A,parameter B,parameter C (参数 A, 参数 B, 参数 C)”指服务请求中须包含的强制参数;
- “[parameter X] (参数 X)”指特殊情况下可选用的参数。

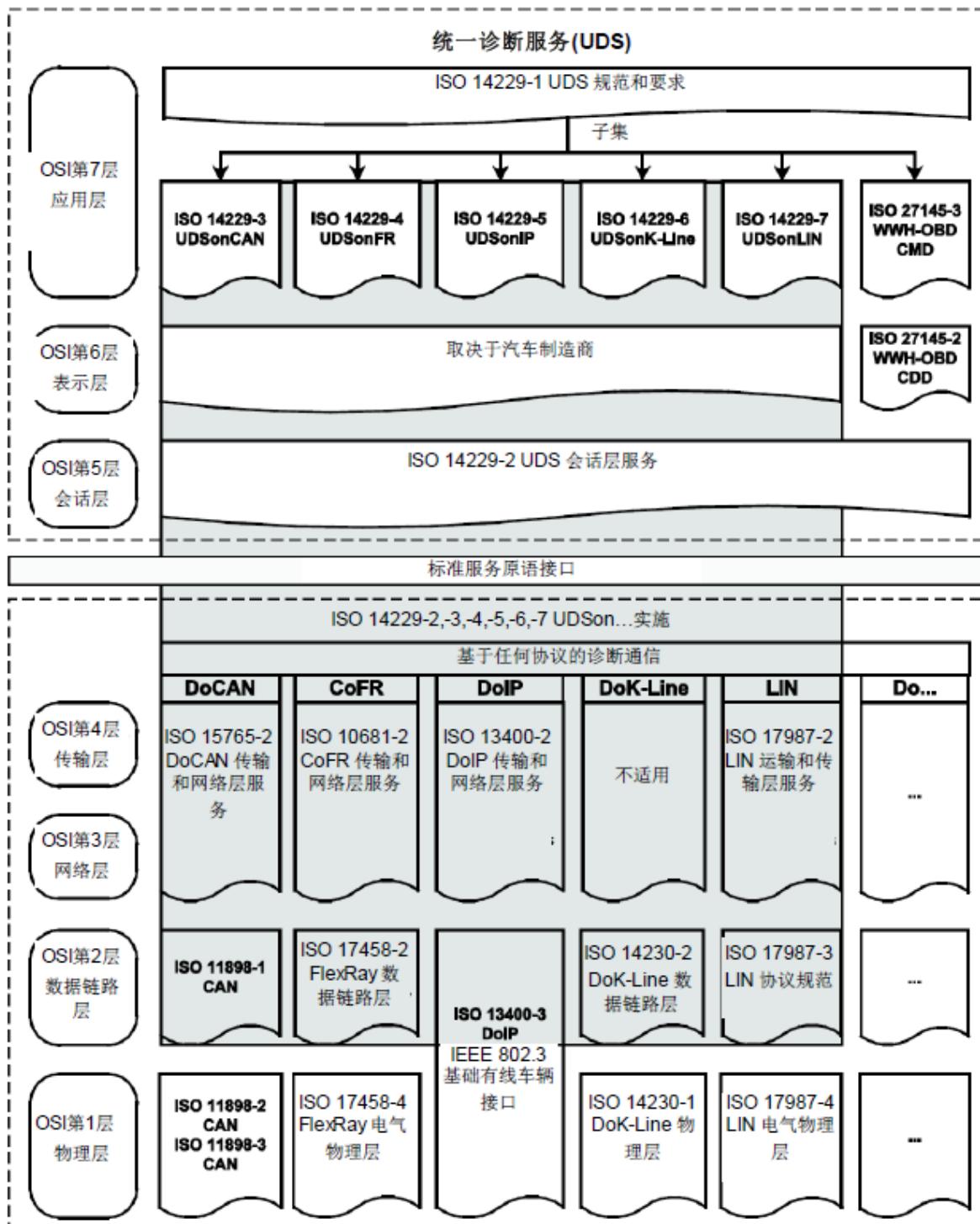


图2 依据 OSI 模型的 UDS 参考文件实施

下图为单帧消息的会话层服务原语。

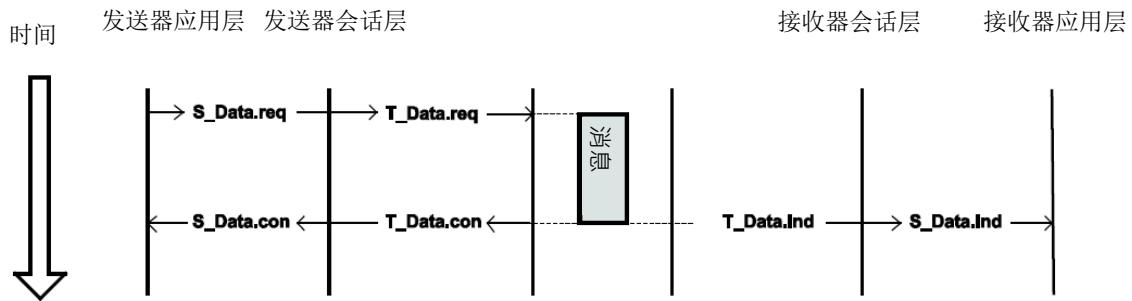
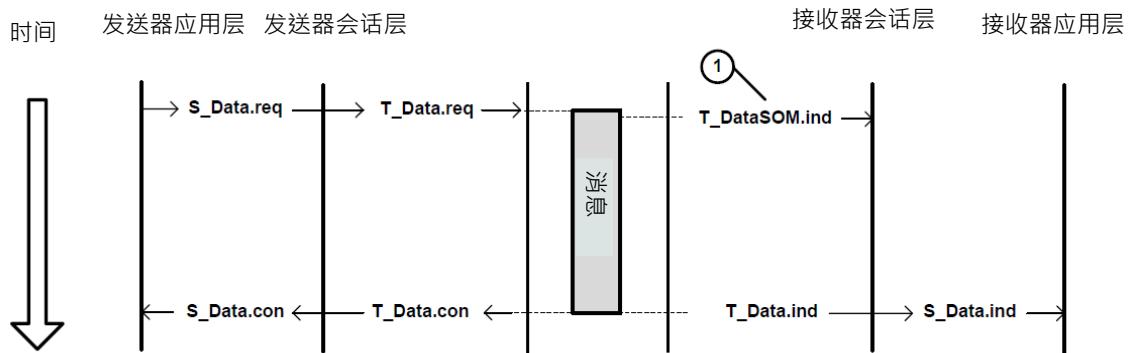


图3 会话层服务原语-单帧消息

下图为多帧消息的会话层服务原语（前提是传输/网络层支持T_DataSOM.ind接口）。



图例:

1 选用, 即传输/网络层支持T_DataSOM.ind接口

图4 会话层服务原语-多帧消息须对下列通信场景加以区分

对下列通信场景加以区分:

a) 物理通信:

- 1) 默认会话期间, 以及
- 2) 非默认会话期间 (要求进行会话处理) ;

b) 功能通信:

- 1) 默认会话期间, 以及
- 2) 非默认会话期间 (要求进行会话处理) 。

任何情况下均应考虑服务器通过否定响应消息 (包括否定响应代码0x78) 请求增强型响应计时时段的概率。使用不同ISO标准 (如ISO 15765-2 DoCAN或ISO 10681-2 CoFR) 中定义的运输/网络层服务执行客户端和服务器内诊断会话管理计时。

6.2.2 会话层服务原语规范

6.2.2.1 概述

为便于描述会话层功能，应考虑提供给上级各层的服务以及会话层内部操作。

6.2.2.2 S_Data.request

服务原语请求将字节数为S_Length的S_Data从发送器传送至S_SA、S_TA、S_TAtype和S_AE中地址消息标识的接收器同级实体。每当请求S_Data.request服务时，会话层应通过发出S_Data.confirm服务请求发送一个信号，告知服务用户消息传输完成（或失败）。

```

S_Data.request ( S_Mtype,
                  S_SA,
                  S_TA,
                  S_TAtype
                [S_AE],
                  S_Data[Data#1,Data#2,...,Data#n],
                  S_Length
)

```

6.2.2.3 S_Data.confirm

S_Data.confirm服务由会话层发出。该服务原语确认S_SA、S_TA、S_TAtype和S_AE中地址消息标识的S_Data.request服务已完成。参数S_Result提供服务请求的状态。

```
S_Data.confirm ( S_Mtype,  
S_SA,  
S_TA,  
S_TAtype,  
[S_AE],  
S_Result  
)
```

6.2.2.4 S_Data.indication

S_Data.indication服务由会话层发出。该服务原语指示S_Result事件，并将接受自经S_SA、S_TA、S_TAtype和S_AE中地址消息标识的同级协议实体且字节长度为S_Length的S_Data传输至上一级。

仅当S_Result等于S_OK时参数S_Data和S_Length有效。

```
S_Data.indication (  
    S_Mtype,  
    S_SA,  
    S_TA,  
    S_TAtype,  
    [S_AE],  
    S_Data[Data#1,Data#2,...,Data#n],  
    S_Length,  
    S_Result  
)
```

6.2.3 会话数据单元规范

6.2.3.1 S_Mtype (会话层消息类型)

类型：枚举。

范围：诊断、远程诊断。

说明：参数Mtype须用于确认服务请求内所含地址信息的内容和范围。本章规定的参数两个数值的范围。该参数旨在方便本文件用户可使用本文件规定的传输/网络层协议规定待使用的其他类型和组合的地址信息参数以扩展数值的范围。如果地址信息范围有更新，则应重新规定Mtype的数值，以便确认新的地址信息：

——若 S_Mtype=诊断，则地址信息应包括参数 S_SA、S_TA 和 S_TAtype。

——若 S_Mtype=远程诊断，则地址信息应包括参数 S_SA、S_TA、S_TAtype 和 S_AE。

6.2.3.2 S_SA (会话层源地址)

类型：2字节无符号整数值。

范围：0x0000–0xFFFF。

说明：参数S_SA用于对会话层协议发送实体进行编码。参数S_SA用于对客户端和服务器标识符进行编码。

6.2.3.3 S_TA (会话层目标地址)

类型：2字节无符号整数值。

范围：0x0000–0xFFFF。

说明：参数S_TA用于对会话层协议接收实体进行编码。参数S_TA用于对客户端和服务器标识符进行编码。

6.2.3.4 S_TAtype (会话层目标地址类型)

类型：枚举。

范围：物理性、功能性。

说明：参数S_TAtype是参数S_TA的一个配置属性。须使用该参数对通信层同级通信实体所使用的通信模型进行编码。共规定了两种通信模型：1至1通信，称为物理定址；1至n通信，称为功能定址。

——各类会话层消息均应支持物理定址（1至1通信）。

——应支持功能性寻址（1至n通信）。传输/网络层要求可能会限制功能定址的应用（如 CAN 数据链路层上的 SingleFrame（单帧））。

6.2.3.5 S_AE (会话层地址扩展, 选用参数)

类型：2字节无符号整数值

范围：0x0000–0xFFFF

说明：S_AE参数用于扩展大型网络的扩展地址范围，并对子网（正在通信的局域网络除外）运输/网络层发送和接收实体进行编码。如果将Mtype设置为远程诊断，则S_AE仅为寻址信息的一部分。

6.2.3.6 S_Length

类型：4字节。

范围：0x0000 0000–0xFFFF FFFF。

说明：本参数包括待传输/接收的数据的长度。

6.2.3.7 S_Data

类型：字节串。

范围：不适用。

说明：本参数包括高级层次实体待交换的所有数据。

6.2.3.8 S_Result

类型：枚举。

范围：S_OK、S_NOK。

说明：本参数包括与服务执行结果相关的状态。

6.2.3.9 消息传输中 S_PDU 与 T_PDU 之间相互映射

定义用于请求传输诊断服务请求/响应的会话层协议数据单元参数按以下方式映射至用于传输客户端/服务器内消息的会话层协议数据单元的参数。

定义用于接收消息的运输/网络层协议数据单元参数按以下方式映射至用于指示诊断响应/请求接收情况的会话层协议数据单元的参数。

传输/网络层确认表明消息传输成功(T_Data.con)。该确认将向前传送至应用程序以开始执行那些须在传输了请求/反应信息后予以立即执行的操作（如ECU重启、波特率变化等）。

消息起始T_PDU(T_DataSOM.ind)接收的传输/网络层指示仅用于在会话层内执行会话层计时（见6.3），因此不会传送至应用层。因此，未定义T_DataSOM.ind至T_PDU的映射。

表2定义了会话层S_PDU与传输/网络层T_PDU之间的相互映射。

表1 会话层 S_PDU 与传输/网络层 T_PDU 间相互映射

参数S_PDU(会话层协议数据单元)	说明	参数T_PDU(网络/传输层协议数据单元)	说明
S_Mtype	会话层消息类型	T_Mtype	传输/网络层数据类型
S_SA	会话层源地址	T_SA	传输/网络层源地址
S_TA	会话层目标地址	T_TA	传输/网络层目标地址
S_TAtype	会话层目标地址类型	T_TAtype	传输/网络层目标地址类型
S_AE ^a	会话层地址扩展	T_AE ^a	传输/网络层地址扩展
S_Data[1]-S_Data[n]	会话层数据	T_Data[1]-T_Data[n]	传输/网络层应用数据
S_Length	会话层数据长度	T_Length	传输/网络层数据长度
S_Result	会话层结果	T_Result	传输/网络层结果

a 若 Mtype=诊断，则地址信息应包括参数 SA、TA 和 TAtype。

若 Mtype=远程诊断，则地址信息应包括参数 SA、TA、TAtype 和 AE。

6.3 计时参数定义

6.3.1 一般应用程序时间参数要求

6.3.1.1 服务器

一个服务器使用一个单独的应用程序计时器(P2_{Server})实施，并由T_Data服务原语接口(T_Data.ind, T_Data.con, T_Data.req)触发（启动和停止）。

P2Server应用程序计时器内加载了一个P2Server_max/P2*Server_max参数值。上述两个参数及其数值见6.3.2、6.6.3表的参数要求。

计时参数P4Server指接收请求(T_Data.indication)至开始传输最终响应(T_Data.request)的时间。最终响应为肯定响应或否定响应，但不包括否定响应代码0x78“requestCorrectlyReceived-ResponsePending（请求正确接收响应待定）”。请求制定定期响应计划时，初始USDT肯定或否定响应（指示接收或未接收制定定期响应计划的请求）视为最终响应。P4Server为性能需求。P4Server_max为P4Server的最大数值。如果P4Server_max与P2Server_max相等，则表明带否定响应代码0x78的否定响应对该服务或数据不适用。

这些要求仅适用于由服务器/ECU支持的服务。不受支持的服务使用一个等于P2Server_max的P4Server_max（即NRC 0x78不适用）。

6.3.1.2 客户端

一个客户端使用一个单独的应用程序计时器(PClient) 实施，并由 T_Data 服务原语接口(T_Data.con、T_DataSOM.ind、T_Data.ind) 触发(启动和停止)。

对于理论上支持 T_DataSOM.ind 服务原语(如DoCAN) 的所有协议，P_{Client} 应用程序计时器始终加载了一个 P2_{Client_max}/P2_{*Client_max} 参数。对于理论上支持 T_Data.ind 服务原语(如DoIP) 的所有协议，P_{Client} 应用程序计时器始终加载了一个 P6_{Client_max}/P6_{*Client_max} 参数。当客户端应用层接收了一个 T_Data.con 服务原语时，P_{Client} 应用程序计时器启动。根据协议类型(有或无 T_DataSOM.ind)，该计时器加载了一个 P2_{Client_max} 或 P6_{Client_max} 参数值。根据协议类型(有或无 T_DataSOM.ind)，P_{Client} 应用程序计时器停止，无论应用程序接收了服务原语是 T_DataSOM.ind 或是 T_Data.ind。

对于支持 T_DataSOM.的协议，客户端应用程序通过比较 P_{Client} 应用程序计时器实际值与 P2_{Client_max} 参数值验证应用程序计时是否正确。如果 P_{Client} 小于或等于 P2_{Client_max} 时接收到 T_DataSOM.ind 或 T_Data.ind，则表明计时满足本标准规定的要求。但如果 P_{Client} 小于或等于 P2_{Client_max} 时接收到 no.ind，则将检测到一个错误状态。此错误状态应标记于应用层，其参数编入 T_DataSOM.ind 或 T_Data.ind 服务原语。

对于仅支持 T_Data.ind 的协议，客户端应用程序通过比较 P_{Client} 应用程序计时器实际值与 P6_{Client_max} 参数值验证应用程序计时是否正确。如果 P_{Client} 小于或等于 P6_{Client_max} 时接收到 T_Data.ind，则表明计时满足本章的要求。但如果 P_{Client} 小于或等于 P6_{Client_max} 时接收到 no.ind，则将检测到一个错误状态。此错误状态应标记于应用层，其参数编入 T_Data.ind 服务原语。

上述两个参数及其数值见(见表3定义和表4参数值)。

6.3.2 应用程序时间参数定义-默认会话

通电后服务器启动 defaultSession (默认会话)。若未启动其他诊断会话，则服务器断电前 defaultSession (默认会话) 保持运行。defaultSession (默认会话) 的计时参数定义应符合表3要求，而参数值定义则应符合表4要求。

表2 defaultSession (默认会话) 消息计时参数定义

计时参数	说明	类型
ΔP2	参数 ΔP2 指车辆网络设计相关消息的最严重传输延迟，包括因网关和总线负载相关仲裁引起的延迟。ΔP2 数值包括向编址服务器/ECU (ΔP2_request) 传输请求的时间；若协议支持 T_DataSOM.ind，还包括直至开始传输 T_DataSOM.或 T_Data.ind 标识的响应(若响应为单帧消息，如 ISO 15765 DoCAN) 的时间。	性能需求
ΔP6	参数 ΔP6 指车辆网络设计相关消息的最严重传输延迟，包括因网关和总线负载相关仲裁引起的延迟。ΔP6 数值包括向编址服务器/ECU (ΔP6_request) 发送请求的时间以及向客户端/测试仪 (ΔP6_resPonse) 传送完整响应的时间。协议是否支持 T_DataSOM.ind (如 ISO 15765 DoCAN) 或不支持 T_DataSOM.ind (如 ISO 13400 DoIP) 均与 ΔP6 无关。	性能需求
P2Server	服务器接收请求消息(由 T_Data.ind 标识)后触发响应消息的性能需求。	性能需求
P2Client	成功传输请求消息(由 T_Data.con 标识)后客户端等待开始接收响应消息(由多帧消息的 T_DataSOM.ind 或单帧消息的 T_Data.ind 标识)的超时。	计时器重新加载值
P6Client	成功传输请求消息(由 T_Data.con 标识)后客户端等待完成接收对应响应消息(由 T_Data.ind 标识，如 ISO 13400 DoIP)的超时。	计时器重新加载值
P2*Server	传输带否定响应代码 0x78 (增强型响应计时) 的否定响应消息(由 T_Data.con 标识)后服务器触发响应信息的性能需求。	性能需求
P2*Client	接收带否定响应代码 0x78 (增强型响应计时) 的否定响应消息(由 T_Data.con 标识)后客户端等待开始接收响应消息(由多帧消息的 T_DataSOM.ind 或单帧消息的 T_Data.ind 标识)的超时。	计时器重新加载值
P6*Client	接收带否定响应代码 0x78 (增强型响应计时) 的否定响应消息(由 T_Data.con 标识)后完成接收对应响应消息(由 T_Data.ind 标识，如 ISO13400 DoIP)的超时。	计时器重新加载值
P3Client_Phys	成功发送一个无响应请求的物理编址请求消息(由 T_Data.con 标识)后客户端	计时器重新加载值

	等待传输下一个物理编址请求消息（见图 19）的最短时间。	
P3Client_Func	成功发送一个无响应请求的物理编址请求（由 T_Data.con 标识）后客户端等待传输下一个物理编址请求消息（见图 19）的最短时间（如未要求进行响应或请求数据仅由物理编址服务器（见 8.3）子集支持）。	计时器重新加载值
P4Server	服务器端接收请求(T_Data.indication)至开始传输最终响应(T_Data.request)之间的时间。	性能需求

各服务器/ECU应能在成功传输先前请求消息的响应消息(T_Data.con)后立即处理新请求消息。但汽车制造商在某些特殊使用情况下可忽视此要求，此时服务器/ECU在执行前次服务请求后请求延时，如ECUReset (ECU重置) 服务。

表3 defaultSession（默认会话）消息计时参数值定义

计时参数	最小值	最大值
ΔP2	0ms	汽车制造商指定数值 ΔP2request+ΔP2resPonse
ΔP6	0ms	汽车制造商指定数值 ΔP6request+ΔP6resPonse
P2Server	0ms	服务器指定数值 推荐数值： 50ms
P2Client	P2Server_max+ΔP2max	---a)
P6Client	P2Server_max+ΔP6max	---a)
P2*Server	0ms ^{b)}	服务器指定数值 推荐数值： 5000ms
P2*Client	P2*Server_max+ΔP2resPonse	---c)
P6*Client	P2*Server_max+ΔP6resPonse	---c)
P3Client_Phys	P2Server_max+ΔP2max	---d)
P3Client_Func	P2Server_max+ΔP2max	---d)
P4Server	P2Server	见 OBD 诊断使用案例相关立法定义。增强型诊断使用案例的汽车制造商指定数值。

a 客户端等待响应消息的最短时间取决于客户端，前提是 P2Client/P6Client 大于 P2Client/P6Client 最大规定值。
b 在增强型响应计时阶段，接续传输否定消息（均带否定响应代码 0x78）的最小间隔时间应为 0.3*P2*Server_max，以防带否定响应代码 0x78 消息的数据链路溢满。
c 客户端所用 P2*Client/P6*Client 的最小数值取决于客户端，前提是该数值大于 P2*Client/P6*Client 的最小规定值。
d 客户端等待传输下一请求消息的最短时间取决于客户端，前提是服务器内非默认会话的 S3Server 计时保持激活。

参数ΔP2/ΔP6考虑了任何系统网络设计相关的延迟，包括因网关和总线带宽引起的延迟与安全余裕之和（如最差50%）。最坏场景（完成客户端至服务器并由服务器返回客户端一个“来回”所需时间）基于系统设计，受以下因素影响：

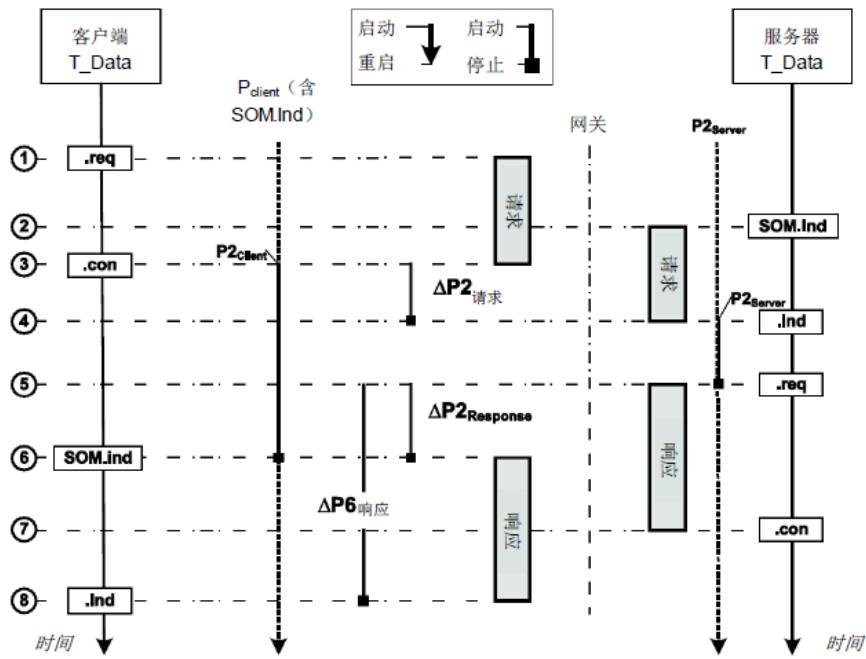
- 所涉及网关的数量；
- 帧传输时间（波特率）；
- 总线利用率；以及
- 传输/网络层装置驱动实施方法（轮询 vs. 中断）和处理时间。

ΔP6数值包括向编址服务器发送请求的时间以及向客户端传送响应的时间。

$$\Delta P2 = \Delta P2_{request} + \Delta P2_{resPonse}$$

$$\Delta P6 = \Delta P6_{request} + \Delta P6_{resPonse}$$

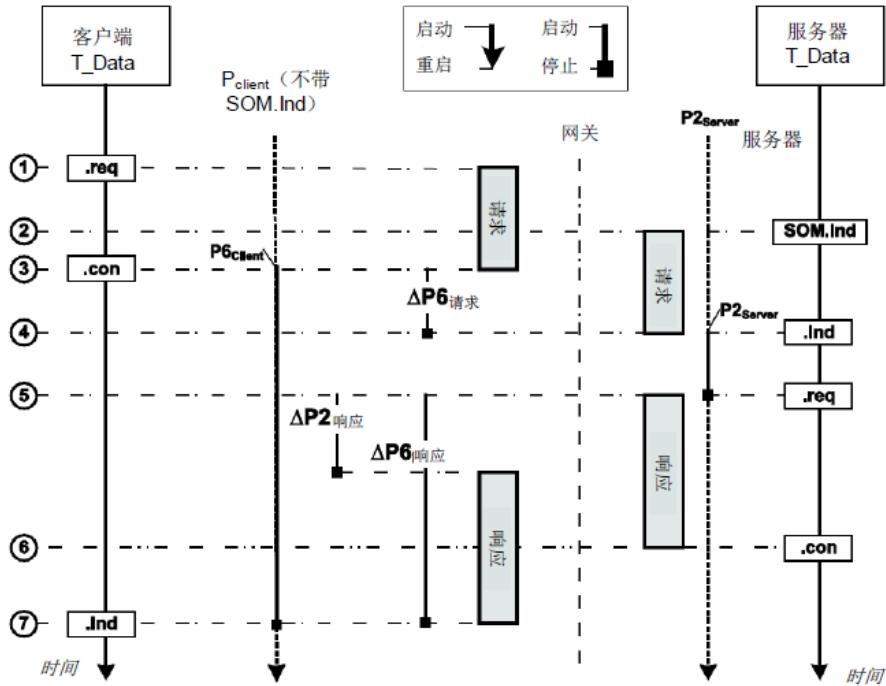
图4和图5为ΔP2与ΔP6组成示例。



图例：

- 1 客户端T_Data.req: 诊断应用程序向传输/网络层发出一个请求消息。
- 2 服务器T_DataSOM.ind: 传输/网络层向诊断应用程序发送请求消息的StartOfMessage（消息起始）。
- 3 客户端T_Data.con: 传输/网络层向诊断应用程序发出请求消息已完成的确认消息。客户端通过默认重载值P2Client=P2Client_max启动PClient计时器。PClient计时器数值设定时应考虑任何因车辆网络设计（网关通信、总线带宽等）引起的延迟。
- 4 服务器T_Data.ind: 传输/网络层向诊断应用程序发出请求消息完成信息。服务器利用P2Server= P2Server_maxP2启动PServer计时器。
- 5 服务器T_Data.req: 诊断应用程序已准备响应消息，并向P2Server内的传输/网络层发出T_Data.req。服务器停止P2Server计时器。
- 6 客户端T_DataSOM.ind: 传输/网络层向诊断应用程序发送响应消息的StartOfMessage（消息起始）。客户端停止PClient计时器。
- 7 服务器T_Data.con: 传输/网络层向诊断应用程序发出响应消息完成信息。
- 8 客户端T_Data.ind: 传输/网络层向诊断应用程序发出T_Data.ind，确认完成响应消息。

图5 ΔP2 和ΔP6 示例-响应消息（带 SOM.ind）



图例：

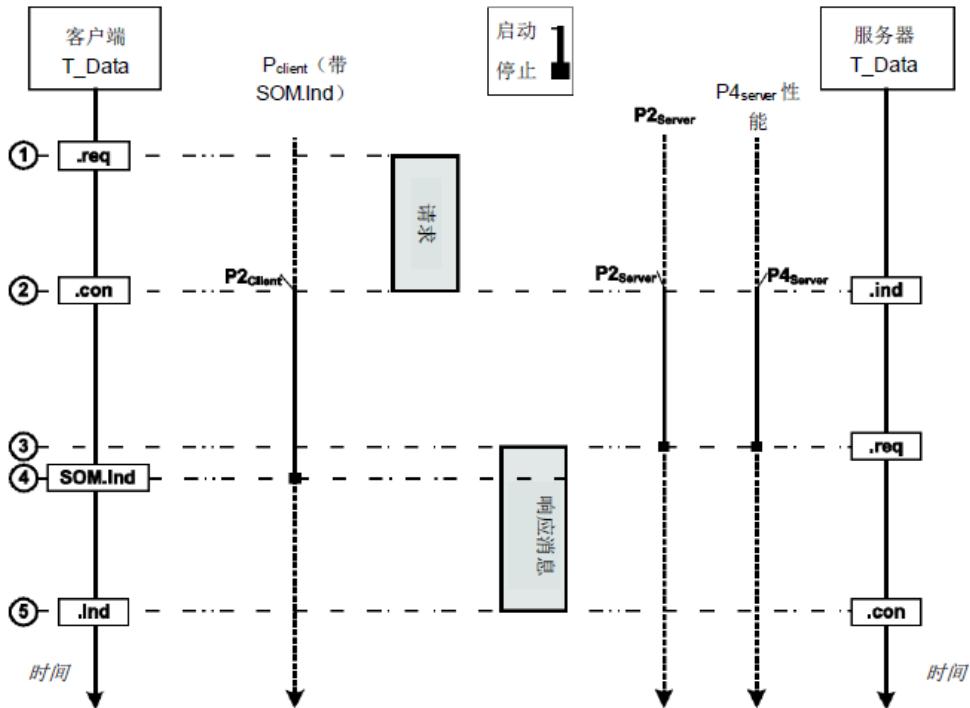
- 1 客户端T_Data.req: 诊断应用程序向传输/网络层发出一个请求消息。
- 2 服务器T_DataSOM.ind: 传输/网络层向诊断应用程序发送请求消息的StartOfMessage（消息起始）。
- 3 客户端T_Data.con: 传输/网络层向诊断应用程序发出请求消息已完成的确认消息。客户端利用默认重新加载值P6Client=P6Client_max启动PClient计时器。PClient计时器数值设定时应考虑任何因车辆网络设计（网关上通信、总线带宽等）引起的延迟。
- 4 服务器T_Data.ind: 传输/网络层向诊断应用程序发出请求消息完成信息。服务器通过P2Server=P2Server_max的默认值启动P2Server计时器。
- 5 服务器T_Data.req: 诊断应用程序已编写好响应消息，并在P2Server内的向传输/网络层发出T_Data.req。服务器停止P2Server计时器。
- 6 服务器T_Data.con: 传输/网络层向诊断应用程序发出响应消息完成信息。
- 7 客户端T_Data.ind: 传输/网络层向诊断应用程序发出T_Data.ind，确认完成响应消息。客户端停止PClient计时器。

图6 ΔP2 和ΔP6 示例—响应消息（不带 SOM.ind）

注：为简化计时参数的描述，在各图中均假设客户端和服务器位于同一网络。所有说明和示意图均按时间顺序呈现。

6.3.3 不带强化响应计时的 P4Server 示例

图6所示为P4_{Server}=P2_{Server}示例。此情况下，服务器响应性能计时参数表明不允许存在包含NRC 0x78的否定响应。



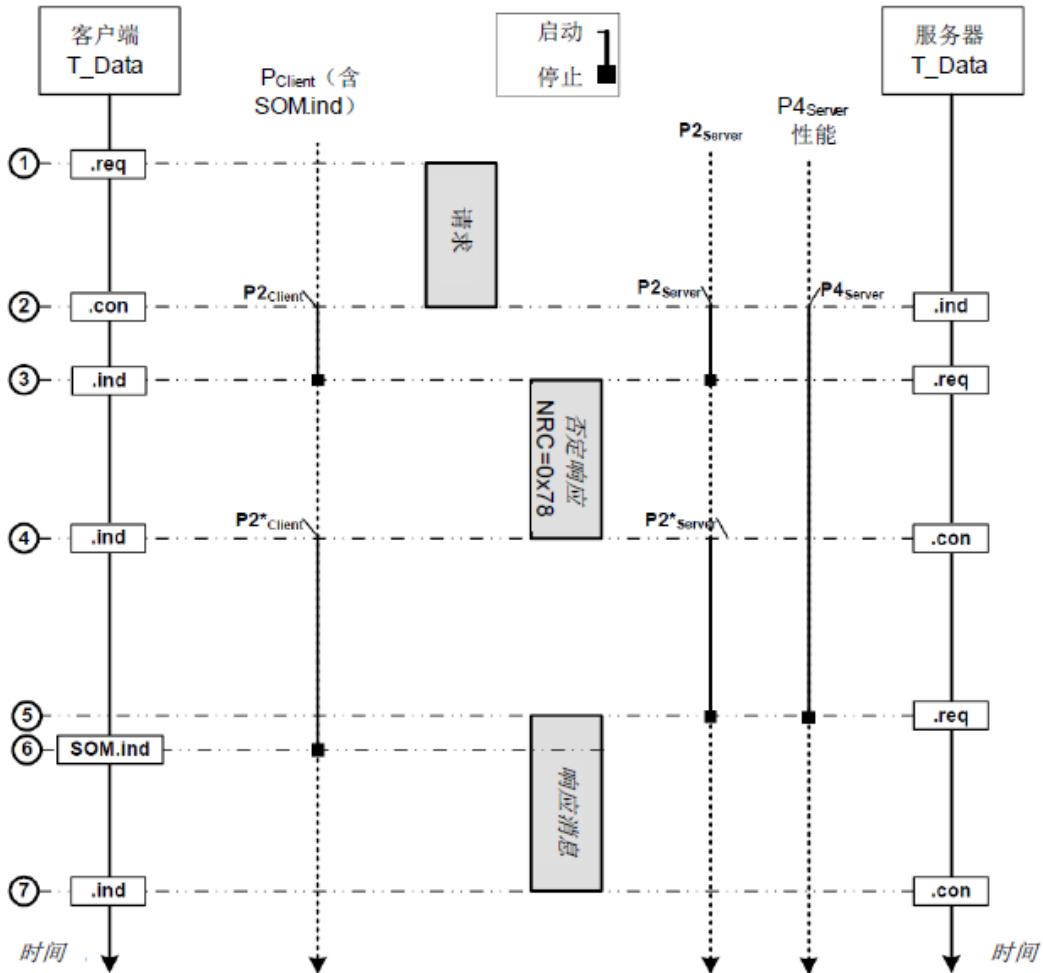
图例：

- 1 客户端T_Data.req: 诊断应用程序向传输/网络层发送请求消息。
- 2 服务器T_Data.ind: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器通过P2Server=P2Server_max默认值启动P2Server计时器，并通过P4Server=P4Server_max默认值启动P4Server计时器。
如果P4Server=P2Server适用于特定T_Data.ind，则服务器应确保在P2Server计时器停止前启动最终肯定或否定响应（即允许存在含NRC 0x78的否定响应）。
- 3 客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息完成确认信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。PClient计时器的值应考虑到基于车辆网络设计（网关通信、总线带宽等）所涉及到的潜在因素。
- 4 服务器T_Data.SOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage（消息起始）接收信息。客户端停止PClient计时器。
- 5 服务器T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。
- 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息完成信息。

图7 不带强化响应计时的 P4Server 示例

6.3.4 含强化响应计时的 P4Server 示例

图7为P4_{Server} > P2_{Server}示例。此情况下，服务器响应性能计时参数表明只要不超过P4_{Server}就允许存在包含NRC 0x78的否定响应。



图例：

- 1 客户端T_Data.req: 诊断应用程序向传输/网络层发送请求消息。
- 2 服务器T_Data.ind: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器通过P2Server=P2Server_max默认值启动P2Server计时器，并通过P4Server=P4Server_max默认值启动P4Server计时器。
如果P4Server>P2Server适用于特定T_Data.ind，则服务器应确保在P2Server计时器停止前启动最终肯定或否定响应（即，只要不超过P4Server，就允许存在含NRC=0x78的否定响应）。
- 3 客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息完成确认信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。PClient计时器的值应考虑到基于车辆网络设计（网关通信、总线带宽等）所涉及到的潜在因素。
- 4 服务器T_Data.req: 诊断应用程序未包含编写好的肯定响应消息，并通过T_Data.req向P2Server内的传输 / 网络层发送含NRC=0x78的否定响应消息。服务器停止P2Server计时器。
在此情况下，服务器仍不能提供强化P2*Server内所需信息，此时，服务器可发送含否定响应代码0x78的单条否定响应消息。这将导致客户端通过P2Client强化重载值重启其PClient计时器。为简单起见，图中仅显示含否定响应代码0x78的单条否定响应消息。
- 5 客户端T_Data.ind: 传输/网络层向诊断应用程序发送响应消息接收信息。客户端停止PClient计时器。
- 6 服务器T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器通过P2*Server=P2*Server_max值（默认强化计时）启动P2Server计时器。
- 7 客户端T_Data.ind: 传输/网络层向诊断应用程序发送响应消息完成信息。

图8 含强化响应计时的 P4Server 示例

6. 3. 5 非默认会话相关会话时间参数定义

开始诊断会话（非默认会话）时，要求通过表5给出的会话层计时参数实现会话。

表4 非默认会话相关会话计时参数定义

计时参数	说明	类型	建议的重载时间 ms	超时 ms
------	----	----	---------------	----------

S3Client	为保持多个服务器内诊断会话（非默认会话）处于激活状态（功能通信），由客户端传输的功能定址 TesterPresent（测试仪保活）(0x3E) 请求消息的时间间隔或向单个服务器传输请求消息（物理通信）的最大时间间隔。S3Client 超时值包括网络（网关延迟等）消息行程时间。	计时器重载值	2000	<S3Server
<S3Server	未接收到任何诊断请求消息时，为使诊断会话（非默认会话）保持在激活状态，服务器所需时间。S3Server 公差为 -0ms, +200ms。	计时器重载值	不适用	5000

表3和表4内规定的计时参数定义及值也适用于非默认会话。此外，过渡至非默认会话时，服务器可更改其应用程序层计时P2_{Server}和P2*_{Server}，以便实现特定性能或弥补非默认诊断会话期间所受限制。DiagnosticSessionControl（诊断会话控制）肯定响应消息报告非默认诊断会话适用计时参数，此情况下，要求传输响应或在不要求传输响应的情况下，客户端需事先了解此消息。当客户端功能性开始非默认会话时，客户端应适应于响应服务器计时参数。

表5规定了客户端和服务器启动/重启S3Client/S3Server计时器的条件。对于客户端，周期性传输的功能定址 TesterPresent（测试仪保活）(0x3E) 请求消息应区别于按顺序传输的物理定址TesterPresent（测试仪保活）(0x3E) 请求消息，后者仅在其他诊断请求消息不存在的情况下传输。对于服务器，无需区别两类TesterPresent（测试仪保活）(0x3E) 请求消息。此外，表5所示为基于传输/网络层服务原语的S3Server计时器处理，这就意味着S3Server计时器也应在接收服务器不支持的诊断请求消息时重启。非默认会话期间，表6中给出的附加计时器资源要求应适用于客户端和服务器。

表5 客户端和服务器会话层计时开始/停止条件

计时参数	动作	物理和功能通信（通过功能定址和周期性传输 TesterPresent（测试仪保活）请求消息）	仅物理通信（通过物理定址和按顺序传输的 TesterPresent（测试仪保活）请求消息）
S3Client	初始启动	T_Data.con表明DiagnosticSessionControl（诊断会话控制）(0x10)请求消息的完成。仅在会话类型为非默认会话时为真。	T_Data.con 表明在不要求响应的情况下 DiagnosticSessionControl（诊断会话控制）(0x10)请求消息的完成。 T_Data.ind 表明在不要求响应的情况下 DiagnosticSessionControl（诊断会话控制）(0x10)请求消息的接收。
	后续启动	T_Data.con表明功能定址 TesterPresent（测试仪保活）(0x3E) 请求消息的完成，此消息在每次S3Client计时器计时超时时传输。	T_Data.con表明在不要求响应的情况下请求消息的完成。 T_Data.con表明单帧或多帧请求消息传输过程中出现错误。 T_Data.ind表明在不要求响应的情况下请求消息的接收。 T_Data.ind表明多帧响应消息接收过程中出现错误。
<S3Server	初始启动	T_Data.con表明在要求响应消息的情况下，从默认会话过渡到非默认会话所需 DiagnosticSessionControl（诊断会话控制）肯定响应消息传输的完成。	
		在不要求/允许响应消息的情况下，从默认会话过渡到非默认会话所需服务 DiagnosticSessionControl（诊断会话控制）(0x10)所要求的动作成功结束。	
	后续停止	T_DataSOM.ind表明多帧请求消息的开始或T_Data.ind显示单帧请求消息的接收。如果 defaultSession（默认会话）激活，则停用S3Server计时器。	
	后续启动	T_Data.con表明在要求/允许传输响应消息的情况下，汇总服务执行情况的任何响应消息（最终响应消息）完成。含否定响应代码0x78的否定响应无法启动S3Server计时器。 在未要求/允许响应消息（肯定和否定）的情况下要求动作完成。 T_Data.ind表明多帧请求消息接收过程中出现错误。 关于要求服务器传输未经请求响应消息（如周期数据或基于事件的响应）时服务器S3Server处理相关信息，见本标准数据链具体实施文件。	

6.3.6 客户端和服务器计时器的资源要求

默认会话和任何非默认会话期间，满足上述计时要求的客户端和服务器所需计时器资源应符合表7和表8规定。

表7规定了默认会话和非默认会话期间计时器资源要求。

表 7--默认会话期间计时器资源要求

计时参数	客户端	服务器
PClient	各逻辑通信通道（物理和功能通信）所需单个计时器，如各点对点通信要求提供独立的通信通道。	不适用
P2Server	不适用	为确保在 P2*Server 失效前传输含否定响应代码 0x78 的后续否定响应消息，要求提供单个计时器进行强化响应计时。
P3Client_Phys	要求为各逻辑物理通信通道提供单个计时器。	不适用
P3Client_Func	要求为各逻辑功能通信通道提供单个计时器。	不适用

表8规定了非默认会话期间附加计时器资源要求。

表 8 默默认会话期间附加计时器资源要求

计时参数	客户端	服务器
S3Client	当通过周期性传输、功能定址 TesterPresent (测试仪保活) (0x3E) 请求消息保持非默认会话期间服务器时，要求提供单个服务器。已经激活的诊断会话无需提供附加计时器。	不适用
	当按顺序传输物理定址 TesterPresent (测试仪保活) (0x3E) 请求消息时，各点对点通信通道要求提供单个计时器，以在无其他诊断请求消息的情况下保持非默认会话中单个服务器。	
<S3Server	不适用	服务器中要求提供单个计时器，因为仅在某个时间单个服务器内的单个诊断会话处于激活状态。

6.3.7 错误处理

物理和功能通信期间，由客户端和服务器执行的应用程序层和会话管理过程中错误处理应符合表9和表10要求，其中假定客户端和服务器依据本章执行引用程序和会话层计时。

表 9 一般客户端错误处理建议

通信阶段	客户端错误类型	客户端处理	
		物理通信	功能通信
请求传输	源于含否定结果值的传输/网络层的 T_Data.con。	客户端应在错误提示时间 P3Client_Phys 出现后重复最后一个请求。在物理定址和按顺序传输的 TesterPresent (测试仪保活) 的情况下，重启 S3Client (因为 S3Client 已根据请求消息传输停止)。	客户端应在错误提示时间 P3Client_Func 出现后重复最后一个请求。
PClient P*Client	超时	客户端应重复最后一个请求。在物理定址和按顺序传输 TesterPresent (测试仪保活) 的情况下，重启 S3Client (因为 S3Client 已根据请求消息传输停止)。	若客户端无法了解响应服务器数量，则表明客户端预计无更多响应消息。不要求重新显示请求消息。客户端应完全接收传输中的所有响应消息直至客户端可继续发出更多请求。 若客户端可了解响应服务器数量，则表明不是所有预计服务器均会响应客户端。客户端应在完全接收到超时发出的响应消息后重复请求。
响应接收	源于含否定结果值的传输/网络层的 T_Data.ind。	客户端应重复最后一个请求。 在物理定址和按顺序传输的 TesterPresent (测试仪保活) 的情况下，重启 S3Client (因为 S3Client 已根据请求消息传输停止)。	客户端应在完全接收到提示错误的响应消息后重复请求。

定义的客户端错误处理次数最多为两次，这就意味着服务请求传输出现的最糟糕情况为三次。

表 10 规定了服务器错误处理。

表 10--服务器错误处理

通信阶段	服务器错误类型	错误处理
请求接收	源于含否定结果值的传输/网络层的 T_Data.ind。	重启 S3Server 计时器 (因为此计时器已根据之前接收到的 StartOfMessage (消息起始) 指示停止)。服务器应忽略请求。

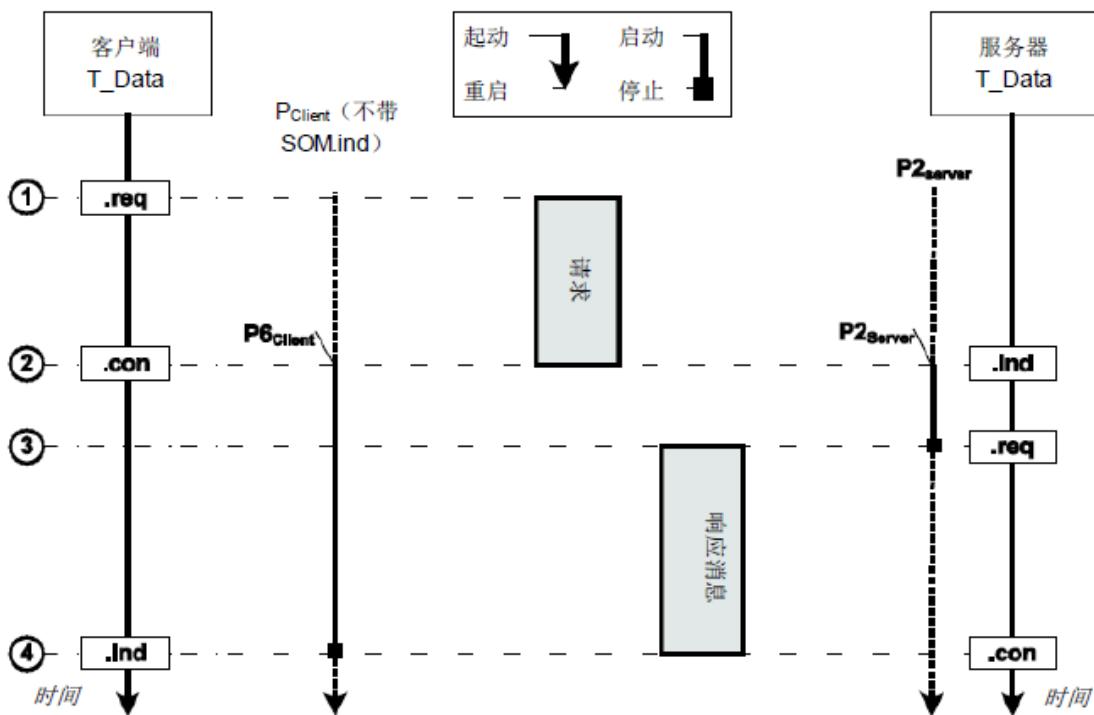
响应传输	源于含否定结果值的传输/网络层的 T_Data.con。	重启 S3Server 计时器（因为此计时器已根据之前接收到的请求消息指示停止）。服务器不应再次传输响应消息。
------	------------------------------	---

6.4 通信期间计时处理

6.4.1 物理通信

6.4.1.1 DefaultSession（默认会话）期间的物理通信--不带 SOM.ind

图8以图解形式说明了默认会话期间用于物理定址请求消息(不带SOM.ind)的客户端和服务器内的计时处理。



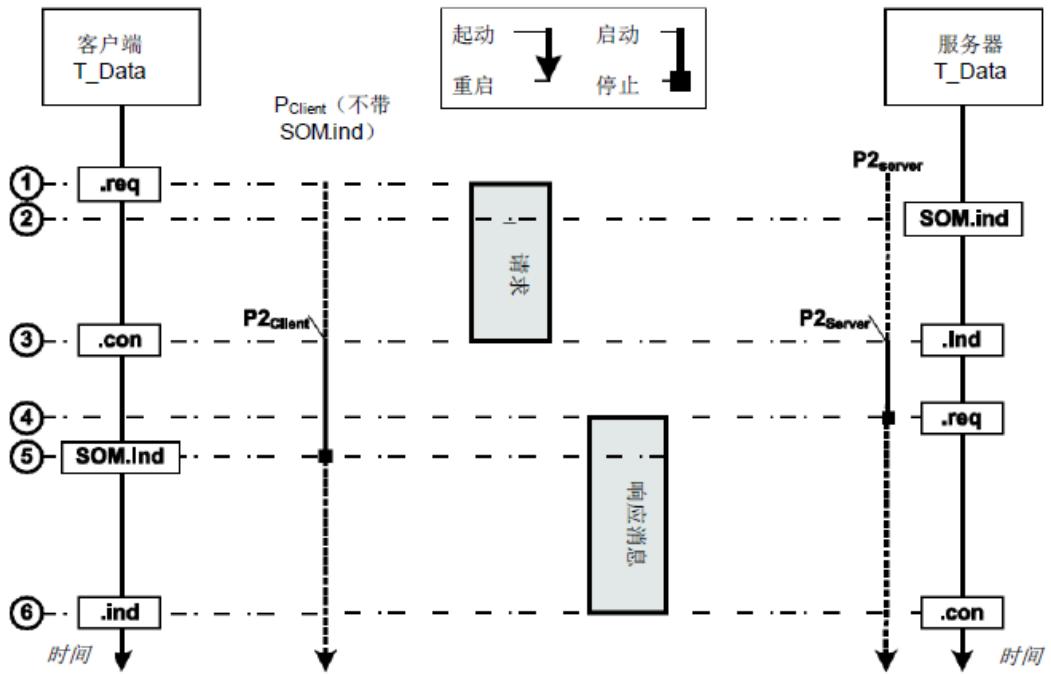
图例：

- 1 客户端 T_Data.req: 诊断应用程序向传输/网络层发送请求消息。
- 2 服务器 T_Data.ind: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器通过 P2Server=P2Server_max 默认值启动 P2Server 计时器。
- 3 客户端 T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息完成确认信息。客户端通过 P6Client=P6Client_max 默认重载值启动其 PClient 计时器。PClient 计时器的值应考虑到基于车辆网络设计（网关通信、总线带宽等）所涉及到的潜在因素。
- 3 服务器 T_Data.req: 诊断应用程序已编写好响应消息，并向 P2Server 内的传输/网络层发送一条 T_Data.req。服务器停止 P2Server 计时器。
- 4 服务器 T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。
- 客户端 T_Data.ind: 传输 / 网络层向诊断应用程序发送请求信息完成信息。客户端停止 PClient 计时器。

图9 默认会话期间的功能通信--不带 SOM.ind

6.4.1.2 默认会话期间的物理通信--不带 SOM.ind

图9以图解形式说明了默认会话期间用于物理定址请求消息(含SOM.ind)的客户端和服务器内的计时处理。



图例：

- 1 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送请求消息。
- 2 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage (消息起始) 的接收信息。
- 3 服务器T_Data.ind: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器通过P2Server=P2Server_max默认值启动P2Server计时器。

客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息接收确认信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。PClient计时器的值应考虑到基于车辆网络设计（网关通信、总线带宽等）所涉及到的潜在因素。为简单起见，图中假定客户端与服务器位于同一网络内。

4 服务端T_Data.req: 诊断应用程序已编写好响应消息，并向P2Server内的传输/网络层发送一条T_Data.req。服务器停止在P2Server计时器。

5 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage (消息起始) 的接收信息。客户端停止PClient计时器。

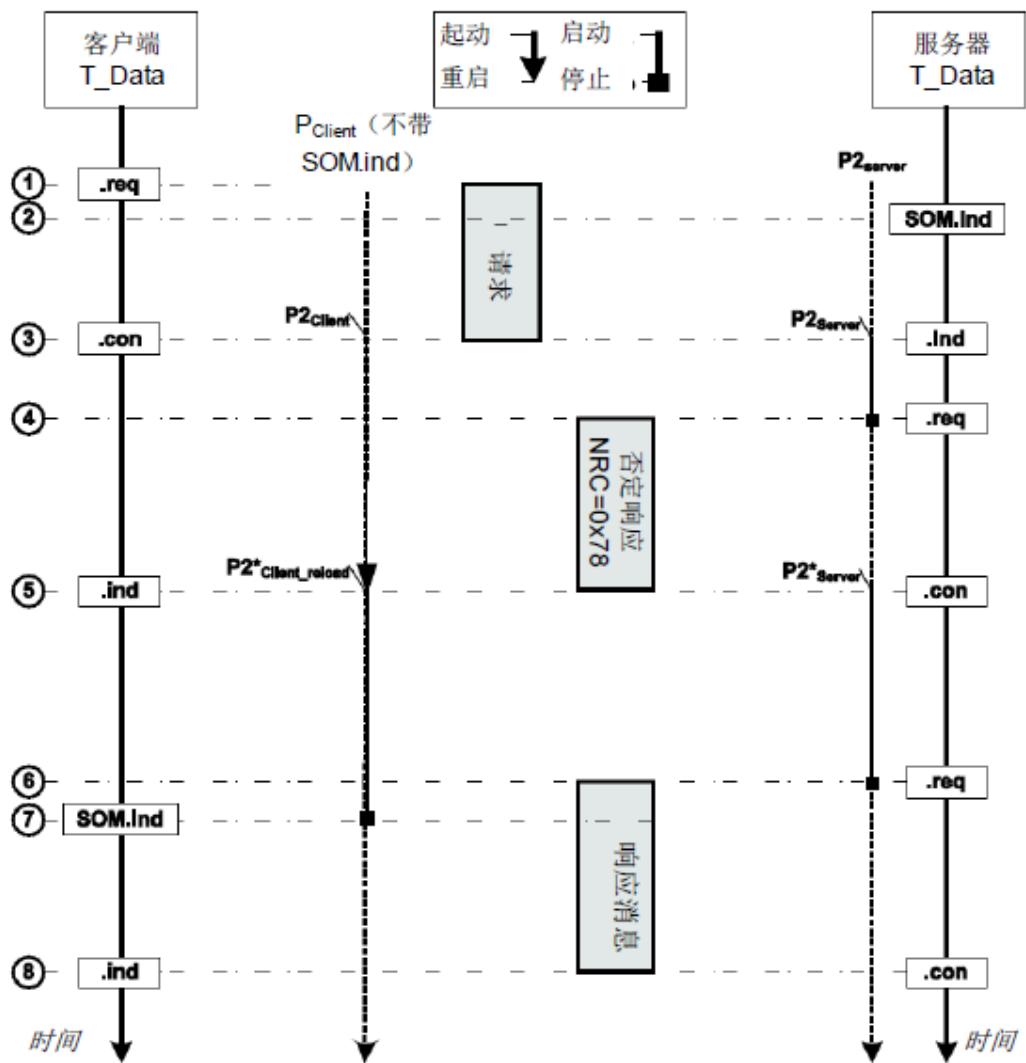
6 服务端T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送请求信息完成信息。

图10 默默认会话期间的功能通信--含 SOM.ind

6. 4. 1. 3 defaultSession (默认会话) 期间的物理通信--含强化响应计时

图10以图解形式说明了默认会话期间用于物理定址请求消息的客户端和服务器内的计时处理和强化响应计时服务器的请求（否定响应代码0x78处理）。



图例

- 1 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送请求消息。
- 2 服务器T_DataSOM.ind: 如果传输 / 网络层支持T_DataSOM.ind界面, 则传输 / 网络层向诊断应用应用程序请求消息StartOfMessage (消息起始) 的接收信息。
- 3 服务器T_Data.ind: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器通过时间P2Server=P2Server_max默认值启动P2Server计时器。
客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息接收确认信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。
- 4 服务器T_Data.req: 诊断应用程序未包含编写好的肯定响应消息, 并通过T_Data.req向P2Server内的传输 / 网络层发送含NRC=0x78的否定响应消息。服务器停止P2Server计时器。
- 5 服务器T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器通过P2*Server=P2*Server_max值 (默认强化计时) 启动P2Server计时器。
在此情况下, 服务器仍不能提供强化P2*Server内所需信息, 此时, 服务器可发送含否定响应代码0x78的否定响应消息。这将导致客户端通过强化重载值P2Client重启其PClient计时器。为简单起见, 图中仅显示含否定响应代码0x78的单条否定响应消息。
- 客户端T_Data.ind: 传输/网络层向诊断应用程序发送响应消息接收信息。客户端停止 PClient 计时器并通过P2*Client=P2*Client_max值 (默认强化计时) 重载PClient。
- 6 服务器T_Data.req: 诊断应用程序向传输/网络层发送响应消息(肯定响应, 而非否定响应代码0x78)。服务器停止P2Server计时器。
- 7 客户端T_DataSOM.ind: 如果传输 / 网络层支持T_DataSOM.ind界面, 则传输 / 网络层向诊断应用程序发送响应消息StartOfMessage (消息起始) 的接收信息。客户端停止PClient计时器。
- 8 服务器T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。
客户端T_Data.ind: 传输/网络层向诊断应用程序发送响应消息接收信息。接收到此提示后, 如果传输/网络协议不支持T_DataSOM.ind界面, 则客户端停止其PClient计时器。

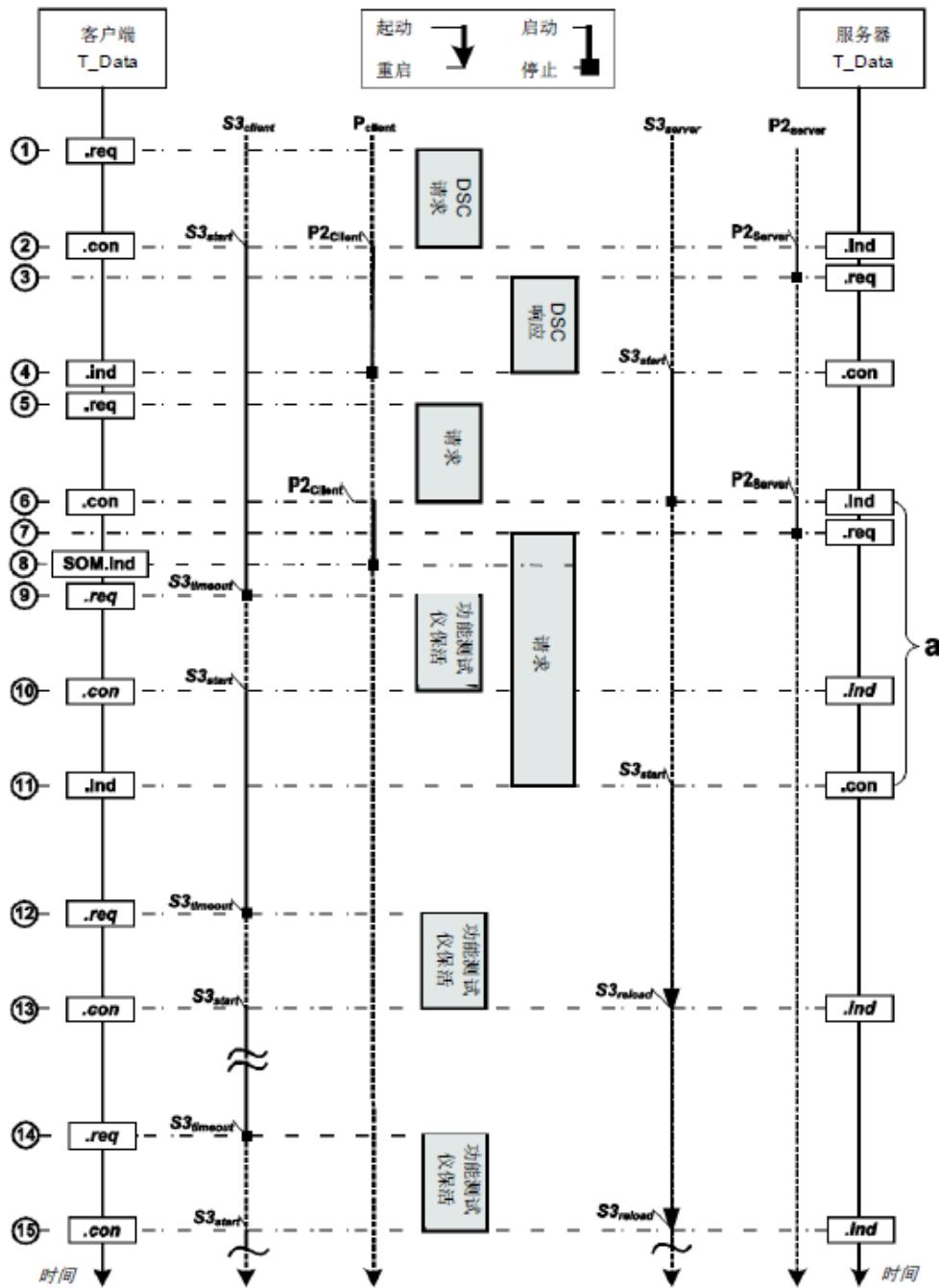
图11 默认会话期间物理通信--强化响应计时

6. 4. 1. 4 非默认会话期间的物理通信

6. 4. 1. 4. 1 功能性寻址 TesterPresent (测试仪保活) (0x3E) 消息

图11以图解形式说明了非默认会话 (如编程会话) 期间执行物理通信和通过功能定址, 周期性传输TesterPresent (测试仪保活) (0x3E) 请求消息 (不要求接收来自服务器的响应消息) 时客户端和服务器内的计时处理。

PClient和P2Server计时与8.1.2所述处理相同。唯一例外是客户端侧重载值和服务器应发送其最终响应时间所用的时间可能不同。其基本条件是转换为除默认会话的会话, 且此会话过程中, 不同PClient计时参数适用 (见第5章中DiagnosticSessionControl (诊断会话控制) (0x10)服务以了解如何向客户端报告计时参数相关的详细信息)。



图例

服务器将忽略停用S3Server计时器时接收的任何testerPresent（测试仪保活）。

1 客户端T_Data.req: 诊断应用程序向传输/网络层发送DiagnosticSessionControl（诊断会话控制）(0x10)请求消息。

传输 / 网络层向服务器传输请求消息。

2 客户端T_Data.con: 传输/网络层向诊断应用程序发送DiagnosticSessionControl（诊断会话控制）(0x10)请求消息的接收信息。8.1.2中所述响应计时PClient适用。客户端中生成的T_Data.con可启动S3Client计时器（会话计时器）。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。

服务器T_Data.ind: 传输/网络层向诊断应用程序发送DiagnosticSessionControl（诊断会话控制）(0x10)请求消息的完成信息。8.1.2中所述响应计时P2Server适用。

3 服务器T_Data.req: 诊断应用程序向传输/网络层发送DiagnosticSessionControl（诊断会话控制）(0x10)肯定响应消息。上图中，假定客户端要求服务器给出响应。

4 服务器T_Data.con: 服务器通过T_Data.con显示响应消息传输的完成服务器启动其S3Server计时器，只要不超时，可以一直保持激活的非默认会话处于激活状态。客户端应负责确保在S3Server计时器超时之前对其进行重置，以使服务器保持在非默认会话状态。

客户端T_Data.ind: 传输/网络层向诊断应用程序发送请求信息结束信息。客户端停止PClient计时器。

5 客户端T_Data.req: 诊断应用程序向传输/网络层发送请求消息。

6 服务器T_Data.ind: 传输/网络层向诊断应用程序发送请求消息完成信息。无论何时服务器在处理任何诊断服务时都会停止其S3Server计时器。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。

客户端T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。

7 服务器T_Data.req: 诊断应用程序向传输/网络层发送肯定响应消息。

8 客户端T_DataSOM.ind: 如果传输/网络层支持T_DataSOM.ind界面, 则传输/网络层向诊断应用程序发送StartOfMessage (消息起始) 的接收信息。T_DataSOM.ind interface is supported by the transport/network layer. 客户端停止PClient计时器。

9 客户端T_Data.req: 客户端中的S3Client计时器一旦开始计时, 每次S3Client计时器超时就会传输功能定址TesterPresent (测试仪保活) (0x3E)请求消息, 这种情况下不需要响应消息。

10 客户端T_Data.con: 表明已通过其传输 / 网络层的T_Data.con完成TesterPresent (测试仪保活) (0x3E)请求消息的传输后, 客户端再次启动其S3Client计时器。这意味着每次S3Client超时都会计时发送功能定址TesterPresent (测试仪保活) (0x3E)请求消息。

服务器T_Data.ind: 服务器可忽略处理另一个请求消息过程中接收到的任何TesterPresent (测试仪保活) (0x3E) 请求消息, 因为其已停止S3Server计时器并在服务完成后重启计时器。

11 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息完成信息。

服务器T_Data.con: 完全处理好诊断服务后, 服务器重启其S3Server计时器。这意味着任何诊断服务 (包括TesterPresent (测试仪保活) (0x3E)) 重置S3Server计时器。开始接收请求消息 (T_DataSOM.ind或T_Data.ind接收) 到最终响应消息传输完成 (此时需要响应消息) 或完成请求引起的任何动作 (此时不需要响应消息, 到达该时间点后将引起响应消息起始) 期间的任何时候都将诊断服务视为正在进行中。

12 客户端T_Data.req: 客户端中的S3Client计时器一旦开始计时, 每次S3Client计时器超时就会传输功能定址TesterPresent (测试仪保活) (0x3E)请求消息, 这种情况下不需要响应消息。

13 客户端T_Data.con: 显示通过其传输 / 网络层的T_Data.con完成TesterPresent (测试仪保活) (0x3E)请求消息的传输后, 客户端再次启动其S3Client计时器。这意味着每次S3Client超时都会计时发送功能定址TesterPresent (测试仪保活) (0x3E)请求消息。服务器T_Data.ind: S3Server计时器激活期间接收到的任何TesterPresent (测试仪保活) (0x3E)请求消息都将重载S3Server计时器。

14 客户端T_Data.req: 客户端中的S3Client计时器一旦开始计时, 每次S3Client计时器超时就会传输功能定址TesterPresent(0x3E) (测试仪保活) 请求消息, 这种情况下不需要响应消息。

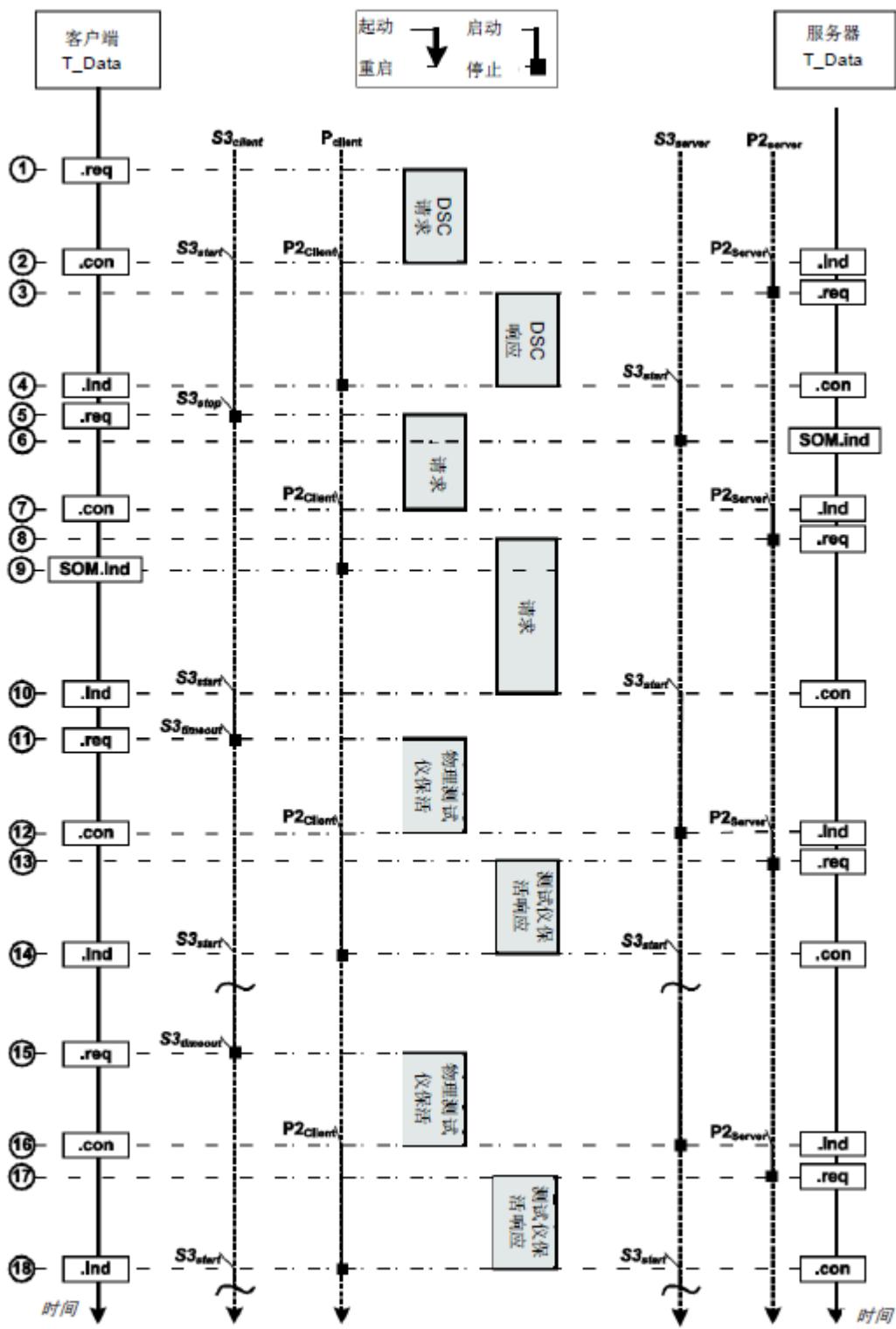
15 客户端T_Data.con: 显示通过其传输 / 网络层的T_Data.con完成TesterPresent (测试仪保活) (0x3E)请求消息的传输后, 客户端重载其S3Client计时器。这意味着每次S3Client超时都会计时发送功能定址TesterPresent (测试仪保活) (0x3E)请求消息。

服务器T_Data.ind: S3Server计时器激活期间接收到的任何TesterPresent (测试仪保活) (0x3E)请求消息都将重载S3Server计时器。

图12 非默认会话期间的物理通信--功能定址 TesterPresent (测试仪保活)

6. 4. 1. 4. 2 物理定址 TesterPresent (测试仪保活) (0x3E) 消息

图12以图解形式说明了非默认会话 (如编程会话) 期间执行物理通信和通过物理定址, 周期性传输TesterPresent (测试仪保活) (0x3E) 请求消息 (要求接收来自服务器的响应消息) 时客户端和服务器内的计时处理, 以便在无其他诊断服务的情况下使诊断会话处于激活状态。



图例:

1 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送DiagnosticSessionControl (诊断会话控制) (0x10)请求消息。
传输 / 网络层向服务器传输请求消息。

2 客户端T_Data.con: 传输 / 网络层向诊断应用程序发送DiagnosticSessionControl (诊断会话控制) (0x10)请求消息的接收信息。8.1.2中所述响应计时PClient适用。客户端中生成的T_Data.con启动S3Client计时器 (会话计时器)。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。

服务端T_Data.ind: 传输 / 网络层向诊断应用程序发送DiagnosticSessionControl (诊断会话控制) (0x10)请求消息的完成信息。8.1.2中所述响应计时P2Server适用。

3 服务端T_Data.req: 诊断应用程序向传输 / 网络层发送DiagnosticSessionControl (诊断会话控制) (0x10)肯定响应消息。上图中, 假定客户端要求服务器给出响应。

4 服务端T_Data.con: 服务端通过T_Data.con显示响应消息传输的完成服务端启动其S3Server计时器, 只要不超时, 可以一直保持激活的非默认会话处于激活状态。客户端应负责确保在S3Server计时器超时之前对其进行重置, 以使服务端保持在非默认会话状态。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送请求信息完成信息。客户端停止PClient计时器。

5 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送请求消息。无论何时客户端想服务器传输请求消息（包括物理定址TesterPresent（测试仪保活）（0x3E）消息），客户端将停止其S3Client计时器。

6 服务器T_DataSOM.ind: 如果传输 / 网络层支持T_DataSOM.ind界面，则传输 / 网络层向诊断应用程序发送StartOfMessage（消息起始）的接收信息。接收到请求消息StartOfMessage（消息起始）后，请停止服务器内S3Server计时器。

7 服务器T_Data.ind: 传输/网络层向诊断应用程序发送请求消息完成信息。

客户端T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。如果客户端不要求响应消息，则客户端将在接收到通过T_Data.con显示的请求消息完成确认信息时启动其S3Client计时器。在此情况下，服务器在其已完成请求动作时启动S3Server计时器。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。

8 服务器T_Data.req: 诊断应用程序向传输 / 网络层发送肯定响应消息。

9 客户端T_DataSOM.ind: 如果传输 / 网络层支持T_DataSOM.ind界面，则传输 / 网络层向诊断应用程序发送StartOfMessage（消息起始）的接收信息。客户端停止PClient计时器。

10 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息完成信息。客户端中生成的T_Data.con可启动S3Client计时器（会话计时器）。

服务器T_Data.con: 传输/网络层向诊断应用程序发送请求消息完成信息。服务器启动其S3Server计时器。

11 客户端T_Data.req: 如果在S3Client超时前，客户端不发送诊断请求消息，则S3Client计时器超时会促使客户端传输物理存执TesterPresent（测试仪保活）（0x3E）请求消息。

12 服务器T_Data.con: 服务器通过T_Data.con显示TesterPresent（测试仪保活）（0x3E）请求消息接收信息。这将导致服务器停止其S3Server计时器。此时，8.1.2中所述响应计时适用。

客户端T_Data.con: 客户端通过T_Data.con显示TesterPresent（测试仪保活）（0x3E）请求消息接收信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。

13 服务器T_Data.req: 诊断应用程序向传输 / 网络层发送TesterPresent（测试仪保活）（0x3E）响应消息。

14 客户端T_Data.ind: 客户端通过T_Data.ind显示TesterPresent（测试仪保活）（0x3E）响应消息完成信息，这将促使客户端启动其S3Client计时器。

客户端T_Data.con: 服务器通过T_Data.con显示TesterPresent（测试仪保活）（0x3E）响应消息完成信息，这将促使服务器启动其S3Server。如果客户端不要求响应消息，则客户端应在接收到通过T_Data.con显示的TesterPresent（测试仪保活）（0x3E）请求消息完成确认信息时启动其S3Client计时器。服务器将在其已完成请求动作时启动S3Server计时器。为简单起见，图示表明要求响应。客户端停止PClient计时器。

15 客户端T_Data.req: 如果在S3Client超时前，客户端不发送任何诊断请求消息，则S3Client计时器超时会促使客户端传输物理定址TesterPresent（测试仪保活）（0x3E）请求消息。

16 服务器T_Data.con: 服务器通过T_Data.con显示TesterPresent（测试仪保活）（0x3E）请求消息接收信息。这将导致服务器停止其S3Server计时器。此时，8.1.2中所述响应计时适用。

客户端T_Data.con: 客户端通过T_Data.con显示TesterPresent（测试仪保活）（0x3E）请求消息接收信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。

17 服务器T_Data.req: 诊断应用程序向传输 / 网络层发送TesterPresent（测试仪保活）（0x3E）响应消息。

18 客户端T_Data.ind: 客户端通过T_Data.ind显示TesterPresent（测试仪保活）（0x3E）响应消息完成信息，这将促使客户端启动其S3Client计时器。

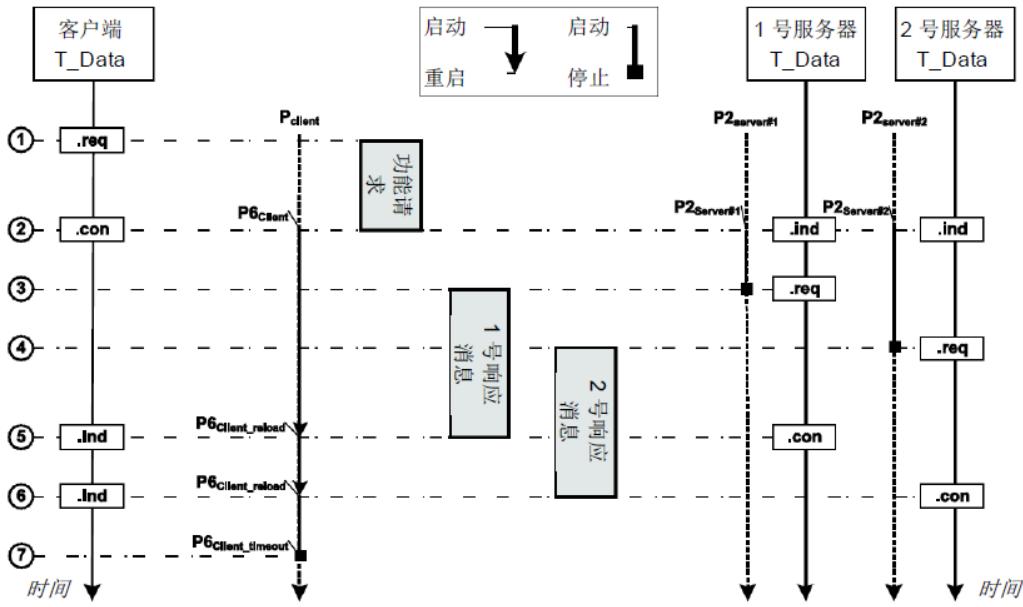
服务器T_Data.con: 服务器通过T_Data.con显示TesterPresent（测试仪保活）（0x3E）响应消息完成信息，这将促使服务器启动其S3Server。如果客户端不要求响应消息，则客户端应在接收到通过T_Data.con显示的TesterPresent（测试仪保活）（0x3E）请求消息完成确认信息时启动其S3Client计时器。服务器将在其已完成请求动作时启动S3Server计时器。为简单起见，图示表明要求响应。

图13 非默认会话期间的物理通信--物理定址 TesterPresent（测试仪保活）

6. 4. 2 功能通信

6. 4. 2. 1 DefaultSession（默认会话）期间的功能通信——不带 SOM.ind

图13以图解形式说明了默认会话期间用于功能定址请求消息的客户端和两个服务器中的计时处理。就服务器而言，与物理定址请求消息相比，计时处理方面并无不同之处，但是与物理通信相比，客户端处理计时的方式应有所不同。



图例：

1 客户端T_Data.req：诊断应用程序向传输 / 网络层发送功能定址请求消息。

2 所有服务器T_Data.ind：传输 / 网络层向诊断应用程序发送请求消息的完成信息。所有服务器均通过P2Server=P2Server_max值启动P2Server计时器。

客户端T_Data.con：传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过P6Client=P6Client_max默认重载值启动其PClient计时器。PClient计时器的值应考虑到基于车辆网络设计（例如，网关通信、总线带宽等）所涉及到的潜在因素。为简单起见，图中假定客户端与服务器位于同一网络内。

3 1号服务器T_Data.req：诊断应用程序已编写好响应消息，并向P2Server内的传输/网络层发送一条T_Data.req。服务器停止P2Server计时器。

4 2号服务器T_Data.req：诊断应用程序已编写好响应消息，并向P2Server_max内的传输/网络层发送一条T_Data.req。服务器停止P2Server计时器。

5 1号服务器T_Data.con：传输 / 网络层向诊断应用程序发送响应消息的完成信息。

客户端T_Data.ind：传输 / 网络层向诊断应用程序发送响应消息的完成信息。这将导致客户端通过P6Client_max默认重载值重启其PClient计时器。

6 2号服务器T_Data.con：传输 / 网络层向诊断应用程序发送响应消息的完成信息。

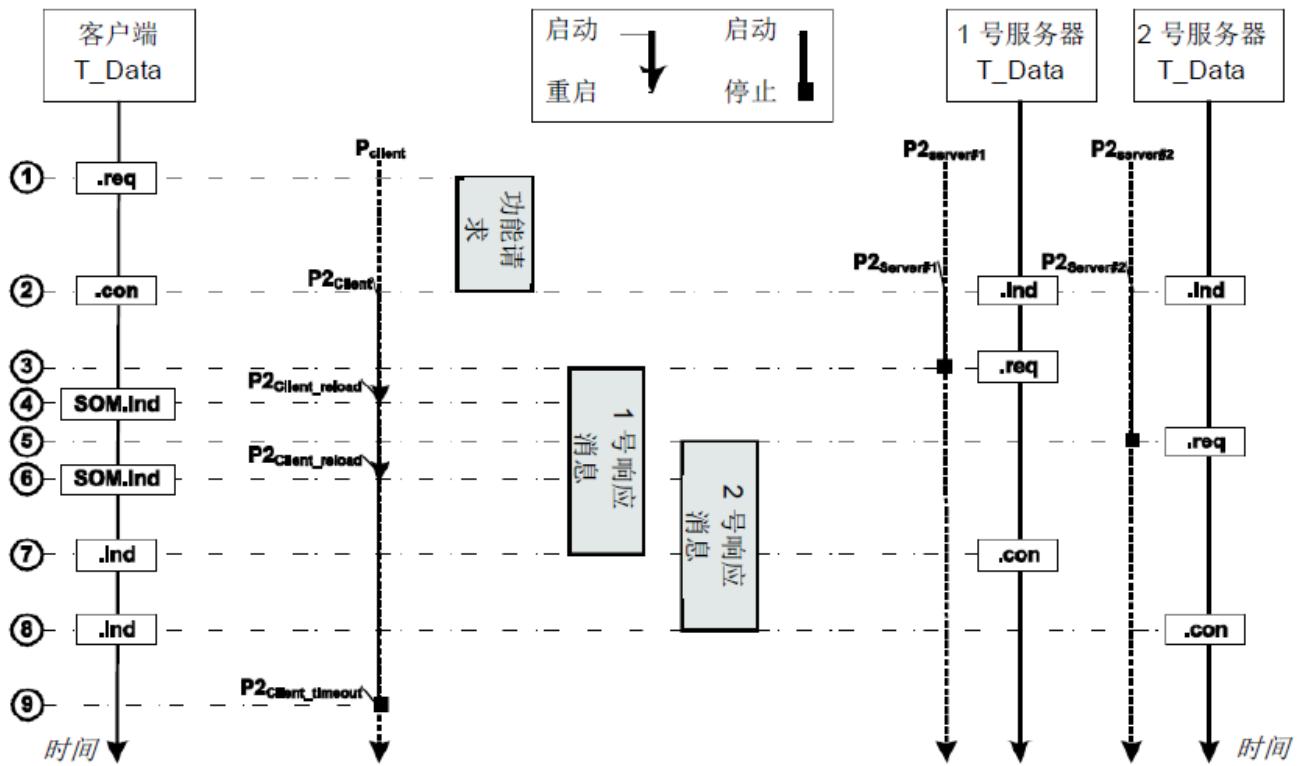
客户端T_Data.ind：传输 / 网络层向诊断应用程序发送响应消息的完成信息。这将导致客户端通过P6Client_max默认重载值重启其PClient计时器。

7 客户端：表明客户端预计不会有更多响应消息且能够继续发出更多请求。

图14 默认会话期间的功能通信——不带 SOM.ind

6. 4. 2. 2 DefaultSession（默认会话）期间的功能通信——含 SOM.ind

图14以图解形式说明了默认会话期间用于功能定址请求消息的客户端和两个服务器中的计时处理。就服务器而言，与物理定址请求消息相比，计时处理方面并无不同之处，但是，与物理通信相比，客户端处理计时的方式应有所不同。



图例：

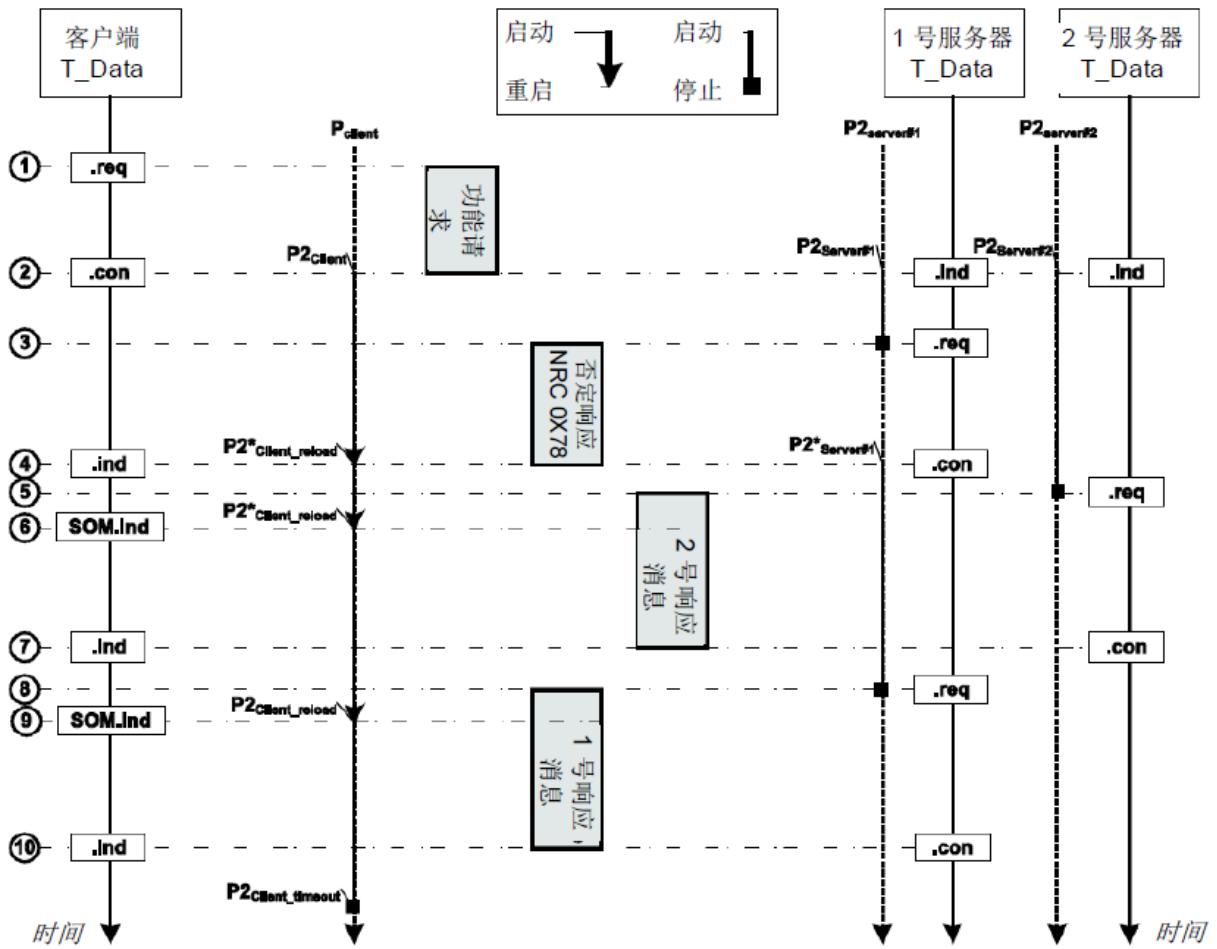
- 1 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送功能定址请求消息。
- 2 所有服务器T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。所有服务器均通过P2Server=P2Server_max值启动P2Server计时器。
- 3 客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。PClient计时器的值应考虑到基于车辆网络设计（例如，网关通信、总线带宽等）所涉及到的潜在因素。为简单起见，图中假定客户端与服务器位于同一网络内。
- 4 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage（消息起始）的接收信息。这将导致客户端通过P2Client_max默认重载值重启其PClient计时器。
- 5 2号服务器T_Data.req: 诊断应用程序已编写好响应消息，并向P2Server内的传输/网络层发送一条T_Data.req。2号服务器停止P2Server计时器。
- 6 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage（消息起始）接收信息。这将导致客户端通过P2Client_max默认重载值重启其PClient计时器。
- 7 1号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。
- 8 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。
- 9 2号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。
- 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。
- 9 客户端：表明客户端预计不会有更多响应消息且能够继续发出更多请求。

图15 默认会话期间的功能通信——不带 SOM.ind

6.4.2.3 含强化响应计时的 defaultSession (默认会话) 期间的功能通信——不带 SOM.ind

图15以图解形式说明了默认会话期间用于功能定址请求消息的客户端和两个服务器中的计时处理，其中一个服务器通过否定响应消息（含否定响应代码0x78）来请求强化响应计时。

就服务器而言，与物理定址请求消息相比，计时处理方面并无不同之处，但是与物理通信相比，客户端处理计时的方式应有所不同。



图例：

1 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送功能定址请求消息。

2 所有服务器T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。所有服务器均通过P2Server=P2Server_max值启动P2Server计时器。

客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。NRC 0x78未决名单为空。PClient计时器的值应考虑到基于车辆网络设计（例如，网关通信、总线带宽等）所涉及到的潜在因素。为简单起见，图中假定客户端与服务器位于同一网络内。

3 1号服务器T_Data.req: 诊断应用程序还未编写好肯定响应消息，通过T_Data.req向P2Server内的传输 / 网络层发送含NRC=0x78的否定响应消息。1号服务器停止P2Server计时器。如果有任何寻址找到的服务器无法在P2Server响应计时内提供请求信息，则其可以通过发送一条否定响应消息（包括否定响应代码0x78）来请求强化响应计时。

4 1号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。1号服务器通过P2*Server#1=P2*Server_max值（默认强化计时）启动P2Server计时器。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的接收信息。客户端通过P2*Client=P2*Client_max值（默认强化计时）重启PClient计时器。将含有响应消息地址的一个条目增加到NRC 0x78未决名单中。

5 2号服务器T_Data.req: 诊断应用程序已编写好响应消息，并向P2Server内的传输/网络层发送一条T_Data.req。2号服务器停止P2Server计时器。

6 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage（消息起始）接收信息。这将导致客户端通过P2*Client默认重载值（默认强化计时）重启其PClient计时器。

7 2号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。

8 1号服务器T_Data.req: 诊断应用程序已编写好响应消息，并向P2*Server内的传输/网络层发送一条T_Data.req。1号服务器停止P2Server计时器。

9 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage（消息起始）接收信息。从未决名单中删除与响应响应地址匹配的条目，现在名单已空。这意味着所有响应消息均不再处于未决状态。这将导致客户端通过P2Client默认重载值重启其PClient计时器。

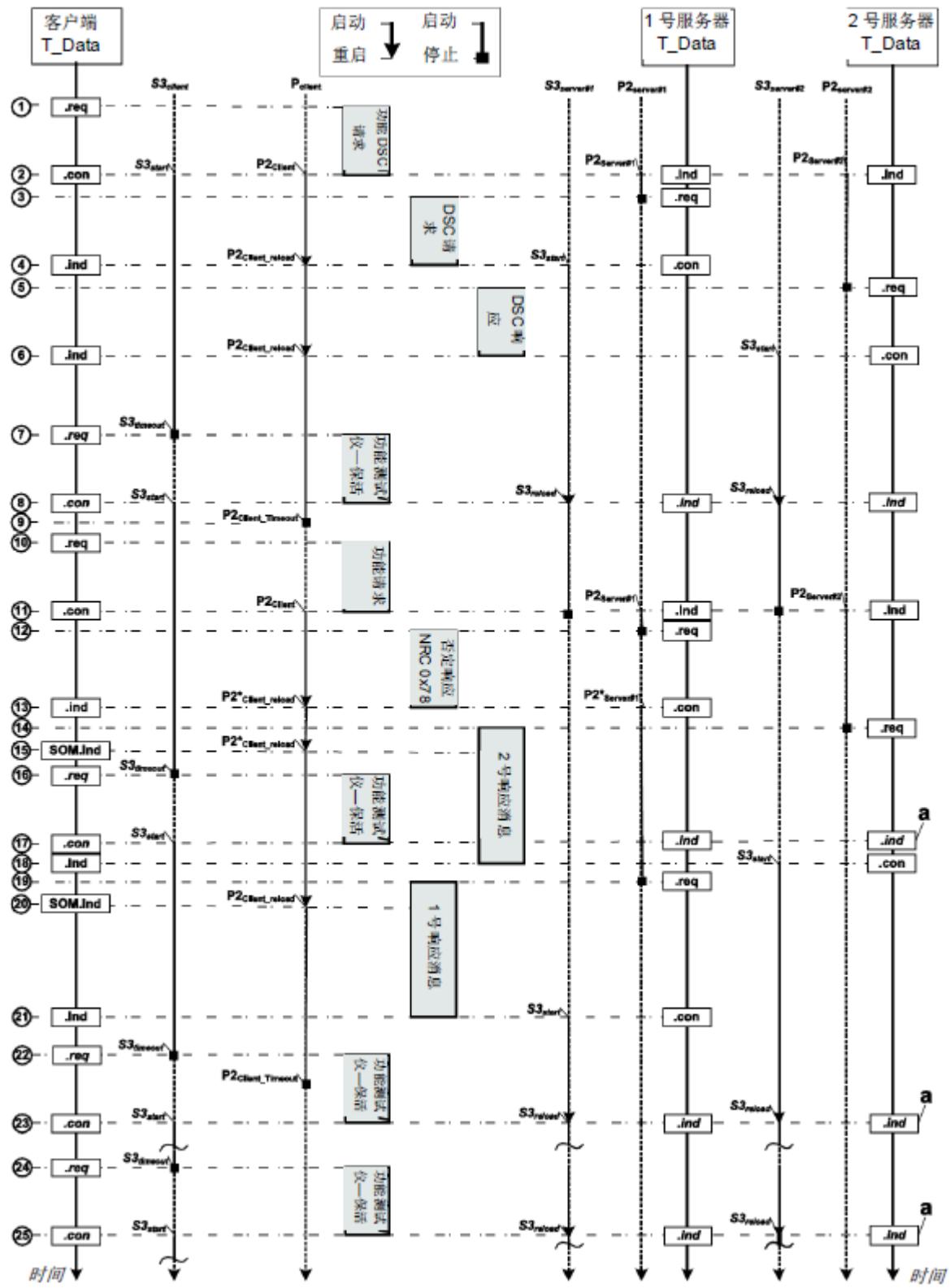
10 1号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。

图16 默认会话期间的功能通信——强化响应计时——含 SOM.ind

6. 4. 2. 4 非默认会话期间的功能通信——不带 SOM.ind

图16以图解形式说明了非默认会话期间（例如programmingSession（编程会话））用于功能定址请求消息的客户端和两个服务器中的计时处理，其中一个服务器通过否定响应消息（含否定响应代码0x78）来请求强化响应计时。



图例

a 服务器将忽略停用S3Server计时器时接收的任何TesterPresent（测试仪保活）。

1 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送功能定址DiagnosticSessionControl（诊断会话控制）(0x10)请求消息。传输 / 网络层向服务器传输请求消息。

2 客户端T_Data.con: 传输 / 网络层向诊断应用程序发送DiagnosticSessionControl（诊断会话控制）(0x10)请求消息的接收信息。8.1.2和8.1.3中所述响应计时PCClient适用。PCClient计时器的值应考虑到基于车辆网络设计（例如，网关通信、总线带宽等）所涉及到的潜在因素。为简单起见，图中假定客户端与服务器位于同一网络内。客户端中生成的 T_Data.con启动S3Client计时器（会话计时器）。

所有服务器T_Data.ind: 传输 / 网络层向诊断应用程序发送DiagnosticSessionControl (诊断会话控制) (0x10) 请求消息的完成信息。所有服务器均通过P2Server=P2Server_max值启动P2Server计时器。8.1.2和8.1.3中所述响应计时P2Server适用。

3 1号服务器T_Data.req: 诊断应用程序向P2Server_max内的传输 / 网络层发送DiagnosticSessionControl(0x10) (诊断会话控制) 肯定响应消息。1号服务器停止P2Server计时器。上图中, 假定客户端要求服务器给出响应。

4 1号服务器T_Data.con: 1号服务器通过T_Data.con表明DiagnosticSessionControl (诊断会话控制) (0x10)肯定响应消息的完成信息。这时, 1号服务器启动其S3Server计时器, 只要不超时, 可以一直保持激活的非默认会话处激活状态。客户端应负责确保在S3Server计时器超时之前对其进行重置, 以便1号服务器保持在非默认会话状态。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。这将导致客户端通过P2Client默认重载值重启其PClient计时器。

5 2号服务器T_Data.req: 诊断应用程序向P2Server内的传输 / 网络层发送DiagnosticSessionControl (诊断会话控制) (0x10)肯定响应消息。2号服务器停止P2Server计时器。上图中, 假定客户端要求服务器给出响应。

6 2号服务器T_Data.con: 2号服务器通过T_Data.con显示DiagnosticSessionControl (诊断会话控制) (0x10)肯定响应消息的完成信息。这时, 2号服务器启动其S3Server计时器, 只要不超时, 可以一直保持激活的非默认会话处于激活状态。客户端应负责确保在S3Server计时器超时之前对其进行重置, 以便2号服务器保持在非默认会话状态。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。这将导致客户端通过P2Client默认重载值重启其PClient计时器。

7 客户端T_Data.req: 客户端中的S3Client计时器一旦开始计时, 每次S3Client计时器超时就会传输功能定址TesterPresent (测试仪保活) (0x3E)请求消息, 这种情况下不需要响应消息。

8 客户端T_Data.con: 表明已通过其传输 / 网络层的T_Data.con完成TesterPresent (测试仪保活) (0x3E)请求消息的传输后, 客户端再次启动其S3Client计时器。这意味着每次S3Client超时都会定时发送功能定址TesterPresent (测试仪保活) (0x3E)请求消息。

所有服务器T_Data.ind: S3Server计时器激活期间接收到任何TesterPresent (测试仪保活) (0x3E)请求消息都将重新启动S3Server计时器。

9 客户端: 表明客户端预计不会有更多响应消息且能够继续发出更多请求。

10 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送功能定址请求消息。

11 所有服务器 T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。所有服务器均通过P2Server=P2Server_max值启动P2Server计时器。无论何时服务器在处理任何诊断服务时都会停止其S3Server计时器。

客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过P2Client=P2Client_max默认重载值启动其PClient计时器。PClient计时器的值应考虑到基于车辆网络设计 (例如, 网关通信、总线带宽等) 所涉及到的潜在因素。为简单起见, 图中假定客户端与服务器位于同一网络内。

12 1号服务器T_Data.req: 诊断应用程序还未编写好肯定响应消息, 通过T_Data.req向P2Server内的传输 / 网络层发送含NRC=0x78的否定响应消息。1号服务器停止P2Server计时器。如果有任何寻址找到的服务器无法在P2Server响应计时内提供请求信息, 则其可以通过发送一条否定响应消息 (包括否定响应代码0x78) 来请求强化响应计时。

13 1号服务器 T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。1号服务器通过P2*Server=P2*Server_max值 (默认强化计时) 启动P2Server计时器。

客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的接收信息。客户端通过P2*Client=P2*Client_max值 (默认强化计时) 重启PClient计时器。将含有响应消息地址的一个条目增加到NRC 0x78未决名单中。

14 2号服务器T_Data.req: 诊断应用程序已编写好响应消息, 并向P2Server内的传输/网络层发送一条T_Data.req。2号服务器停止P2Server计时器。

15 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage (消息起始) 接收信息。这将导致客户端通过P2*Client默认重载值 (默认强化计时) 重启其PClient计时器。

16 客户端T_Data.req: 客户端中的S3Client计时器一旦开始计时, 每次S3Client计时器超时就会传输功能定址TesterPresent (测试仪保活) (0x3E)请求消息, 这种情况下不需要响应消息。

17 客户端T_Data.con: 表明已通过其传输 / 网络层的T_Data.con完成TesterPresent (测试仪保活) (0x3E)请求消息的传输后, 客户端再次启动其S3Client计时器。这意味着每次S3Client超时都会定时发送功能定址TesterPresent (测试仪保活) (0x3E)请求消息。

所有服务器T_Data.ind: S3Server计时器激活期间接收到任何TesterPresent (测试仪保活) (0x3E)请求消息都将重新启动S3Server计时器。服务器将忽略停用S3Server计时器时接收的任何TesterPresent (测试仪保活) 。

18 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。

2号服务器T_Data.con: 完全处理好诊断服务后, 2号服务器重启其S3Server计时器。这意味着任何诊断服务 (包括TesterPresent (测试仪保活) (0x3E)) 重置S3Server计时器。开始接收请求消息 (T_DataSOM.ind或T_Data.ind接收) 到最终响应消息传输完成 (此时需要响应消息) 或完成请求引起的任何动作 (此时不需要响应消息, 到达该时间点后将引起响应消息起始) 期间的任何时候都将诊断服务视为正在进行中。

19 1号服务器T_Data.req: 诊断应用程序已编写好响应消息, 并向P2*Server内的传输/网络层发送一条T_Data.req。1号服务器停止P2Server计时器。

20 客户端T_DataSOM.ind: 传输 / 网络层向诊断应用程序发送StartOfMessage (消息起始) 接收信息。从未决名单中删除与响应响应地匹配的条目, 现在名单已空。这意味着所有响应消息均不再处于未决状态。这将导致客户端通过P2Client默认重载值重启其PClient计时器。

21 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。

1号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。这时, 1号服务器启动其S3Server计时器。

22 客户端T_Data.req: S3Client计时器每次超时都会引起功能定址TesterPresent (测试仪保活) (0x3E)请求消息的传输, 这种情况不需要响应消息。

23 客户端T_Data.con：表明已通过其传输 / 网络层的T_Data.con完成TesterPresent（测试仪保活）(0x3E)请求消息的传输后，客户端再次启动其S3Client计时器。这意味着每次S3Client超时都会定时发送功能定址TesterPresent（测试仪保活）(0x3E)请求消息。

所有服务器T_Data.ind：S3Server计时器激活期间接收到任何TesterPresent（测试仪保活）(0x3E)请求消息都将重新启动S3Server计时器。

24 客户端T_Data.req：客户端中的S3Client计时器一旦开始计时，每次S3Client计时器超时就会传输功能定址TesterPresent（测试仪保活）(0x3E)请求消息，这种情况下不需要响应消息。

25 客户端T_Data.con：表明已通过其传输 / 网络层的T_Data.con完成TesterPresent（测试仪保活）(0x3E)请求消息的传输后，客户端再次启动其S3Client计时器。这意味着每次S3Client R超时都会定时发送功能定址TesterPresent（测试仪保活）(0x3E)请求消息。

所有服务器T_Data.ind：S3Server计时器激活期间接收到任何TesterPresent（测试仪保活）(0x3E)请求消息都将重新启动S3Server计时器。

图17 非默认会话期间的功能通信—含 SOM.ind

PClient和P2Server计时的处理与8.1.2和8.1.3中处理方式相同，唯一的不同点在于客户端侧重载值和服务器应发送其最终响应时间所用的时间可能不同。其基本条件是转换为非默认会话的、可适用不同PClient计时参数的会话（见第5章中DiagnosticSessionControl(0x10)（诊断会话控制）服务以了解如何向客户端报告计时参数相关的详细信息）。

6.4.3 客户端请求消息之间的最短时间

要求客户端传输的请求消息之间达到最短时间，以便允许服务器中出现轮询驱动服务数据解释。以正常功能为基础，服务器可能以设计指定的进度速率（例如，10ms）来处理诊断请求消息。诊断服务数据解释计划程序所用时间应小于性能需求P2Server，以符合8.1.1、8.1.2及8.1.3中规定的服务器。

请求消息之间的最短时间用计时参数分为以下两个计时参数：

——P3Client_Func：该计时参数适用于任何功能定址请求消息，因为可能出现以下这种情况，即如果服务器不支持请求的数据，则其不需要对功能定址请求消息做出响应。

——P3Client_Phys：该计时参数适用于服务器不需要传输任何响应时(suppressPosRspMsgIndicationBit=TRUE)的任何物理定址请求消息。

服务器需要做出响应时的物理通信情况下，客户端可以在完全接收上一个响应消息后立即传输下一个请求，因为服务器已对请求做出完整响应——这意味着完全由服务器来处理请求。

图17以图解形式说明了客户端已确定所有预期服务器已对之前的请求消息做出响应后立即传输下一条请求时，功能通信期间可能出现的问题的示例。该场景不仅适用于功能定址请求，还适用于客户端不想接收任何响应消息时(suppressPosRspMsgIndicationBit=TRUE)的物理定址请求。

为了处理上述场景，为客户端定义了物理或功能定址请求消息结束到开始新的物理或功能定址请求消息之间的最短时间P3Client_Phys和P3Client_Func：

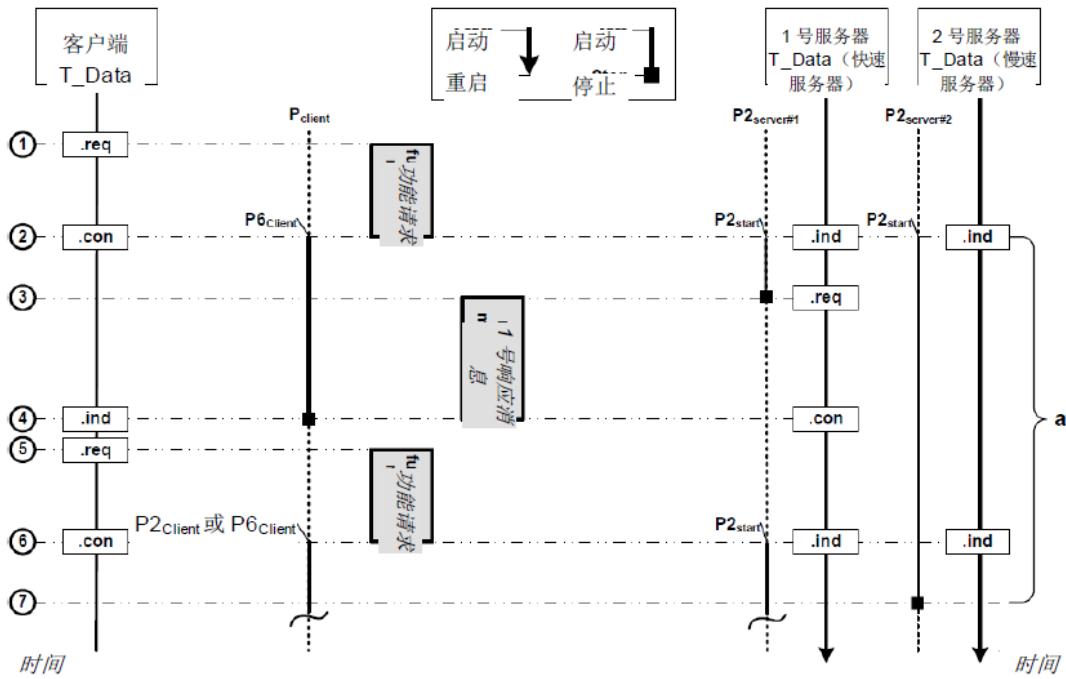
a) 值 P3Client_Phys 将与物理定址服务器的 P2Server_max 相同。计时适用于任何诊断会话（默认及非默认会话）中的任何物理定址请求消息，服务器未要求响应时亦适用。

每次不要求响应的物理定址请求消息成功地传输到总线时（通过客户端中的 T_Data.con 给出指示），客户端中的 P3Client_Phys 计时器就会启动。若客户端欲在已完全处理一条请求后接着传输一条新的物理定址请求消息，则仅在客户端欲传输物理定址请求消息时 P3Client_Phys 计时器不再有效的情况下方可允许进行传输。客户端欲传输一条新的物理定址请求消息时如果 P3Client_Phys 仍处于有效状态，则应推迟传输直至 P3Client_Phys 超时。

b) 值 P3Client_Func 将是用于任何诊断会话（默认及非默认会话）中任何功能定址请求消息的所有功能定址服务器 P2Server_max 的最大（最糟）值。

每次要求响应或不要求响应的功能定址请求消息成功地传输到总线时（通过客户端中的 T_Data.con 给出指示），客户端中的 P3Client_Func 计时器就会启动。若客户端欲在已完全处理一条请求后接着传输一条新的功能定址请求消息，则仅在客户端欲传输功能定址请求消息时 P3Client_Func 计时器不再有效的情况下方可允许进行传输。客户端欲传输一条新的功能定址请求消息时如果 P3Client_Func 仍处于有效状态，则应推迟传输直至 P3Client_Func 超时。

注：“完全处理”指，不要求任何响应的情况下未接收到任何响应，或已知响应的服务器且要求响应的情况下已接收到功能定址请求的所有预期响应，或响应的服务器未知且要求响应的情况下发生PClient超时这三种情形。服务器的要求是应在P2Server内以其响应消息起始。这意味着服务器的诊断数据解释速率应小于P2Server_max。



a 2号服务器的诊断服务数据解释速率

1 客户端 T_Data.req: 诊断应用程序向传输 / 网络层发送请求消息。

2 客户端 T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过 P2Client=P2Client_max 默认重载值启动其 PClient 计时器。

所有服务器 T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。所有服务器通过 P2Server=P2Server_max 默认值启动 P2Server 计时器。

3 1号服务器 T_Data.req: 诊断应用程序已编写好响应消息，并向 P2Server 内的传输/网络层发送一条 T_Data.req。1号服务器停止 P2Server 计时器。对于请求消息，假定仅1号服务器支持所请求的信息，这意味着 2号服务器无响应。1号服务器是一个快速服务器，能够立即处理已接收的请求消息，并在 P2Server 内传输其响应。

4 客户端 T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。客户端预期只从1号服务器得到响应消息，因此客户端停止其计时器 PClient。

1号服务器 T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。

5 客户端 T_Data.req: 诊断应用程序向传输 / 网络层发送请求消息。完成所有预期的响应消息后，客户端将立即发送下一个请求。

6 客户端 T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过 P2Client=P2Client_max 默认重载值（如果支持 T_DataSOM.ind）或 P6Client=P6Client_max 启动其 PClient 计时器。

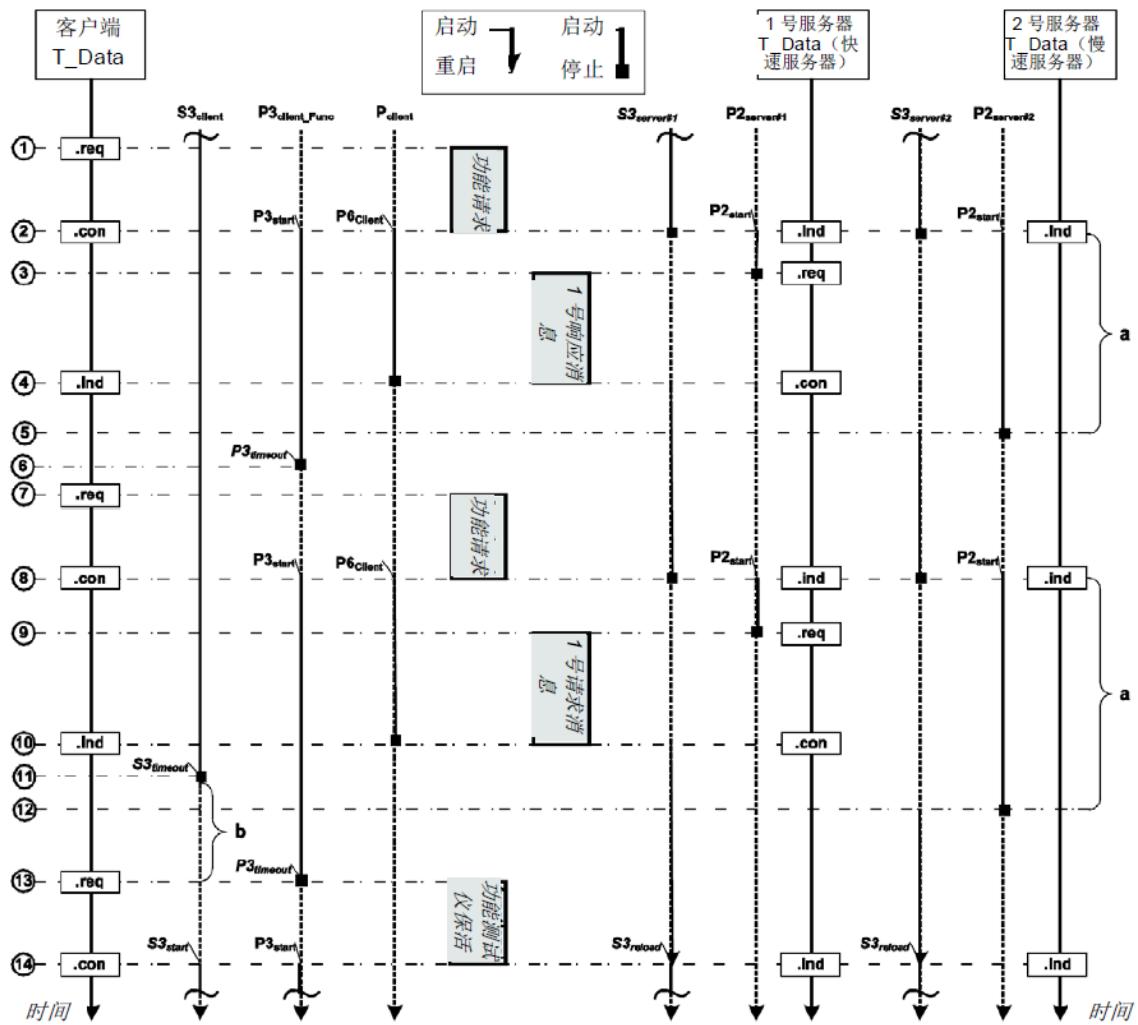
1号服务器 T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。1号服务器通过 P2Server=P2Server_max 默认值启动 P2Server 计时器。

2号服务器 T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。仅由1号服务器来处理请求消息，因为2号服务器还未处理之前的请求。2号服务器将忽略2号服务器诊断服务数据解释速率期间接收到的任何请求消息。

7 2号服务器: 2号是一个慢速服务器，可（以诊断服务数据解释速率）定时解释已接收的请求。在最糟糕的情况下，最后一次检查是否有传入请求消息也是发生在传输 / 网络层接收功能定址请求消息之前。这意味着该请求将存储在缓冲区，下次计划程序检查是否有传入请求时将最早处理该请求。2号服务器处理请求时，该服务器决定其不需要回应，因为其不支持所请求的信息（例如，服务器不发送任何响应消息，因为功能定址请求消息的情况下通常会取消否定响应代码 requestOutOfRange （请求超过限值））。如图所示，这发生在完成1号服务器的响应消息后，甚至是发生在客户端完成下一条请求消息的传输后。

图18 过早传输下一条请求时的关键问题示例

图18以图解形式说明了客户端的P3Client_Func计时处理（根据图16中阐述的通信场景进行说明）。此外，图18还说明了在S3Client超时时P3Client_Func计时器仍有效这种情况下客户端中的功能定址TesterPresent（测试仪保活）(0x3E)请求消息的处理（请求将推迟直到至P3Client_Func超时）。



a 2号服务器的诊断服务数据解释速率

b 功能TesterPresent (测试仪保活) 延迟

1 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送请求消息。

2 客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过P2Client=P2Client_max默认重载值（如果支持T_DataSOM.ind）或P6Client=P6Client_max以及其P3Client_Func计时器启动其PClient计时器。

所有服务器T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。所有服务器通过P2Server=P2Server_max默认值启动P2Server计时器。所有服务器均停止S3Server计时器。

3 1号服务器T_Data.req: 诊断应用程序已编写好响应消息，并向P2Server内的传输/网络层发送一条T_Data.req。1号服务器停止P2Server计时器。对于请求消息，假定仅1号服务器支持所请求的信息，这意味着2号服务器无响应。1号服务器是一个快速服务器，能够立即处理已接收的请求消息，并在P2Server内传输其响应。

4 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。客户端预期只从1号服务器得到响应消息，因此客户端停止其PClient计时器。

1号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。这时，1号服务器启动其S3Server计时器。

5 2号服务器: 2号是一个慢速服务器，可（以诊断服务数据解释速率）定时解释已接收的请求。在最糟糕的情况下，最后一次检查是否有传入请求消息也是发生在传输 / 网络层接收功能定址请求消息之前。这意味着该请求将存储在缓冲区，下次计划程序检查是否有传入请求时将最早处理该请求。2号服务器处理请求时，该服务器决定其不需要回应，因为其不支持所请求的信息（例如，服务器不发送任何响应消息，因为功能定址请求消息的情况下通常会取消否定响应代码requestOutOfRange（请求超过限值））。现在2号服务器停止P2Server计时器并启动其S3Server计时器。

6 客户端: 即使客户端已收到功能定址请求消息的所有预期响应消息，允许其传输下一条请求消息之前客户端还是应继续等待直到P3Client_Func超时。此时P3Client_Func超时。

7 客户端T_Data.req: 诊断应用程序向传输 / 网络层发送请求消息。完成所有预期的响应消息后，客户端将立即发送下一个请求。

8 客户端T_Data.con: 传输 / 网络层向诊断应用程序发送请求消息的完成确认信息。客户端通过P2Client=P2Client_max默认重载值（如果支持T_DataSOM.ind）或P6Client=P6Client_max以及其计时器P3Client_Func启动其PClient计时器。

所有服务器T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。所有服务器通过P2Server=P2Server_max默认值启动P2Server计时器。所有服务器均停止S3Server计时器。

9 1号服务器T_Data.req: 诊断应用程序已编写好响应消息，并向P2Server_max内的传输/网络层发送一条T_Data.req。1号服务器停止P2Server计时器。对于请求消息，假定仅1号服务器支持所请求的信息，这意味着2号服务器无响应。1号服务器是一个快速服务器，能够立即处理已接收的请求消息，并在P2Server内传输其响应。

10 客户端T_Data.ind: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。客户端预期只从1号服务器得到响应消息，因此客户端停止其PClient计时器。

1号服务器T_Data.con: 传输 / 网络层向诊断应用程序发送响应消息的完成信息。这时，1号服务器启动其S3Server计时器。

11 客户端: 客户端的S3Client计时器超时，强制客户端传输功能定址TesterPresent(0x3E)（测试仪保活）请求消息，且不需要从寻址服务器获得响应消息。根据此时P3Client_Func计时器仍处于有效状态这一情况，应推迟TesterPresent（测试仪保活）(0x3E)的传输直至P3Client_Func计时器到时间。

12 2号服务器: 2号是一个慢速服务器，可（以诊断服务数据解释速率）定时解释已接收的请求。在最糟糕的情况下，最后一次检查是否有传入请求消息也是发生在传输 / 网络层接收功能定址请求消息之前。这意味着该请求将存储在缓冲区，下次计划程序检查是否有传入请求时将最早处理该请求。2号服务器处理请求时，该服务器决定其不需要回应，因为其不支持所请求的信息（例如，服务器不发送任何响应消息，因为功能定址请求消息的情况下通常会取消否定响应代码requestOutOfRange（请求超过限值））。这时，2号服务器停止P2Server计时器并启动其S3Server计时器。

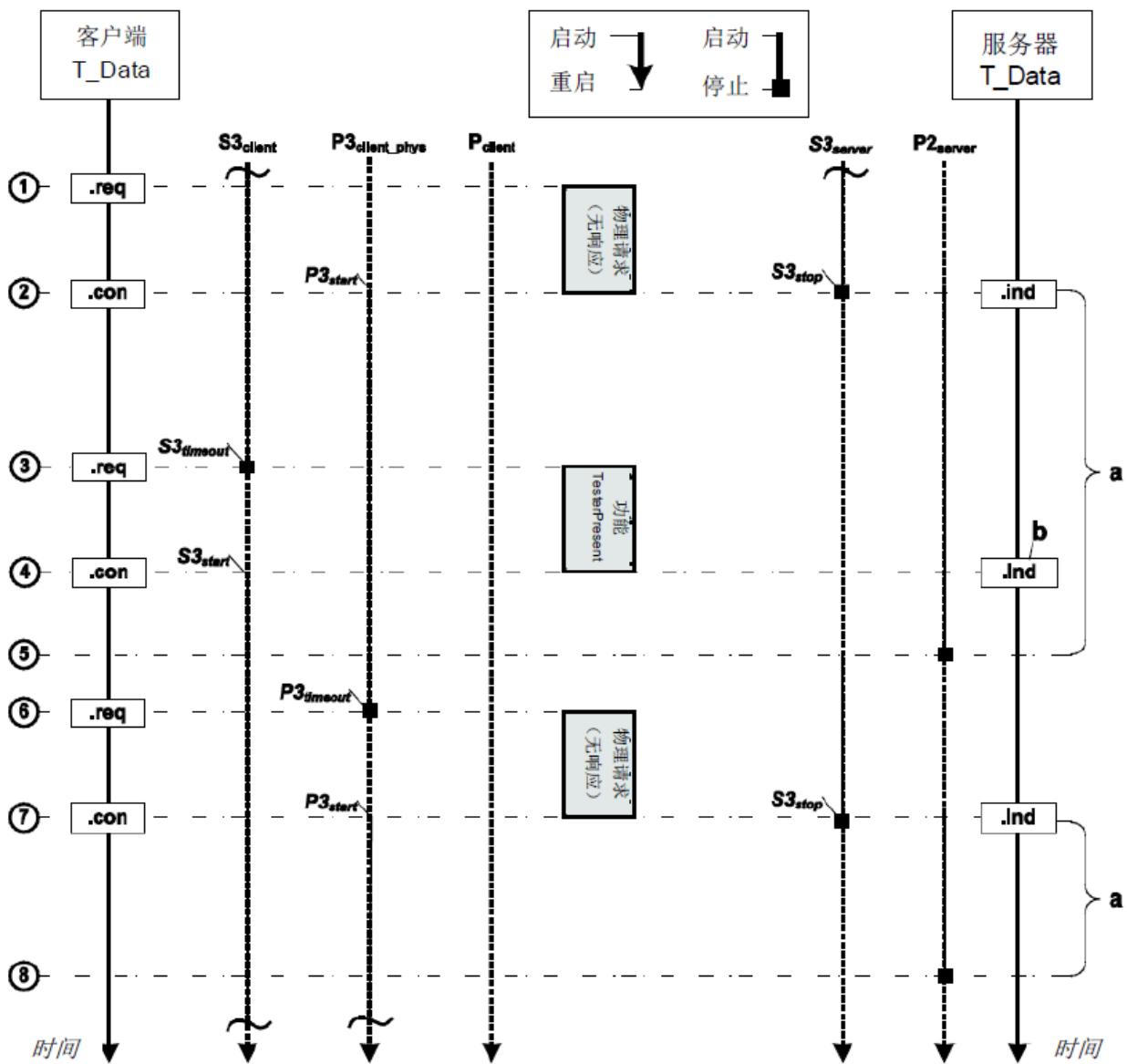
13 客户端T_Data.req: P3Client_Func计时器超时时，客户端可以通过T_Data.req来传输功能定址TesterPresent（测试仪保活）(0x3E)请求。

14 客户端T_Data.con: 表明已通过其传输 / 网络层的T_Data.con完成TesterPresent（测试仪保活）(0x3E)请求消息的传输后，客户端再次启动其S3Client计时器。这意味着每次S3Client超时都会定时发送功能定址TesterPresent（测试仪保活）(0x3E)请求消息。客户端通过P2Client=P2Client_max默认重载值（如果支持T_DataSOM.ind）或P6Client=P6Client_max以及其计时器P3Client_Func启动其PClient计时器。

所有服务器T_Data.ind: 传输 / 网络层向诊断应用程序发送请求消息的完成信息。S3Server计时器激活期间接收到任何TesterPresent（测试仪保活）(0x3E)请求消息都将重新启动S3Server计时器。

图19 功能定址请求消息的最短时间间隔(P3Client_Func)

图19为客户端的P3Client_Phys定时处理。显示了客户端S3Client超时时无需响应的物理定址请求的处理，以及功能定址TesterPresent（测试仪保活）(0x3E)请求消息的处理。



图例

a 诊断服务数据的解释率

b 服务器可忽略禁用S3Server计时器时接收的任一TesterPresent（测试仪保活）。

1 客户端T_Data.req: 诊断应用程序向无需响应的传输/网络层发送请求消息。

2 客户端T_Data.con: 传输/网络层向诊断应用程序发送完成确认消息。客户端通过用默认重载值P3Client_Phys=P3Client_Phys_max启动P3Client_Phys计时器。由于没有响应需要传输，因此客户端无需启动PClient计时器。

服务器T_Data.ind: 传输/网络层向诊断应用程序发送请求消息的完成信息。服务器用默认值P2Server=P2Server_max启动P2Server计时器，并在非默认会话中停用S3Server计时器。

3 客户端T_Data.req: 客户端的S3Client计时器超时时会使客户端传输功能定址TesterPresent（测试仪保活）(0x3E) 请求消息，无需来自寻址服务器的响应消息。假定此时未及时启用P3Client_Func计时器，则表示立即传输请求。

4 客户端T_Data.con: 通过传输/网络层的T_Data.con完成TesterPresent（测试仪保活）(0x3E) 请求消息的传输时，客户端会立即再次启动S3Client计时器。这表示在每次S3Client超时时会定期发送功能定址TesterPresent（测试仪保活）(0x3E) 请求消息。

服务器T_Data.ind: 服务器可忽略在处理其他请求消息时接收的任一TesterPresent (0x3E) 请求消息，因为此时已停用其S3Server计时器，并在处理完毕正在运行的服务后会再次将其启动。

5 服务器: 服务器会定期解释(诊断服务数据的解释率)接收的请求。在下一次调度程序检查传入请求时处理请求。执行完毕的服务会在非默认会话中重启S3Server计时器并停用P2Server计时器。

6 客户端T_Data.req: 客户端的P3Client_Phys计时器超时时，客户端会通过向其传输/网络层发送T_Data.req传输下一个物理定址请求消息。

7 客户端T_Data.con：传输/网络层向诊断应用程序发送完成请求的消息。客户端此时会再次启动其P3Client_Phys计时器。由于没有响应需要传输，因此客户端无需启动其PClient计时器。

服务器T_Data.ind：传输/网络层向诊断应用程序发送完成请求的消息。服务器用默认值P2Server=P2Server_max启动P2Server计时器，并在非默认会话中停用S3Server计时器。

8 服务器：服务器会定期解释（诊断服务数据的解释率）接收的请求。在下一次调度程序检查传入请求时处理请求。执行完毕的服务会在非默认会话中重启S3Server计时器并停用P2Server计时器。

图20 (P3Client_Phys)物理定址请求消息间的最短时间

7 UDSonCAN

7.1 文件概述

图1说明了实施UDSonCAN所需的文件。

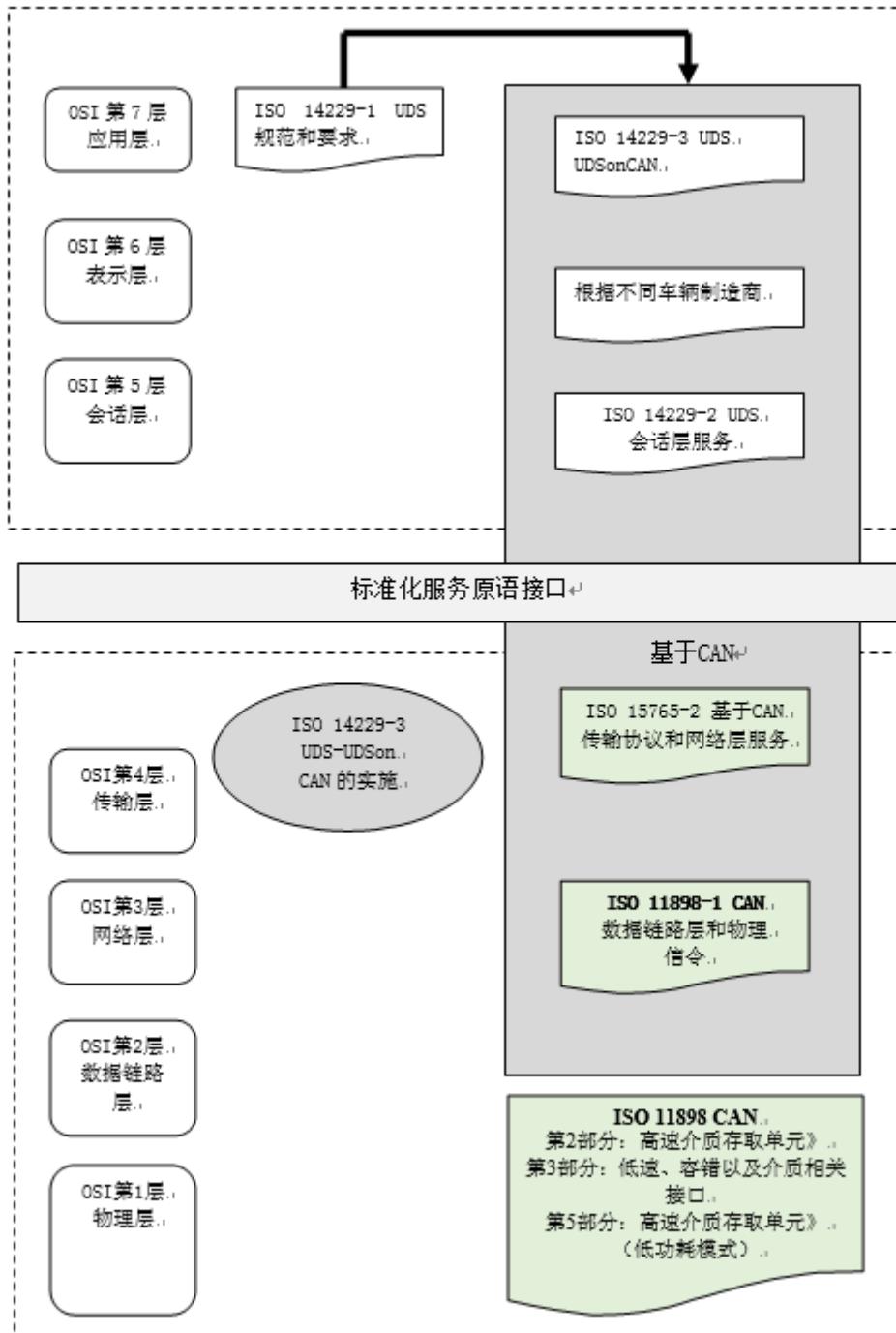


图21 UDSonCAN 文件引用情况与 OSI 模型对照

7.2 基于 CAN 实施统一诊断服务

7.2.1 概述

本条说明了如何对基于CAN实施如第5章定义的诊断服务。针对各种适用服务定义了适用的子功能和数据参数。

注：子功能参数的定义中，假定最高有效位用于第5章中定义的suppressPosRspMsgIndicationBit（抑制肯定响应消息指示位）参数。

7.2.2 对 CAN 实施的 UDS 的服务综述

表2旨在参考所有的第5章和第6章服务，因为这些服务均可应用在本章所述的UDSonCAN上。表2包含了所有适用的服务。使用本章实施UDSonCAN的某些应用程序会限制可用服务数量并在某些应用领域/诊断会话（默认会话、编程会话等）中将上述服务进行归类。

表2中凡标记为“CAN对此无特殊要求”的服务应按照第5章和第6章中的定义实施，不加以任何额外限制。凡标记为“CAN对此有特殊要求”的服务，应按照表2列出的对应子款中的定义进行实施。

表 2——适用的第 5 章统一诊断服务和数据范围概览

诊断服务名称（第 5 章）	备注	本文档引用情况
诊断和通信管理功能单元		
DiagnosticSessionControl（诊断会话控制）	CAN 对此无特殊要求	—
ECUReset（ECU 复位）	CAN 对此无特殊要求	—
SecurityAccess（安全访问）	CAN 对此无特殊要求	—
CommunicationControl（通信控制）	CAN 对此无特殊要求	—
TesterPresent（测试仪在线）	CAN 对此无特殊要求~	—
SecuredData-Transmission（安全数据传输）	CAN 对此无特殊要求~	—
ControlDTCSetting（诊断故障代码设置控制）	CAN 对此无特殊要求~	—
ResponseOnEvent（事件响应）	CAN 对此有特殊要求	见 6.3
LinkControl（链路控制）	CAN 对此无特殊要求~	—
数据传输功能单元		
ReadDataByIdentifier（通过标识符读数据）	CAN 对此无特殊要求	—
ReadMemoryByAddress（通过地址读内存）	CAN 对此无特殊要求	—
ReadScalingDataByIdentifier（通过标识符读缩放数据）	CAN 对此无特殊要求~	—
ReadDataByPeriodicIdentifier（通过周期标识符读数据）	CAN 对此有特殊要求	见 6.4
DynamicallyDefineDataIdentifier（动态定义数据标志符）	CAN 对此无特殊要求	—
WriteDataByIdentifier（通过标识符写数据）	CAN 对此无特殊要求	—
WriteMemoryByAddress（通过地址写内存）	CAN 对此无特殊要求~	—
ReadDTCInformation（读取诊断故障代码信息）	CAN 对此无特殊要求~	—
ClearDiagnosticInformation（清除故障信息）	CAN 对此无特殊要求	—
数据传输功能单元		
ReadDTCInformation（读取诊断故障代码信息）	CAN 对此无特殊要求	—
ClearDiagnosticInformation（清除故障信息）	CAN 对此无特殊要求	—
输入/输出控制功能单元		
InputOutputControlByIdentifier（通过标识符进行输入输出控制）	CAN 对此无特殊要求	—
常规功能单元的远程激活		
RoutineControl（例程控制）	CAN 对此无特殊要求	—
上传/下载功能单元		
RequestDownload（请求下载）	CAN 对此无特殊要求	—
RequestUpload（请求上传）	CAN 对此无特殊要求	—
TransferData（传输数据）	CAN 对此无特殊要求	—
RequestTransferExit（请求退出传输）	CAN 对此无特殊要求	—
RequestFileTransfer（请求文件传输）	CAN 对此无特殊要求	—

7.2.3 ResponseOnEvent（事件响应）（0x86）服务基于 CAN 实施的要求

除了第5章规定的一般实施要求外，还需适用以下附加的CAN具体实施要求。

当ResponseOnEvent（事件响应）服务激活时，服务器应能够处理并发的诊断请求和相应的响应消息。这应通过一组不同的serviceToRespondTo（待响应的服务）CAN标识符完成。若相同的诊断请求/响应CAN标识符用于诊断通信和服务ToRespondTo-response（待响应的服务-响应）中，则以下限制应适用：

- a) 在一个事件发生之后，服务器会忽略即将到来的由启动事件的客户端发出的诊断请求，并且开始处理 serviceToRespondTo-response（待响应的服务-响应），直至 serviceToRespondTo-response（待响应的服务-响应）完成。若另一个客户端发出了请求，则服务器会根据其能力选择忽略或不忽略该请求。
- b) 若客户端在发出诊断请求后收到响应，则该响应应当根据可能的 serviceToRespondTo-response（待响应的服务-响应）和先前已发请求的预期诊断响应进行分类。

- 1) 若该响应是一个 serviceToRespondTo-response 响应服务-响应（由 ResponseOnEvent-service 事件响应-服务设定的可能的响应之一），客户端应在完全收到 serviceToRespondTo-response 响应服务-响应后重复该请求。
- 2) 若该响应不确定【即，该响应可能产生于一个事件启动的 serviceToRespondTo 响应服务也可能产生于一个诊断请求的响应】，客户端应将该响应同时作为一个 serviceToRespondTo-response （待响应的服务-响应）和一个诊断请求的响应。此时，客户端一般不得重复该请求，除非收到的是 NegativeResponseCode 否定响应代码当中的 busyRepeatRequest 重复请求忙（0x21）（见第 5 章中否定响应码的定义）。

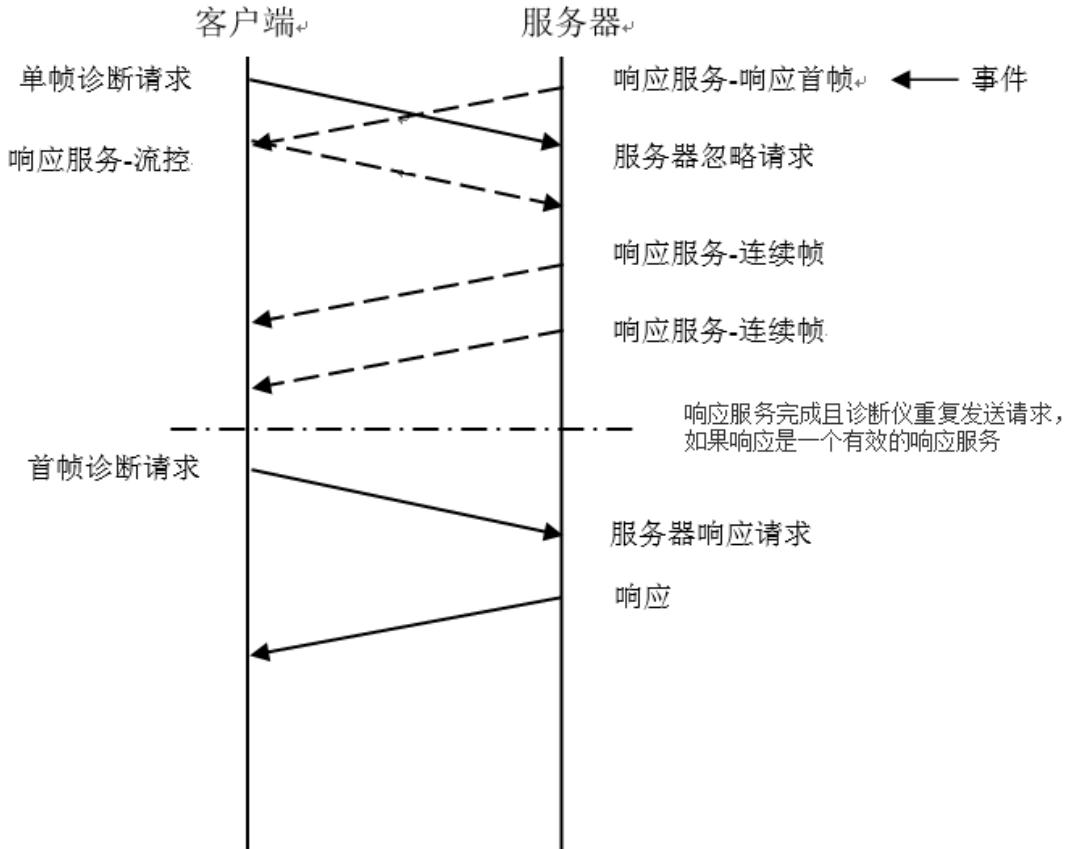


图22 事件发生时的并发请求

- c) 当检测到指定的事件时，服务器应立即使用适当的响应服务-响应消息做出响应。上述响应服务-响应消息不得打断其它任何正在进行的诊断请求或响应的传输【即，应当延迟响应服务-响应，直至当前的消息传输结束，见下图】。

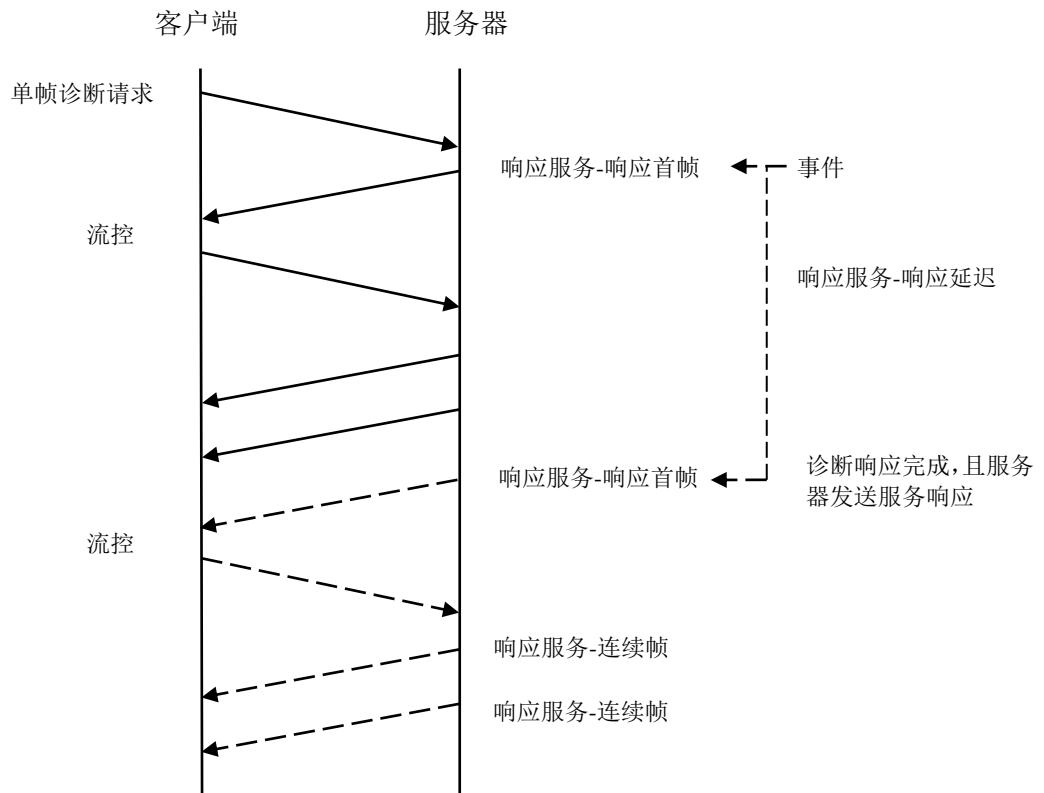


图23 在消息处理过程中发生的事件

7.2.4 ReadDataByPeriodicIdentifier（通过周期性标识符读数据）（0x2A）服务

7.2.4.1 周期性传输响应消息

ReadDataByPeriodicIdentifier（通过周期性标识符读数据）服务允许客户端请求从由一个或多个 periodicDataIdentifier（周期性数据标识符）标识的服务器周期性传输数据记录值。根据第5章规定，对于在 CAN中实施的服务0x2A，周期性传输响应消息是受支持的。

使用周期性传输响应消息时应考虑到整个车辆是否支持同一种数据格式，车辆可能包括除CAN外的其它数据链路。例如，如果除了CAN以外，FlexRay是电动汽车结构的一部分，则总DID数据长度不得超过CAN协议的长度限制，以确保单一的数据格式。

表4介绍了周期性传输响应消息。

表6 周期性传输——响应消息映射要求

消息类型	客户端请求要求	服务器响应要求	更多服务器限制
UUDT 使用一个不同的 CAN 标识符进行周期性传输	没有限制	对于周期性传输只存在单帧响应对于新（非周期性传输）请求可能有多帧响应	周期性传输请求会被作为一个常规诊断请求进行处理并通过网络层发送响应（作为一个以 0x6A 为服务标识符的 USDT 消息）。 在收到表示肯定响应已经传输完成的 N_USData.con 时，应用程序会开启一个独立调度器处理周期性传输。 服务器中的调度器会将周期性传输作为一个旁路中的 CAN 单帧响应消息处理（即，直接将消息写给 CAN 控制器/数据链路层驱动器驱动程序，而无需使用网络层）。 响应消息中既不包含协议控制信息（PCI），也不包含服务标识符（SId）。仅包含周期性标识符和相应的数据。

7.2.4.2 周期性传输响应消息的处理

7.2.4.2.1 概述

由于周期性响应消息既不支持协议控制信息，也不支持服务标识符信息（传输层旁路），所以需考虑以下服务原语。如下定义的服务原语使用了ISO 15765-2中规定的单个参数。

7.2.4.2.2 N_UUData.request

服务原语请求从发送器向接收器中根据N_SA和N_TA中的地址信息确定的对等实体周期性传输长度为<Length>字节的<MessageData>消息数据。每次调用N_UUData.request服务时，传输/网络层应通过发出一个N_UUData.confirm服务调用请求的方式向服务器用户发出信息传输已经完成（或失败）的信号：

N_UUData.request(

```
    N_SA,  
    N_TA,  
    <Length>,  
    <MessageData>  
)
```

7.2.4.2.3 N_UUData.confirm

N_UUData.confirm服务由传输/网络层发出。此服务原语用于确认由N_SA和N_TA中的地址信息标识的N_UUData.request服务是否已经完成。参数<N_Result>指出了服务请求的状态(关于<N_Result>取值的定义，请参见ISO15765-2)。

N_UUData.confirm(

```
    N_SA,  
    N_TA,  
    <N_Result>  
)
```

注：ISO 15765-2中定义的结果值并非全部适用于响应消息传输类型UUDT。

7.2.4.2.4 N_UUData.indication

N_UUData.indication服务由传输/网络层发出。服务原语将从根据N_SA和N_TA中的地址信息确定的对等协议实体处收到的长度为<Length>字节的<MessageData>传递给上一层(见ISO15765-2中的定义)。

参数<MessageData>和<Length>仅在已指示服务原语时有效。若出现接收错误，则不会生成任何指示。因此将不需要<N_Result>参数。

N_UUData.indication(

```
    N_SA,  
    N_TA,  
    <Length>,  
    <MessageData>  
)
```

7.2.4.3 CAN 帧格式

7.2.4.3.1 概述

由于周期性传输响应消息既不支持PCI信息，也不支持服务标识信息，所以协议数据单元(PDU)格式与ISO 15765-2中定义的格式不同。但是，CAN周期性传输PDU格式中包含了表4中指定的三个字段。

表7 周期性传输——PDU 格式

地址信息	长度信息	数据字段
CAN 标识符 C_AI	DLC	<MessageData>

7.2.4.3.2 地址信息

C_AI用于识别周期性消息PDU的通信对等实体。网络层数据的交换受常规寻址的支持。地址信息映射到CAN标识符上，见ISO15765-2。

7.2.4.3.3 长度信息

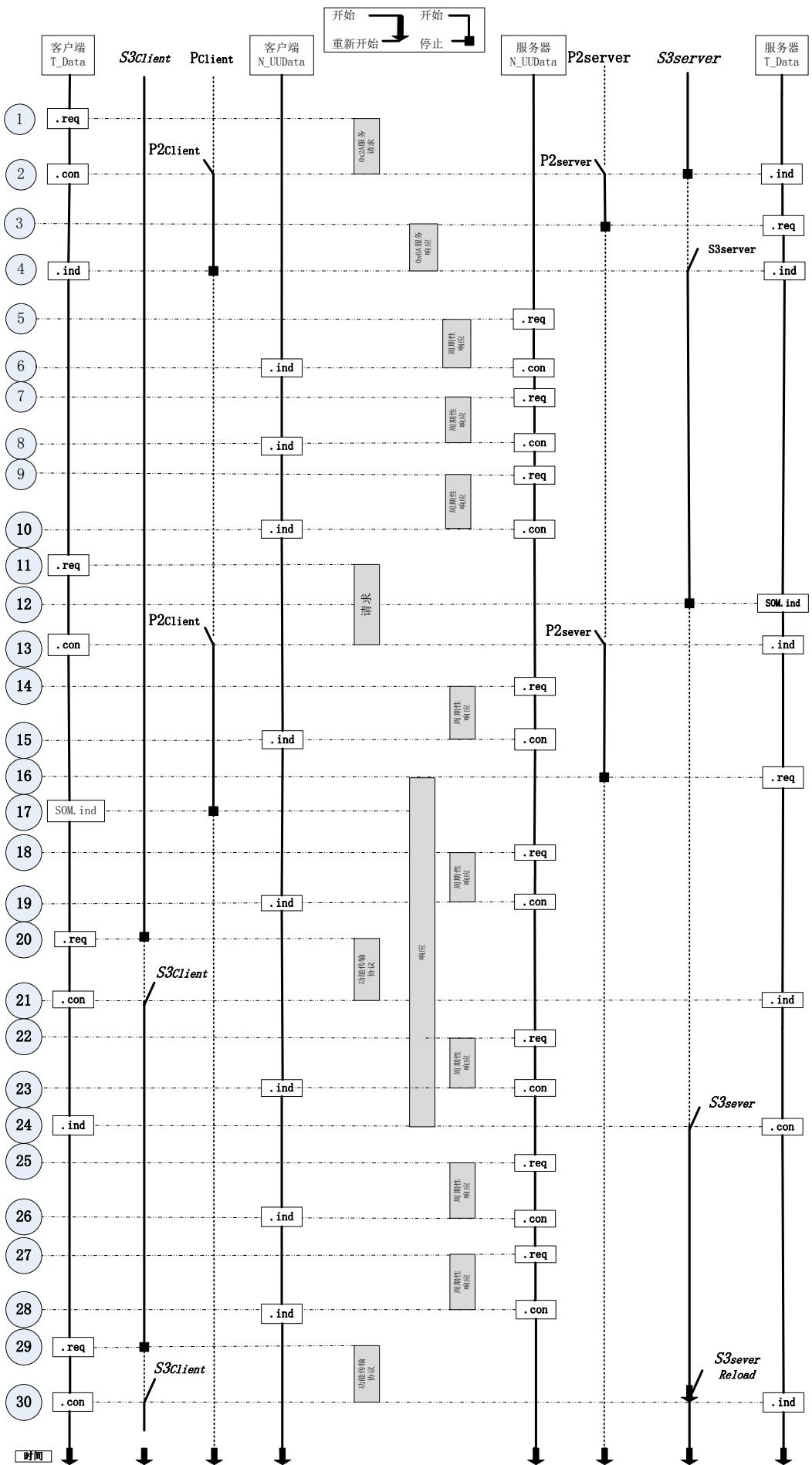
DLC参数说明了CAN帧传输的数据字节数量。

7.2.4.3.4 数据场

<MessageData>字段包含了用一个periodicDataIdentifier（周期性数据标识符）（PDID）和其相应dataRecord（数据记录）定义的周期性数据信息。<MessageData>字段的最大长度为八个字节。

7.2.4.4 周期性传输消息流

图4以图形方式说明了上述周期性响应消息的处理。正如7.2.4.2节已经提到，该图说明了周期性传输的响应消息对服务器的S3sever计时器没有任何影响。该图中假定在配置周期性调度器之前有一个non-defaultSession（非默认会话）已激活【ReadDataByPeriodicIdentifier（通过周期性标识符读取数据）服务需要一个non-defaultSession（非默认会话）方可执行】。



图例

1 客户端 T_Data.req:客户端诊断应用程序通过向其传输/网络层发出一个 T_Data.req 开始传输 ReadDataByPeriodicIdentifier (按周期性标识符读取数据) (0x2A) 请求消息。传输/网络层将 ReadDataByPeriodicIdentifier (通过定期性标识符读取数据) (0x2A) 请求消息传输给服务器。该请求消息可以是一个分段或不分段的消息【取决于请求消息中包含的 periodicDataIdentifier (周期性数据标识符) 数量】。上例中假定了该请求消息适合于一个单帧。

2 客户端 T_Data.con: 在客户端, 通过 T_Data.con 表示请求消息已经完成。此时, 第 6 章所述的响应计时适用。服务器 T_Data.ind:在服务器中, 通过 T_Data.ind 表示请求消息已经完成。此时, 第 6 章所述的响应计时适用。此外, 服务器停止其 S3Server 计时器。

3 服务器 T_Data.req:假定客户端需要来自服务器的响应。服务器应传输 ReadDataByPeriodicIdentifier (通过周期性标识符读取数据) 肯定响应消息, 以显示请求已处理完成。

4 服务器 T_Data.con:在服务器端, 通过 T_Data.con 表示 ReadDataByPeriodicIdentifier (按周期性标识符读取数据) 响应消息的传输完成。此时, 服务器重新启动其 S3server 计时器, 该计时器会使激活的非默认会话保持激活状态, 直至超时为止。现在会开始传输周期性消息。

客户端 T_Data.ind:在客户端使用此服务表示响应消息已经接收。

5 服务器 N_UUData.req:服务器开始传输周期性响应消息(基于 CAN 单帧消息)。每个周期性消息即不包含 PCI 信息, 也不包含服务标识, 所使用的源地址与各 USDT 响应消息均不相同。因此, 每当发送周期性消息时, 无论服务器当时正在处理什么其它信息, 均会独立发出一条 N_UUData.req。这意味着, 即使服务器正在处理另一诊断服务请求, 周期性响应消息的传输仍将继续。周期性响应消息的传输不会对 S3server 计时器产生任何影响。

6 服务器 N_UUData.con:在服务器中使用此服务表示周期性响应消息的传输已经完成。

客户端 N_UUData.ind:在客户端使用此服务表示周期性响应消息的接收已经完成。

7 见 (5)

8 见 (6)

9 见 (5)

10 见 (6)

11 客户端 T_Data.req:客户端诊断应用程序通过向其传输/网络层发出一个 T_Data.req 开始传输下一条请求消息。传输/网络层会将该请求消息传输给服务器。该请求消息可以是一条不分段或分段的消息。在上图的例子中, 假设请求消息是一条多帧消息。

12 服务器 T_Data_SOM.ind:在服务器中, 在周期性调度器活动期间, 使用 T_Data_SOM.ind 表示请求消息的开始。在处理接收到的请求消息期间, 服务器并不会停止周期性调度。这意味着服务器在处理诊断服务期间会继续发送更多的周期性消息。客户端应注意接收这些周期性的响应消息。此外, 服务器在处理任何诊断服务期间均会停止其 S3server 计时器。

13 客户端 T_Data.con:在客户端, 通过 T_Data.con 表示请求消息已经完成。现在, 第 6 章所述响应计时适用。

服务器 T_Data.ind: 多帧请求信息的完成会通过 T_Data.ind.显示在服务器中。此时, 第 6 章所述的响应计时适用。

14 见(5)

15 见(6)

16 服务器 T_Data.req: 假定客户端需要来自服务器的响应。服务器应通向其传输/网络层发出一个 T_Data.req 开始传输一条肯定(或否定)响应消息。此例中假定了该响应是一条多帧消息。当通过传输/网络层传输多帧响应消息时, 周期性调度器会继续传输周期性响应消息。

17 客户端 T_Data_SOM.ind: 响应信息的开始会显示在客户端中。

18 见(5)

19 见(6)

20 客户端 T_Data.req: 当客户端的 S3client 计时器超时, 客户端会传输一条功能寻址的 TesterPresent (测试仪在线)(0x3E)请求消息, 以重新启动服务器中的 S3Server 计时器。

21 服务器 T_Data.ind: 服务器正在传输先前请求的多帧响应消息。因此, 服务器不对接收到的 TesterPresent(0x3E) 请求消息做出任何反应, 因为其 S3server 计时器尚未被重新激活。

客户端 T_Data.con: 在客户端, 使用 T_Data.con 表示 TesterPresent(测试仪在线)(0x3E)请求信息已经接收成功。

22 见(5)

23 见(6)

24 服务器 T_Data.con:当诊断服务处理完成后, 服务器重新启动其 S3server 计时器。这意味着包括 TesterPresent(测试仪在线)(0x3E)在内的任何诊断服务均会重新启动 S3se 「ver 计时器。诊断服务在进行之中, 是指: 从开始接收请求消息(T_Data_SOM.ind 或 T_Data.ind 接收)至响应消息传输完成(在需要响应消息的情况下)或至该请求所导致的行动已经完成(在不需要响应的情况下)(在这个时间点本来会导致响应消息的开始)之间的任意时间点。这包括如响应代码 0x78 等否定响应消息。

客户端 T_Data.ind: 在客户端使用此服务表示响应消息已经接收。

25 见(5)

26 见(6)

27 见(5)

28 见(6)

29 客户端 T_Data.req: 每当 S3c”ent 计时器在客户端中超时时【在 non-defaultSession(非默认会话)激活的情况下】，客户端就会传输一条功能定址 TesterPresent(测试仪在线)(0x3E)请求消息(不需要响应消息)。

30 客户端 T_Data.con: 在传输/网络层的 T_Data.con 表示 TesterPresent(测试仪在线)(0x3E)请求消息传输完成时，客户端会重新启动其 S3Client 计时器。这意味着每次 S3Client 计时器超时均将发送功能定址的 TesterPresent(0x3E)(测试仪在线)请求消息，具有周期性质。

服务器 T_Data.ind: TssterPresent(测试仪在线)(0x3E)请求信息的接收显示在服务器中。服务器应重新激活 S3server 计时器。

图24 周期性传输响应消息的处理

7.3 应用层要求

7.3.1 应用层服务

本章针对基于客户端-服务器的系统使用如第5章定义的应用层服务来执行车载车辆服务器的诸如测试、检查、监测、诊断或编程等功能。

7.3.2 应用层协议

本章使用如第5章所定义的应用层协议。

7.3.3 应用层定时

7.3.3.1 概述

以下分条款说明了消息定时参数和其在客户端和服务器上的应用方式。

7.3.3.2 消息定时参数值

应用层定时参数取值应按照第6章的定义。

7.3.3.3 非请求响应消息

非请求消息是指服务器根据一个周期性调度器【见6.4中的服务ReadDataByPeriodicIdentifier（根据周期性标识符读取数据）】或一个已配置触发事件，如诊断故障代码状态变化或dataIdentifier（数据标识符）值变化【见6.3中的服务ResponseOnEvent（事件响应）】而传输的消息。

非请求传输的响应消息不得重置服务器中的S3Server计时器。不然，当某个周期性消息传输已激活，或已在服务器中配置某个计时器触发事件，并且事件间的时间间隔小于S3Server时，服务器中会出现诊断会话锁存。只有在传输的响应消息是处理一个请求消息和传输最终响应消息【如，表示一个或多个periodicDataIdentifier（周期性数据标识符）的安排请求已成功完成的初始肯定响应】的直接结果时，才允许重置S3server计时器。

7.4 表示层要求

表示层要求由车辆制造商负责。

7.5 会话层要求

会话层要求见第6章。

7.6 传输/网络层接口适配

7.6.1 一般信息

本章使用了第6章中定义的网络层服务来传输和接收诊断消息。本分条款定义了从独立于数据链路的传输/网络层协议数据单元（T_PDU）到CAN数据链路特定网络层协议数据单元（N_PDU）的映射。

注：使用传输/网络层服务进行应用层和诊断会话管理定时。

7.6.2 基于 CAN 传输/网络层接口适配

7.6.2.1 从独立于数据链路的服务原语到 CAN 数据链路特定服务原语的映射

表5显示了第6章中的DoCAN传输协议和网络层服务与第6章中针对诊断消息的发送和接收而定义的会话层服务之间的映射接口。

表8 从 T_PDU 服务原语到 N_PDU 服务原语的映射

会话层到传输层服务原语 (根据第 6 章, 独立于数据链路层)	基于 CAN 网络层服务原语 (根据 ISO 15765-2, 与数据链路层相关)
TData.indication	NUSData.indication
TDataSOM.indication	NUSDataFF.indication
TData.confirm	NUSData.confirm
T_Data.request	N_USData.request

7.6.2.2 消息传输用 T_PDU 到 N_PDU 的映射

为请求传输诊断服务请求/响应而定义的应用层协议数据单元, 其参数与客户端/服务器中用于消息传输的传输层/网络层通信协议数据单元的参数的映射关系如表 6 所示。

表9 从 T_PDU 参数到 N_PDU 参数的映射

T_PDU 参数 (根据第 6 章, 独立于数据链路层)	N_PDU 参数 (根据第 6 章, 与 CAN 数据链路相关)
T_Mtype	N_Mtype
T_SA	N_SA
T_TA	N_TA
TTAtype	NTAtype
T_AE	N_AE
若 Mtype=诊断, 则地址信息应包括参数 SA、TA 以及 TAtype。 若 Mtype=远程诊断, 则地址信息应包括参数 SA、 TA、TAtype 以及 AE。	
T_Data[]	<MessageData>
T_Length	<Length>
TResult	<NResult>

网络层确认消息发送成功 (C_Data.con) 时要将此确认转发给应用程序, 因为应用程序收到该确认才能启动那些应在成功传输请求/响应消息【ECUReset (ECU复位)、BaudrateChange (波特率改变) 等】之后应立即执行的操作。

网络层和OSI更高层之间的地址映射并不一定是数据链路层上经过编码的地址值的精确副本, 而是取决于具体实施概念。

映射的定义见第6章。排放相关OBD的值的定义见ISO15765-4。

7.7 数据链路层诊断实施要求

7.7.1 使用法定 11 位 OBD CAN 标识符进行增强型诊断

法定OBD的11位CAN标识符也可用于增强型诊断 (例如, 功能请求CAN标识符可用于功能寻址 TesterPresent (测试仪在线) (0x3E) 请求消息, 以使一个non-defaultSession (非默认会话) 保持激活状态)。

若11位CAN标识符被用于ISO 15765-4中规定的增强型诊断, 则应适用以下要求:

- a) 符合 ISO15765-4 的网络层计时参数也应适用于增强型诊断
- b) DLC (CAN 数据长度码) 应设为八, 而 CAN 帧应包括八个字节 (应填充未用字节)。

7.7.2 使用法定 29 位 OBD CAN 标识符进行增强型诊断

排放相关法定OBD服务器的29位CAN标识符符合ISO15765-2中规定的常规固定寻址模式并也可用于强化诊断。

若ISO 15765-4中规定的29位CAN标识符被重复用于增强型诊断, 则应适用以下要求:

- a) ISO 15765-4 中规定的网络层计时参数也应适用于增强型诊断;
- b) DLC 应设为八, 而 CAN 帧应包括八个字节 (应填充未用字节)。

注: ISO 15765-2中给出的CAN标识符值针对优先级信息使用默认值。

8 UDSonFR

8.1 文件概述

图24显示了第5章、第6章、ISO 10681-1、ISO 10681-2的文档引用情况。本章仅用到第5章中定义的诊断服务的一个子集 (见表2)。

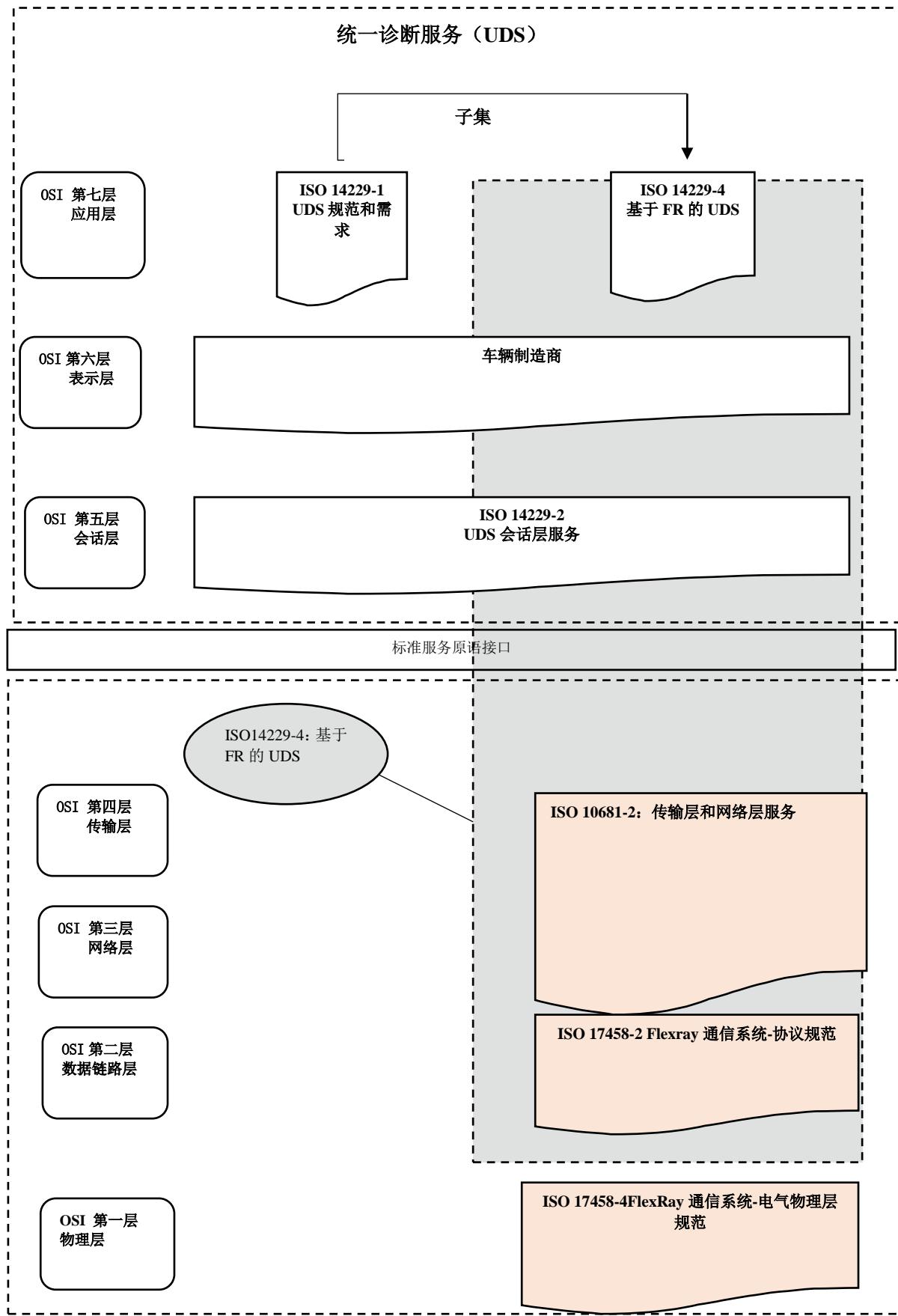


图25 UDSonFR 文档引用情况与 OSI 模型的对照

8.2 基于 FlexRay 的统一诊断服务

8.2.1 概述

本章节说明了如何在FlexRay总线上实施第5章所定义的诊断服务。针对各个服务，定义了适用的子功能和数据参数。

注：按照第5章定义，在子功能的参数定义中，最高有效位是肯定响应抑制指示位。

8.2.2 基于FlexRay的UDS服务概述

表2列出了第5章和第6章定义的所有适用于本章的服务。使用此部分UDS on FR的某些应用程序，可能会限制可用服务的数量并在某些应用领域/诊断会话（默认会话、编程会话等）中将上述服务进行归类。

表2中凡标记“FlexRay对此无特殊要求”的服务，应按照第5章和第6章的定义进行实施，不加以任何额外限制。凡标记为“FlexRay对此有特殊要求”的服务，应按表2列出的相应章节的定义进行实施。

表2——适用的第5章统一诊断服务和数据范围概览

诊断服务名(第5章)	备注	本文档引用情况
诊断和通信管理功能单元		
诊断会话控制	FlexRay对此无特殊要求	
ECU复位	FlexRay对此无特殊要求	
安全访问	FlexRay对此无特殊要求	
通讯控制	FlexRay对此有特殊要求	见6.3
测试仪保活	FlexRay对此无特殊要求	
安全数据传输	FlexRay对此无特殊要求	
诊断故障码控制设置	FlexRay对此无特殊要求	
响应事件	FlexRay对此有特殊要求	见6.4
链路控制	FlexRay对此有特殊要求	见6.5
数据传输功能单元		
通过标识符读取数据	FlexRay对此无特殊要求	
通过地址读取内存	FlexRay对此无特殊要求	
通过标识符读取消放数据	FlexRay对此无特殊要求	
通过周期性标识符读取数据	FlexRay对此有无特殊要求	见6.6
动态定义数据标识符	FlexRay对此无特殊要求	
通过标识符写数据	FlexRay对此无特殊要求	
通过地址写内存	FlexRay对此无特殊要求	
读取诊断故障码信息	FlexRay对此无特殊要求	
清除诊断信息	FlexRay对此无特殊要求	
输入/输出控制功能单元		
输入输出控制	FlexRay对此无特殊要求	
常规功能单元的远程激活		
例程控制	FlexRay对此无特殊要求	
上传/下载功能单元		
请求下载	FlexRay对此无特殊要求	
请求上传	FlexRay对此无特殊要求	
传输数据	FlexRay对此无特殊要求	
请求传输退出	FlexRay对此无特殊要求	
请求文件传输	FlexRay对此无特殊要求	

8.2.3 CommunicationControl（通信控制）(0x28)服务

在FlexRay中实施该服务时，应遵循表3所述要求。

表3——FlexRay中0x28服务的子功能实施定义

十六进制(位6-0)	描述	CVT	助记符
00	启用Rx和Tx —打开FlexRay在静态段的通信； —打开FlexRay在动态段（如适用）的通信；	M	ERXTX
01	启用RX并禁用TX 启用在静态段和动态段的接收通信； 禁用在静态段和动态段的发送通信； —在静态段，各个单独的时隙仅发送空帧。 —在动态段（如适用），通信关闭（结果为仅有微时隙minislots）	M	ERXDTX
02	禁用Rx并启用Tx	M	DRXETX

	—禁用在静态段和动态段的接收通信; —启用在静态段和动态段的发送通信;		
03	禁用Rx和Tx 禁用在静态段和动态段的通信（接收和发送）。对静态段和动态段的接收和发送影响，与0x01和0x02子功能所述一致。	M	DRXTX
a需确保基本网络功能（例如，同步功能）在禁用发送通信时不受影响。			

8.2.4 ResponseOnEvent（基于事件响应）(0x86)服务

除了在第5章中规定的一般实施要求外，还需适用以下FlexRay附加的实施要求：

当ResponseOnEvent（基于事件响应）服务激活时，服务端应能够处理并发的诊断请求并回复相应的响应消息。这应使用与serviceToRespondTo（待服务端响应）不同的FlexRay源地址完成。如果用于诊断通信和用于serviceToRespondTo-responses（待服务端响应-响应）的FlexRay源地址和目标地址相同，则应适用以下限制：

- a) 发生事件后，在serviceToRespondTo-response（待服务端响应-响应）服务进行过程中，服务端应忽略收到的诊断请求，直到serviceToRespondTo-response（待服务端响应-响应）完成。
- b) 若客户端在发出诊断请求后收到响应，则应先判断该响应是serviceToRespondTo-response（待服务端响应-响应）的响应，还是先前发送的诊断请求响应。
 - 1) 若该响应是serviceToRespondTo-response（待服务端响应-响应）【由ResponseOnEvent-service（基于事件响应-服务）设定的可能的响应之一】，客户端应在完全收到serviceToRespondTo-response（待服务端响应-响应）后重复该请求。
 - 2) 若该响应不确定【即，该响应可能产生于一个事件启动的serviceToRespondTo（待服务端响应）也可能产生于一个诊断请求的响应】，客户端应将该响应同时作为serviceToRespondTo-response（待服务端响应-响应）和诊断请求的响应。此时，客户端一般不得重复发送请求，除非收到的否定响应代码0x21（忙碌请重复请求）。（否定响应代码的定义见第3章。）

a 所示示例中，假设事件所引起的STRT-ResponseSTF（待服务端响应-响应的起始帧）与来自客户端的诊断请求在同一FR周期内发出。

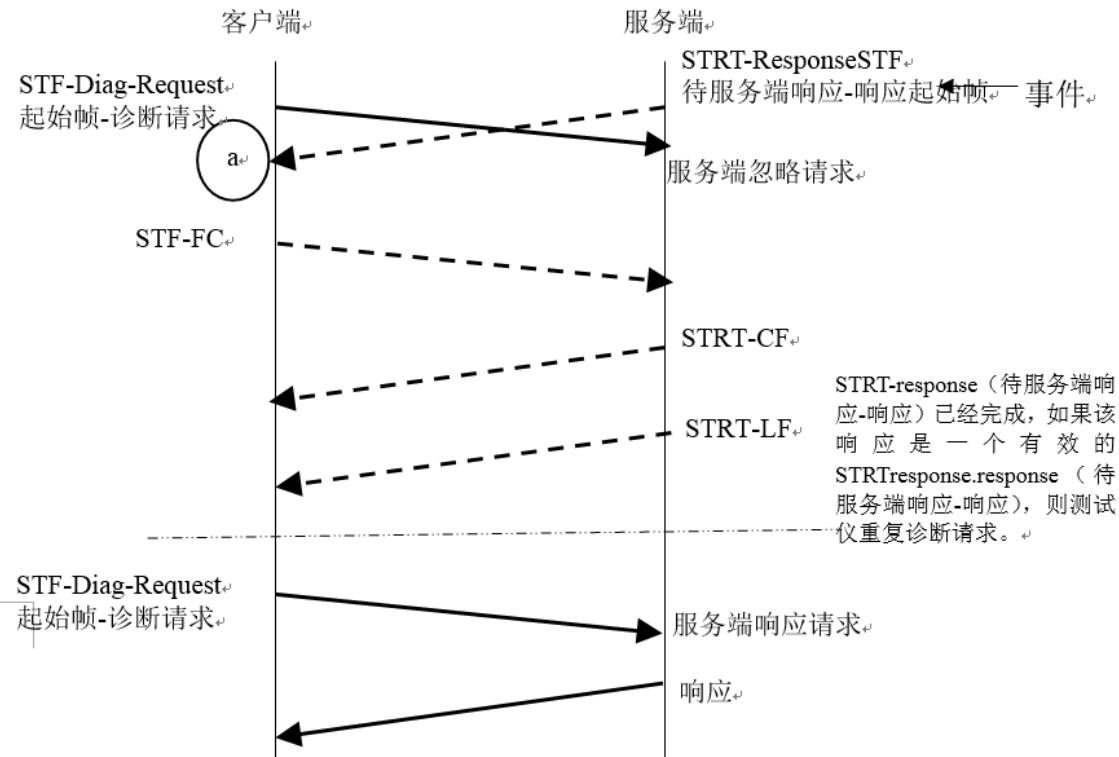


图2——事件触发时的并发请求

- c) 当检测到指定的事件时，服务端应立即使用适当的serviceToRespondTo-response（待服务端响应-响应）消息做出响应。上述serviceToRespondTo-response（待服务端响应-响应）消息不得打断其它任何正在进行的诊断请求或响应的传输【即，应当延迟serviceToRespondTo-response（待服务端响应-响应）直至当前的消息传输结束，见图3】。

注：如果事件发生时，服务端的上层显示尚未处于诊断请求进行状态，则服务端将发送serviceToRespondTo-response（待

服务端响应-响应），而忽略诊断请求（见图2）。

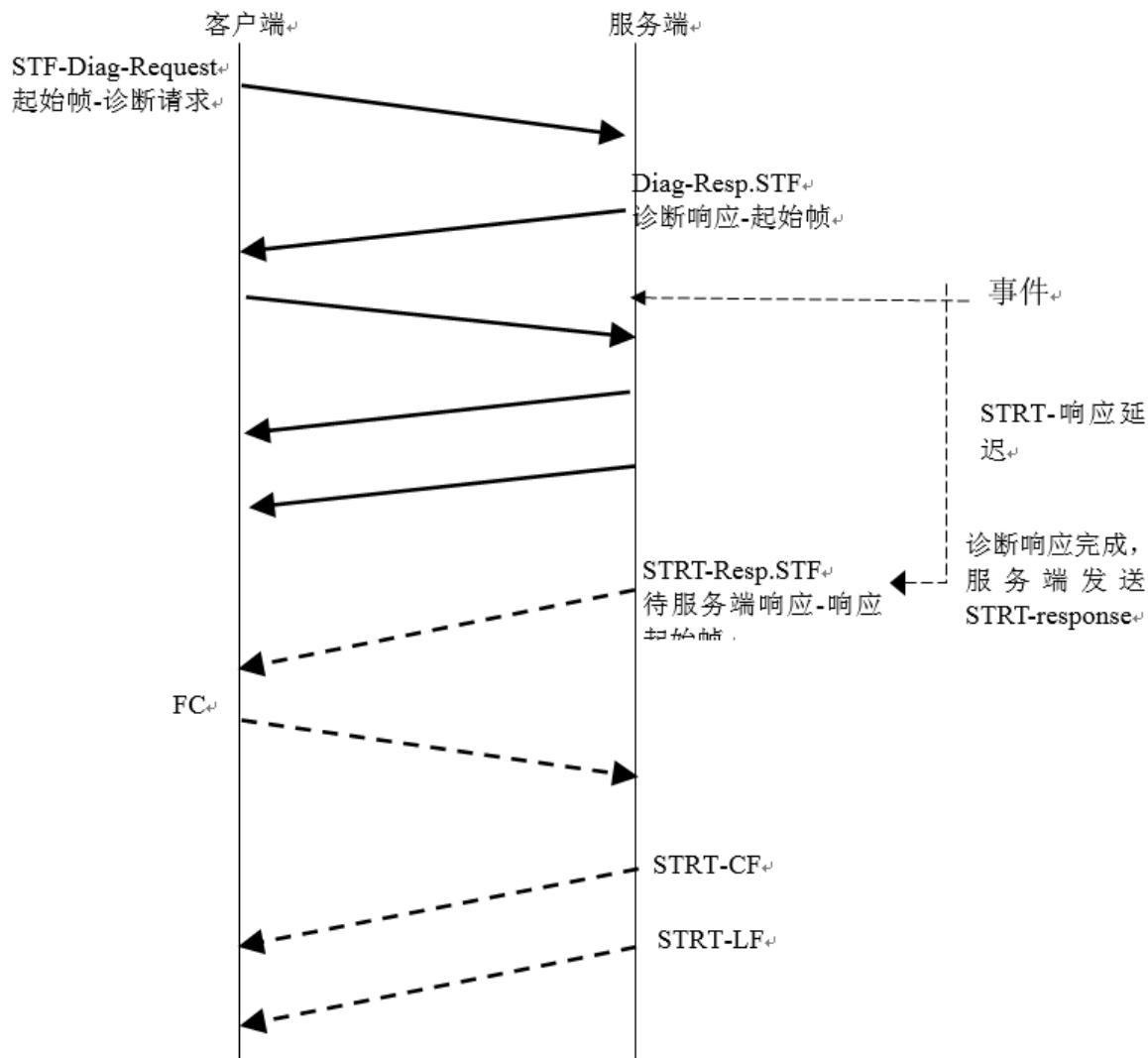


图 3——在消息处理过程中触发事件

8.2.5 LinkControl (链路控制) (0x87)服务

在FlexRay中，链路控制被用于调整数据链路相关参数，使诊断通信带宽利用率最大化（例如，软件刷写）。调整FlexRay调度表（例如，增加动态段，调整诊断消息的帧长度等）。然而，具体实施应在具体车辆制造商的实施规范中阐明。

在FlexRay中实施该服务时，应适用以下要求。通信策略遵循第5章要求：

——第1步：应遵循第5章中所述定义。

——第2步：按照各自的实施规范，客户端请求模式转换来调整FlexRay调度表。只有第1步已经成功完成时，才会请求此步骤。通过功能寻址进行模式转换时，建议将(suppressPosRspMsgIndicationBit=TRUE)肯定响应消息抑制标志位置位，在此情况下服务端执行模式切换后将不发送肯定响应。因为一台服务端已经转换到新的模式时，其它服务端可能仍处于转换进行中。

具体FlexRay实施应遵循表4要求。

表 4——请求消息的子功能参数定义

十六进制(位6-0)	描述	CVT	助记符
01	验证使用固定变量传输模式 该参数用于验证是否可以使用一个预定义的FlexRay调度设计进行模式转换。该调度设计是由linkControlModeIdentifier（链路控制模式标志符）数据参数指定。	M	ERTX
02	验证使用特定参数模式转 不适用于FlexRay	M	ERDTX
03	转换模式 该子功能参数请求服务端将FlexRay转换到前述验证消息所请求的调度设计	M	TM

8.2.6 (通过周期型标志符读取数据) (0x2A)服务

8.2.6.1 周期性传输响应消息

ReadDataByPeriodicIdentifier (通过周期性标识符读取数据) 服务允许客户端使用由一个或多个periodicDataIdentifier (周期性数据标识符) 标识的请求，来获取服务端的周期性数据记录值。第5章定义的2A服务响应消息格式，同样适用于FlexRay。

使用周期性响应传输消息时，应考虑整个车辆网络是否支持单一的数据格式。车辆可能存在除FlexRay外的其它数据链路，例如：除了FlexRay外，车辆架构还存在CAN通讯，则总的DID数据长度不得超过CAN协议的长度限制，以确保单一的数据格式。

表5描述了针对周期性传输响应消息的要求。

表5——周期性传输——响应消息映射要求

消息类型	客户端请求要求	服务端响应要求	更多服务端限制
周期性传输响应消息与周期性消息使用不同的FlexRay源地址传输	没有限制	对于周期性传输，仅可能出现FlexRay单帧响应 在周期性响应正在进行的过程中，可能同时存在多帧响应（非周期性传输）	周期性传输请求被作为常规诊断请求处理，通过通信层发送响应（使用0x6A服务标识符的FlexRay消息） 在收到表示肯定响应的发送已经完成的C_Data.confirm时，应用程序会开启一个独立的调度表来处理周期性传输。 服务端中的这个调度表将处理周期性传输FlexRay单帧响应消息，从by-pass旁路传到传输层 响应消息中既不包含协议控制信息(PCI)，也不包含服务标识符(SId)。仅包含周期性标识符和相应的数据。

8.2.6.2 周期性传输响应消息的处理

8.2.6.2.1 概述

由于周期性传输响应消息既不支持协议控制信息，也不支持服务标识符信息（传输层by-pass旁路），因此处理过程中应将以下服务原语考虑在内。以下定义的服务原语源自ISO 10681-2：

8.2.6.2.2 C_UUData.request

服务原语请求从发送器向接收器中根据C_SA和C_TA中的地址信息所指定的对等实体，周期性传输长度为<Length>字节的<MessageData>。每次调用C_UUData.request服务时，通信层应通过发送一个C_UUData.confirm服务调用请求的方式向服务端用户发出消息传输已经完成（或失败）的信号。

C_UUData.request(

C_SA,
C_TA,
<Length>,

```
<MessageData>
    )
```

8.2.6.2.3 C_UUData.confirm

C_UUData.confirm服务由通信层发出。此服务原语用于确认由C_SA和C_TA中的地址信息标识的C_UUData.request服务已经完成。参数<C_Result>指出了该服务请求的状态（关于<C_Result>取值的定义，请参见ISO10681-2。）

```
C_UUData.confirm(
    C_SA,
    C_TA,
    <C_Result>
)
```

注：ISO 10681-2中定义的结果取值并非全部适用于周期性传输响应消息。

8.2.6.2.4 C_UUData.indication

此C_UUData.indication服务由通信层发出。服务原语将从C_SA和C_TA的地址信息确定的对等实体处接收到的长度为<Length>字节的<MessageData>传递给上一层（其定义见ISO 10681-2。）

参数<MessageData>和<Length>仅在服务已经完成指示时有效。若出现接收错误，则不会生成任何指示。因此将不需要<C_Result>参数。

```
C_UUData.indication(
    C_SA,
    C_TA,
    <Length>,
    <MessageData>
)
```

8.2.6.2.5 FlexRay 帧格式

8.2.6.2.5.1 概述

由于周期性传输响应消息既不支持PCI信息也不支持服务标识信息，因此协议数据单元(PDU)的格式不同于ISO 10681-2中的定义。然而，FlexRay周期性传输PDU的格式也包含三个字段，如表6所示。

表 6——周期性传输 PDU 格式

地址信息	长度信息	数据字段
C_AI	FPL	C_Data <MessageData>

8.2.6.2.5.2 地址信息

C_AI用于识别周期性消息PDU的通信对等实体。接收到的C_AI信息中包括C_SA和C_TA，应予以复制并纳入周期性传输PDU。每个周期性消息使用与任何其它诊断响应消息均不相同的源地址(C_SA)。

8.2.6.2.5.3 长度信息

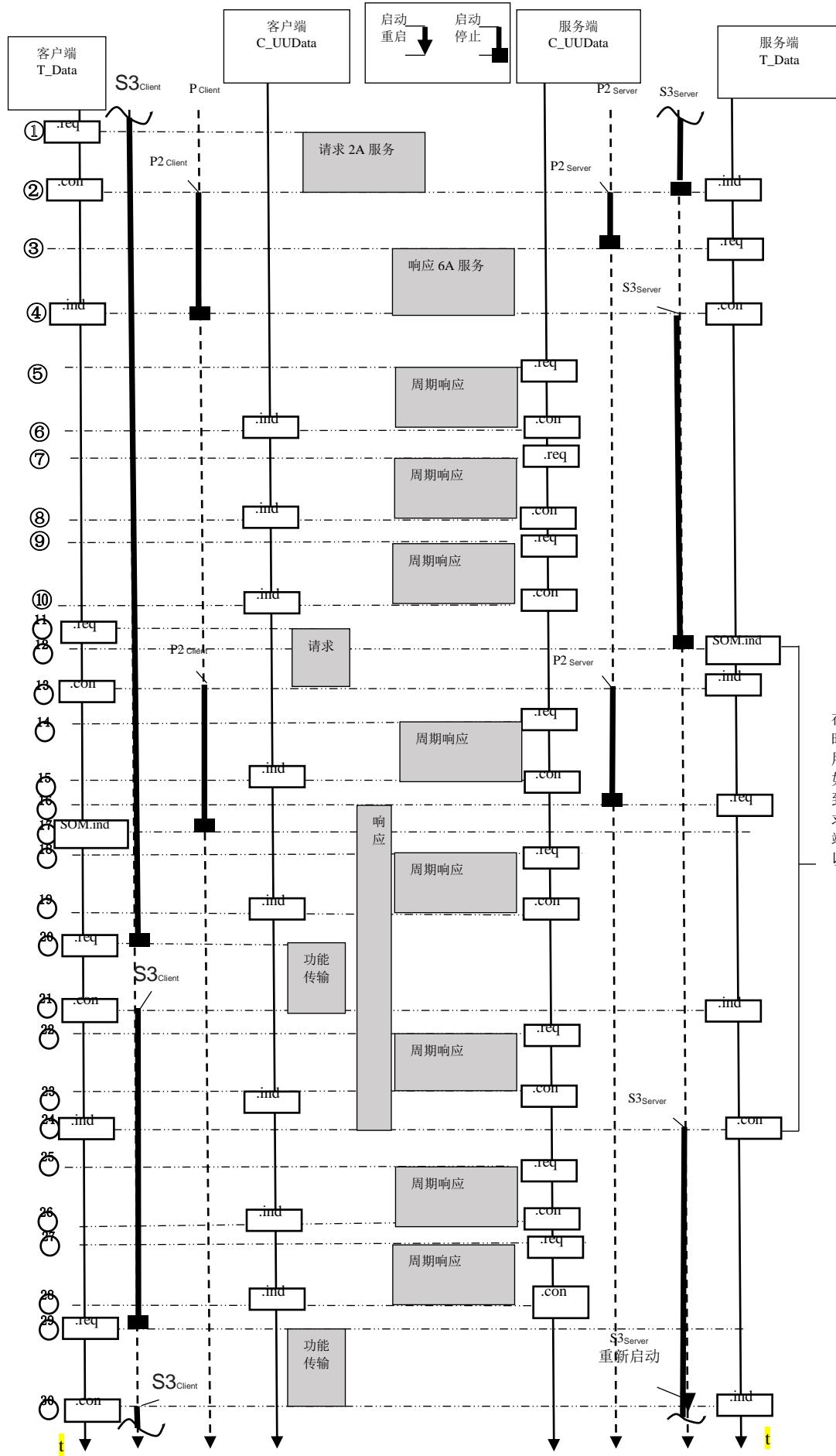
需要使用一个“帧负载长度”(FPL，定义见ISO 10681-2)参数来确定有效负载信息的数据范围。对于从FlexRay到任何其它数据链路(如CAN->FR)的周期消息，或者从任何其它数据链路到FlexRay(如CAN->FR)的周期消息，均需要此信息。

8.2.6.2.5.4 数据字段

在<MessageData>字段中，包含了使用periodicDataIdentifier(周期数据标志符，PDID)和相应数据定义的周期性数据信息。

8.2.6.3 周期性传输消息流

图4以图形方式显示了上述所述的周期性传输响应消息在服务端的处理。正如第6.6.2节提到的，此图说明周期性传输的响应消息对于服务端的S3Server计时器没有任何影响。该图假定在配置周期性调度表之前已经启动了一条non-defaultSession(非默认会话)【ReadDataByPeriodicIdentifier(根据周期标志符读取数据)服务需要一条non-defaultSession(非默认会话)才能执行】。



说明

- 1 客户端T_Data.req: 客户端的诊断应用程序向其通信层发出T_Data.req, 开始传输ReadDataByPeriodicIdentifier (通过周期性标识符读取数据) (0x2A) 请求消息。通信层将ReadDataByPeriodicIdentifier (通过周期性标识符读取数据) (0x2A) 请求消息传输给服务端。该请求消息可能是一个分段或不分段的消息【取决于请求消息中包含的periodicDataIdentifier (周期性标识符) 数量以及支持的FlexRay帧长度】。上例中假定了请求消息适合于单帧
- 2 客户端T_Data.con: 在客户端, 通过T_Data.con表示请求消息已经完成。此时, 第6章所述的响应计时适用。
- 服务端T_Data.ind: 在服务端中, 通过T_Data.ind表示请求消息已经完成。此时, 第6章所述的响应计时适用。此外, 服务端停止其S3Server计时器。
- 3 服务端T_Data.req: 假定客户端需要来自服务端的响应。服务端必须发送ReadDataByPeriodicIdentifier (通过周期性标识符读取数据) 肯定响应消息以表明该请求已被处理, 并且周期性消息的传输将随后开始
- 4 服务端T_Data.con: 在服务端, 通过T_Data.con表示ReadDataByPeriodicIdentifier (通过周期性标识符读取数据) 响应消息的传输完成。此时, 服务端重新启动其S3Server计时器, 保持已经激活的非默认会话在活动状态, 直至此计时器超时。
- 客户端T_Data.ind: 在客户端使用此服务表示响应消息已经接收。
- 5 服务端C_UUData.req: 服务端开始传输周期性响应消息 (FlexRay单帧消息)。每个周期消息既不包含PCI信息, 也不包含服务识别信息, 所使用的源地址与其它任何响应消息均不同。因此, 每次发送周期性消息时, 不管服务端当时正在处理其他任何信息, 均独立发出一条C_UUData.req请求。这意味着, 即使服务端正在处理另一诊断服务请求, 周期性响应消息的传输仍将继续。周期性响应消息的传输不会对S3Server计时器产生任何影响。
- 6 服务端C_UUData.con: 在服务端中使用此服务表示周期性响应消息的传输已经完成。
- 客户端C_UUData.ind: 在客户端使用此服务表示周期性响应消息的接收已经完成。
- 7 同(5)
- 8 同(6)
- 9 同(5)
- 10 同(6)
- 11 客户端T_Data.req: 客户端的诊断应用程序向其通信层发出一条T_Data.req请求, 以开始下一条请求消息的传输。通信层将该请求消息发送到服务端。该请求消息可以是一条单帧或多帧的消息。在上图的例子中, 假设请求消息是一条多帧消息。
- 12 服务端T_Data_SOM.ind: 在服务端中, 在周期性调度表活动期间, 使用T_Data_SOM.ind表示请求消息的开始。在处理接收到的请求消息期间, 服务端并不会停止周期性调度。这意味着服务端在处理诊断服务期间会继续发送周期性消息。客户端应注意接收这些周期性的响应消息。此外, 服务端在处理任何诊断服务期间均会停止其S3Server计时器。
- 13 客户端T_Data.con: 在客户端, 通过T_Data.con表示请求消息已经完成。此时, 第6章所述的响应计时适用。服务端T_Data.ind: 在服务端中, 通过T_Data.ind表示多帧请求消息已经完成。此时, 第6章所述的响应计时适用。
- 14 见(5)
- 15 见(6)
- 16 服务端T_Data.req: 在上图的例子中, 假设客户端需要来自服务端的响应。服务端应通过向其通信层发出T_Data.req请求来传输肯定 (或否定) 响应消息。在上例中, 假设该响应是一条多帧消息。在通过通信层发送此多帧响应消息的同时, 周期性调度表继续发送周期性响应消息。
- 17 客户端T_Data_SOM.ind: 在客户端, 使用此服务来表示响应消息的开始。
- 18 见(5)
- 19 见(6)
- 20 客户端T_Data.req: 每当客户端中的S3Client计时器超时时, 客户端会发送功能寻址的TesterPresent (测试仪保活) (0x3E) 请求消息, 以复位服务端中的S3Server计时器。
- 21 服务端T_Data.ind: 服务端正在传输先前请求的多帧响应消息。因此, 服务端不对接收到的TesterPresent (0x3E) 请求消息做出任何反应, 因为其S3Server计时器尚未被重新激活。
- 客户端T_Data.con: 此服务表示TesterPresent (测试仪保活) (0x3E) 请求消息已经被服务端收到。
- 22 见(5)
- 23 见(6)
- 24 服务端T_Data.con: 当诊断服务处理完成后, 服务端重新启动其S3Server计时器。这意味着任何诊断服务, 包括TesterPresent (测试仪保活) (0x3E), 均将复位S3Server计时器。诊断服务正在进行中, 是指: 从开始接收请求消息 (T_Data_SOM.ind或T_Data.ind接收) 至响应消息发送完成 (在需要响应消息的情况下) 或至该请求导致的动作已经完成 (在不需要响应消息的情况下) (在这个时间点本来会导致响应消息的开始) 之间的任意时间点。这包括响应代码0x78在内的否定响应消息。
- 25 见(5)
- 26 见(6)
- 27 见(5)
- 28 见(6)
- 29 客户端T_Data.req: 一旦客户端的S3Client定时器启动【non-defaultSession (非默认会话) 激活】, 则每次S3Client超时时, 均将发送功能寻址的TesterPresent (测试仪保活) (0x3E) 请求消息, 该消息不需要响应消息。
- 30 客户端T_Data.con: 一旦使用通信层的T_Data.con指示TesterPresent (测试仪保活) (0x3E) 请求消息的传输已经完成, 客户端将再次启动其S3Client计时器。这意味着功能寻址的TesterPresent (0x3E) (测试仪保活) 请求消息, 是以S3Client计时器超时为周期发送的。

图 4——周期性传输响应消息的处理

8.3 应用层要求

8.3.1 应用层服务

本章使用第5章定义的应用层服务（基于客户端-服务端系统），来执行车载服务端的诸如测试、检查、监测、诊断或编程等功能。

8.3.2 应用层协议

本部分使用第5章所定义的应用层协议。

8.3.3 应用层计时

应用层定时参数值应遵循第6章的定义。

8.4 表示层要求

表示层要求由车辆制造商负责。

8.5 会话层要求

会话层要求见第6章。

8.6 传输层/网络层接口适配

8.6.1 一般信息

本章使用第6章中定义的网络层服务来传输和接收诊断消息。本节定义了从独立于数据链路定义的传输层/网络层协议数据单元(T_PDU)，到FlexRay数据链路特定的网络层协议数据单元(N_PDU)的映射。

注：传输层/网络层服务于执行应用层会话和诊断会话的定时管理。

对于诊断信息通信，本章仅使用第6章中传输已知长度数据的技术规定。

8.6.2 DoFR 传输层/网络层接口适配

8.6.2.1 从独立于数据链路的服务原语到 FlexRay 数据链路特定服务原语的映射

表7显示了ISO 10681-2“DoFR”第2部分：通信层服务与第6章针对诊断消息的发送和接收而定义的会话层服务之间的映射接口。

表7——从 T_PDU 服务原语到 C_PDU 服务原语的映射

会话层到传输层/网络层服务原语 (根据第6章，独立于数据链路层)	DoFR网络层服务原语 (根据ISO 10681-2，与数据链路层相关)
T_Data.indication	C_Data.indication
T_DataSOM.indication	C_Data_STF.indication
T_Data.confirm	C_Data.confirm
T_Data.request	C_Data.request

8.6.2.2 从 T_PDU 到 C_PDU 的消息传输映射

为请求诊断服务请求/响应的传输定义了应用层协议数据单元参数，为了客户端/服务器中传输消息定义了通信层协议数据单元参数，两参数之间的映射关系如表8所示。

表8——从 T_PDU 参数到 C_PDU 参数的映射

T_PDU参数 (根据第6章，独立于数据链路层)	C_PDU参数 (根据ISO10681-2，与FlexRay数据链路层相关)
T_Mtype	N/A ^a
T_SA	C_SA
T_TA	C_TA
T-TAtype	C_TAtype
T-AE	N/A ^b
T-DATA[]	<MessageData>
T_Length	<Length>
N/A	<ActualLength> ^c
T_Result	<C_Result>

a: FlexRay通信层不支持远程诊断功能。

b: FlexRay通信层不支持扩展寻址。

c: 在FlexRay通信层设置为零。

网络层在确认消息发送成功(C_Data.con)时要将此确认发送给应用层，因为应用层需要收到该确认才能启动一些操作，该操作需要在传输请求/响应消息成功之后立即执行（ECU复位、波特率改变等）。

网络层和OSI更高层之间的地址映射并不一定精确复制数据链路层上编码的地址值，而是取决于具体实施概念。

8.7 数据链路层诊断实施要求

FlexRay的数据链路层实施应遵循ISO 10681-2和ISO 17458-2所述要求。

9 UDSonIP

9.1 文件概述

图25概述了实施UDSonIP所需的文件。

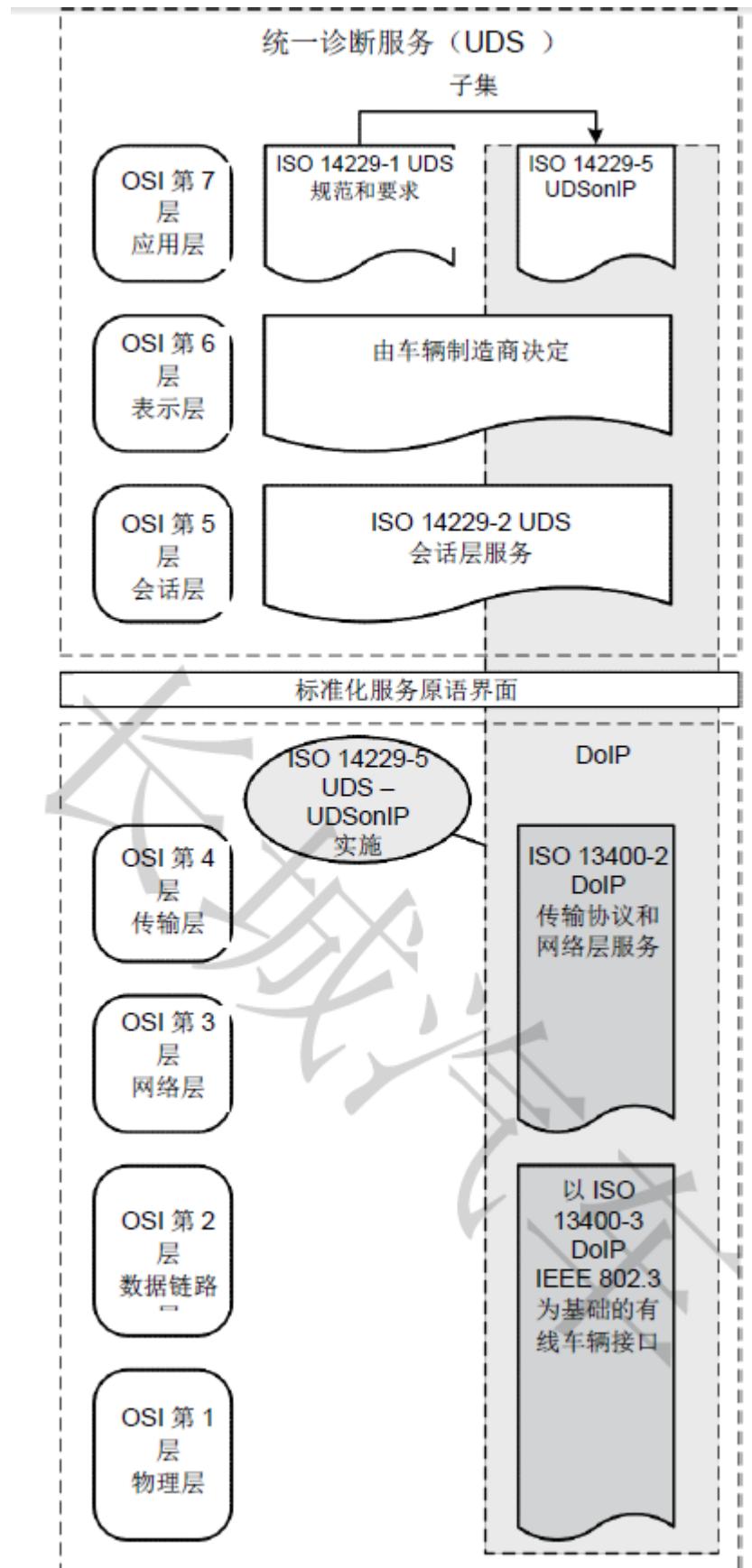


图26 OSI 模型的 UDSSonIP 文件参考

9.2 互联网协议的统一诊断服务实施

9.2.1 概述

本节描述了第5章中定义的诊断服务如何应用到IP上，并针对所有相关服务规定了适用的子功能和数据参数。

注：子功能参数定义内考虑了第5章中suppressPosRspMsgIndicationBit参数所用的最高有效位。

9.2.2 UDS on IP 服务概述

表2列举了第5章和第6章内适于实施本章内所述UDSonIP的所有服务。当特定应用程序根据本章内容实施UDSonIP时，可能会受到可用服务数量限制，并可能将该部分内容划分为特定的应用领域/诊断会话（默认会话、编程会话等）。

表2内标有“无特定IP要求”字样的服务的实施只须符合第5章和第6章内的相关规定即可，别无其他限制要求。而对于标有“特定IP要求”字样的服务，则须依据表2内所列子条款要求实施。

表2—适用的第5章统一诊断服务和数据范围概述

诊断服务名称（第5章）	备注	本章参考章节
诊断与通信管理功能单元		
DiagnosticSessionControl（诊断会话控制）	特定IP要求	参见6.3
ECUReset（ECU复位）	特定IP要求	参见6.4
SecurityAccess（安全访问）	无特定IP要求	——
CommunicationalControl（通信控制）	无特定IP要求	——
TesterPresent（保持通信）	无特定IP要求	——
SecuredDataTransmission（安全数据传输）	无特定IP要求	——
ControlDTCSetting（控制DTC设置）	无特定IP要求	——
ResponseOnEvent（事件响应）	无特定IP要求	——
数据传输功能单元		
ReadDataByIdentifier（通过标识符读取数据）	无特定IP要求	——
ReadMemoryByAddress（通过地址读取内存）	无特定IP要求	——
ReadScallingDataByIdentifier（通过标识符读取换算数据）	无特定IP要求	——
ReadDataByPeriodicIdentifier（通过周期标识符读取数据）	无特定IP要求	参见6.5
DynamicallyDefineDataIdentifier（动态定义数据标识符）	无特定IP要求	——
WriteDataByIdentifier（通过标识符写入数据）	无特定IP要求	——
WriteMemoryByAddress（通过地址写入内存）	无特定IP要求	——
存储数据传输功能单元		
ClaerDiagnosticInformation（清除诊断信息）		
ReadDTCInformation（读取诊断故障代码DTC信息）	无特定IP要求	——
ReadDTCInformation（读取诊断故障代码DTC信息）	无特定IP要求	——
输入/输出控制功能单元		
InputOutputControlByIdentifier（通过标识符控制输入和输出）	无特定IP要求	——
常规功能单元的远程激活		
RoutineControl（常规控制）	无特定IP要求	——
上传/下载功能单元		
RequestDownload（请求下载）	无特定IP要求	——
RequestUpload（请求上传）	无特定IP要求	——
TransferData（传输数据）	无特定IP要求	——
RequestTransferExit（请求终止传输）	无特定IP要求	——
RequestFileTransfer（请求文件传输）	无特定IP要求	——

9.2.3 DiagnosticSessionControl（0x10）服务

除第5章内所述的一般实施要求外，在实施UDSonIP时还须考虑以下因素：继续执行诊断通信前，为按照ISO13400-2规定创建新的TCP连接并激活路由，会发生会话变更，TCP可能会因此断连。

9.2.4 ECURestart（0x11）服务

除第5章内所述的一般实施要求外，在实施UDS on IP时还须考虑因素：当在服务器内执行ECUReset时，客户端和服务器间的TCP连接将会断开。在ECU发挥路由器功能时，也会发生这种情况。鉴于此，在已复位ECU及其他相关ECU继续执行诊断通信前，需按照ISO13400-2规定创建新的TCP连接，并激活路由。

9.2.5 ReadDataByPeriodicIdentifier (0x2A) 服务的 DoIP 实施要求

9.2.5.1 周期数据响应消息

ReadDataByPeriodicIdentifier服务允许客户端向一个或多个pDID所识别的服务器发出数据记录值周期发送请求。对于DoIP上所实施的ReadDataByPeriodicIdentifier服务，由于其响应消息结构不同于其他UDS服务的响应消息结构，因此，其所使用地址信息（DoIP_AI）也须不同。

注：若车辆制造商要求各周期相应消息需有时间戳，则时间戳的尺寸、清晰度及其在数据记录内的位置均由车辆制造商自行决定。

使用ReadDataByPeriodicIdentifier的服务时，应考虑指定一种整车支持的数据格式，使其适用于所有由数据链路而非DoIP构成的车载网络，如，若电气车辆体系结构内除了DoIP外还有CAN，则DID数据总长度不得超出CAN协议的长度限制以确保数据格式的统一。

表3内规定了周期数据响应信息的相关要求。

表 3—周期传输—周期数据响应消息映射要求

消息类型	客户端请求要求	服务器响应要求	更多服务器限制条件
周期数据响应消息采用不同的 DoIP 逻辑地址 (DoIP_AI) 进行周期消息的传输	无限制	周期传输只能传输单个的 DoIP 响应消息	<p>周期传输被视为定期诊断请求处理，对应的响应通过网络层发出（作为 DoIP 诊断信息，服务标志符为 0x6A）。</p> <p>收到显示正响应传输已完成的 DoIP_Data.confirm 确认后，应用程序立即启动一个单独的调度程序处理周期传输。</p> <p>服务期内的该调度程序会将周期传输视为一个编址 (DoIP_AI) DoIP 诊断信息进行处理，而这种情况仅限于周期数据响应。该地址须由车辆制造商选定，以指示周期响应消息。</p>

9.2.5.2 周期传输响应消息处理

9.2.5.2.1 概述

由于周期响应消息既不支持协议控制信息也不支持服务标识符信息，因此，需考虑到，按照ISO13400-2规定，以下服务原语须采用单独参数。

9.2.5.2.2 DoIP_Data 请求

服务原语请求发送器以<Length> (<长度>) 字节的方式向接收器对等实体传输<MessageData> (<消息数据>)，对等实体均由DoIP_SA、DoIP_TA及DoIP_TAtype类地址信息区别标识（相关参数定义见ISO13400-2）。

每次发出DoIP_Data.request服务请求后，DoIP层须向服务客户端发送一个DoIP_Data.confirm确认服务呼叫以提示消息传输已完成（或传输失败）：

DoIP_Data.request 请求（

```

DoIP_SA
DoIP_TA
DoIP_TAtype
<MessageData>
<Length>
)
```

9.2.5.2.3 DoIP_Data.confirm 确认

DoIP层发出DoIP_Data.confirm确认服务后，由服务原语确认DoIP_SA、DoIP_TA和DoIP_TAtype类地址信息所标识的DoIP_Data.request请求服务的完成状态，最后由参数<DoIP_Result>显示服务请求状态（相关参数定义请见ISO13400-2）。

DoIP_Data.confirm 确认（

```

DoIP_SA
DoIP_TA
```

```

DoIP_TAtype
<DoIP_Result>
)

```

9.2.5.2.4 DoIP_Data.indication 指示

DoIP层发出DoIP_Data.indication指示服务后，由服务原语指示<DoIP_Result>事件并利用DoIP_SA、DoIP_TA和DoIP_TA类地址信息所标识的对等协议实体传输的<Length>字节向相邻上层传输<MessageData>（相关参数定义请见ISO13400-2）

只有当<DoIP_Result>和 DoIP_OK 对等时，参数<MessageData>和<Length>方才有效。

DoIP_Data.indication指示（

```

DoIP_SA
DoIP_TA
DoIP_TAtype
<MessageData>
<Length>
<DoIP_Result>
)

```

收到DoIP诊断消息后，随即发出DoIP_Data.indication指示服务呼叫。

若DoIP层探测出DoIP诊断消息内存在任何错误，其将会忽略该消息，且不会向相邻上层发送DoIP_Data.indication指示。

9.2.5.2.5 DoIP 帧格式

9.2.5.2.5.1 格式

DoIP周期传输响应消息由表4内所述字段组成。

表4—周期传输响应消息—DoIP 消息帧示例

消息方向:		车辆→客户端		
消息类型:		周期传输响应信息		
ISO13400-2	第 5 章			
DoIP 通用标题字节	-	名称	字节值	助记符
#1	-	ISO13400—协议版本	0x02	-
#2	-	ISO13400—反向协议版本	0xFE	-
#3	-	ISO13400—净负荷类型	0x8001	GH_PT
#4	-	ISO13400—净负荷类型		GH_PT
#5	-	ISO13400—净负荷长度	12	GH_PL
#6	-	ISO13400—净负荷长度		GH_PL
#7	-	ISO13400—净负荷长度		GH_PL
#8	-	ISO13400—净负荷长度		GH_PL
DoIP 净负荷字节	A_Data_Byte	名称	字节值	助记符
#1	-	ISO13400—源地址	如 0x6A0 (VM 地址)	SA
#2	-	ISO13400—源地址		SA
#3	-	ISO13400—目标地址	e.g.0x0E00(测试仪地址)	TA
#4	-	ISO13400—目标地址		TA
#5	#1	ISO13400—用户数据/第 5 章周期数据标志符	0x00—0xFF	Pdid
#6	#2	ISO13400—用户数据/第 5 章 dataRecord.data#1	0x00—0xFF	DATA_1
#7	#3	ISO13400—用户数据/第 5 章 dataRecord.data#2	0x00—0xFF	DATA_2
#8	#4	ISO13400—用户数据/第 5 章 dataRecord.data#3	0x00—0xFF	DATA_3

#9	#5	ISO13400—用户数据/第 5 章 dataRecord.data#4	0x00—0xFF	DATA_4
#10	#6	ISO13400—用户数据/第 5 章 dataRecord.data#5	0x00—0xFF	DATA_5
#11	#7	ISO13400—用户数据/第 5 章 dataRecord.data#6	0x00—0xFF	DATA_6
#12	#8	ISO13400—用户数据第 5 章 dataRecord.data#7	0x00—0xFF	DATA_7

9.2.5.2.5.2 DoIP 通用标题

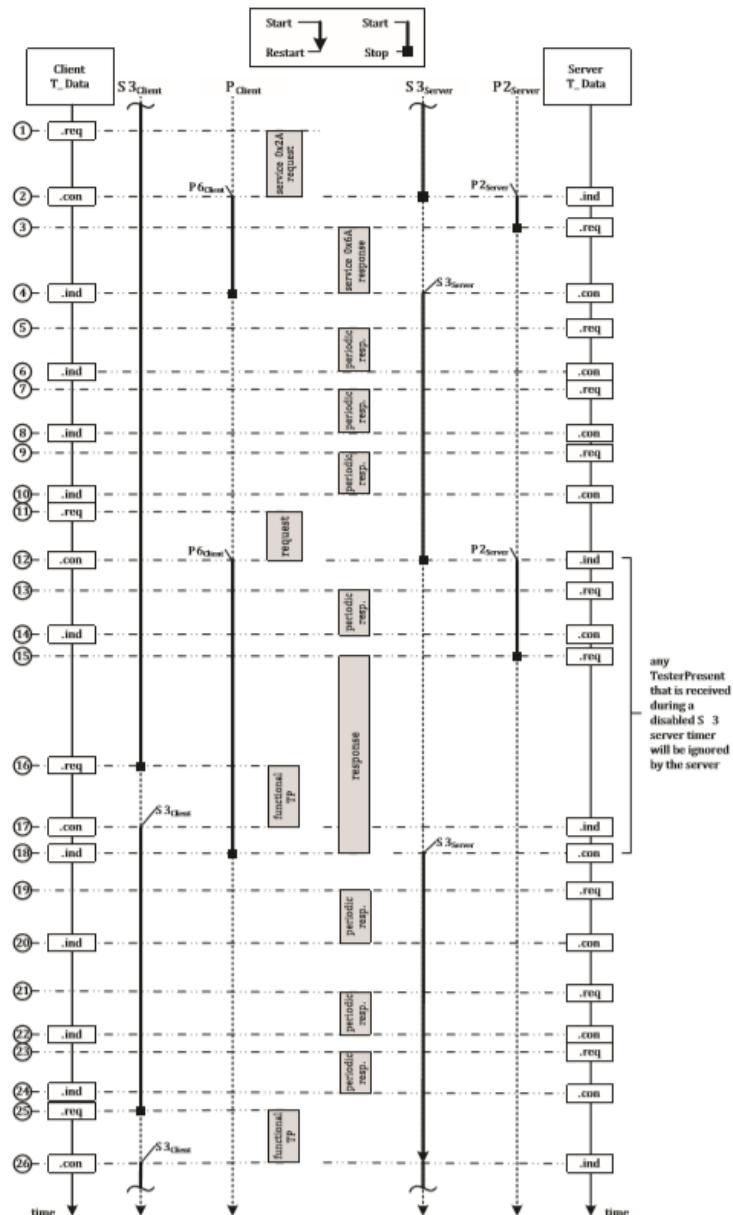
周期响应消息与非周期响应消息的区别在于是否有特定的DoIP_AI。如何实施该类不同的DoIP_AI（如，SA不同、TA不同或二者均不同）则取决于车辆制造商。

表4内的目标地址(TA)为ISO13400-2内所述测试设备地址范围的逻辑地址。源地址(SA)则指ISO13400-2内所述车辆制造商地址范围的逻辑地址，指示周期传输响应而非非周期诊断响应消息。

9.2.5.2.5.3 DoIP 净负荷

表4内所述DoIP净负荷包含了SA信息、TA信息和周期数据信息，包括pDID和相应的数据记录。

图2图解了服务器处理DoIP周期传输响应消息的程序。此外，从图2可以看出周期传输的响应消息对服务器的S3Server定时器无任何影响。图2内假定在配置周期调度程序前已激活了一个非默认会话（因为要执行ReadDataByPeriodicIdentifier服务，需有一个非默认会话）。



说明

1 客户端 T_Data.req：客户端诊断应用程序向其通信层发出一个 T_Data.req 进而启动 ReadDataByPeriodicIdentifier (0x2A) 请求消息的传输。之后，由该通信层向服务器传输 ReadDataByPeriodicIdentifier (0x2A) 请求消息。

2 客户端 T_Data.con: 客户端通过 T_Data.con 显示请求消息的完成状况。接着，立即执行第 6 章内所述的响应定时。

服务器 T_Data.ind: 服务器通过 T_Data.ind 显示请求消息的完整状况。接着，立即执行第 6 章内所述的响应定时。之后，服务器启动 S3Server 定时器关闭程序，关闭该定时器。

3 服务器 T_Data.req: 假设客户端要求服务器做出响应，则服务器应传输 ReadDataByPeriodicIdentifier 的正响应消息以表明相关请求已被处理，稍后即将传输周期消息。

4 服务器 T_Data.con: 服务器通过 T_Data.con 显示 ReadDataByPeriodicIdentifier 响应消息的传输完成状况，接着，重启 S3Server 定时器使非默认会话在到时前保持激活状态。

客户端 T_Data.ind: 客户端内显示响应消息的接收情况。

5 服务器 T_Data.req: 服务器开始传输周期响应消息（单个的 DoCAN 帧消息）。本例中的周期消息所采用的源地址不同于任何其他响应消息所采用的地址。须将服务器发出的各响应消息（周期和非周期消息）序列化，但是，周期和非周期消息应采用不同的净负荷类型 0x8001 和不同的源地址。周期响应消息的传输对 S3Server 定时器无任何影响。

6 服务器 ServerT_Data.con: 服务器内显示周期响应消息传输的完成状况。

客户端 ClientT_Data.ind: 客户端显示周期响应消息接收的完成状况。

7 见 (5)。

8 见 (6)。

9 见 (5)。

10 见 (6)。

11 客户端 T_Data.req: 客户端诊断应用程序通过向其通信层发出 T_Data.req，启动下一请求消息的传输。之后，由通信层向服务器传输请求消息。

12 客户端 T_Data.con: 客户端通过 T_Data.con 显示请求消息的完成状况。接着，立即执行第 6 章内所述的响应定时。

服务器 T_Data.ind: 服务器通过 T_Data.ind 显示多帧请求消息的完成状况。接着，立即执行第 6 章内所述的响应定时。

13 见 (5)。

14 见 (6)。

15 服务器 T_Data.req: 如图所示，假定客户端已请求服务器发出响应，因此，服务器应通过向其通信层发送 T_Data 请求来传输正（或负）响应消息。

16 客户端 T_Data.req: 客户端 S3client 定时器到时后，客户端会发出一个功能性编址的 TesterPresent (0x3E) 请求消息以重置服务器内的 S3Server 定时器。

17 服务器 T_Data.ind: 服务器正在传输之前请求的响应消息。而且，由于其 S3Server 定时器尚未重新激活，因此，其不得对收到的 TesterPresent (0x3E) 请求消息做出任何响应。

客户端 T_Data.con: 服务器内显示 TesterPresent (0x3E) 请求消息的接收情况。

18 服务器 T_Data.con: 当诊断服务处理完毕后，服务器会重新启动 S3Server 定时器，也即，包括 TesterPresent (0x3E) 在内的所有诊断服务会重置 S3Server 定时器，此处的诊断服务是指请求消息的接收（T_Data.ind 接收）至有响应消息要求的情况下响应消息传输结束，或无响应消息要求（虽无要求，但是一旦到时，还是会发出响应消息）情况下因请求所产生的任何反应结束期间进行的诊断服务。其中包括响应代码为 0x78 的负响应消息。

19 见 (5)。

20 见 (6)。

21 见 (5)。

22 见 (6)。

23 见 (5)。

24 见 (6)。

25 客户端 T_Data.req: 一旦客户端内的 S3client 定时器启动（非默认会话激活），即使无响应消息要求，只要 S3client 定时器到时，就会开始传输功能性编址的 TesterPresent (0x3E) 请求消息。

26 客户端 T_Data.con: 一旦客户端通信层的 T_Data.con 显示 TesterPresent (0x3E) 请求消息传输已完成，客户端就会再次启动其 S3client 定时器，即，每当 S3client 定时器到时，将会定时发送功能性编址的 TesterPresent (0x3E) 请求消息。

服务器 T_Data.ind: 服务器内显示 TesterPresent (0x3E) 请求消息的接收状况。之后，服务器将重新激活 S3Server 定时器。

图 2—周期性传输响应消息的处理

9.3 应用层要求

9.3.1 应用层服务

本章内容针对客户机系统采用了第5章中定义的应用层服务，以实现对车载服务器的测试、检查、监控、诊断或编程等功能。

9.3.2 应用层协议

本章采用第5章内所述的应用层协议。

9.3.3 应用层定时

应用层定时参数值须符合第6章内相关定义。

9.4 表示层要求

表示层要求由车辆制造商自行决定。

9.5 会话层要求

会话层要求请见第6章内相关规定。

9.6 传输/网络层接口适配

9.6.1 一般信息

本章采用了第6章内规定的网络层服务，用于传输和接收诊断消息。本节规定了数据链路独立传输/网络层协议数据单元 (T_PDU) 在IP数据链路专有网络层协议数据单元 (DoIP_PDU) 上的映射方法。

注：应用层和诊断会话管理定时均通过传输/网络层服务执行。

9.6.2 DoIP 传输/网络层接口适配

9.6.2.1 数据链路独立服务原语至 IP 数据链路非独立服务原语的映射

表5规定了第6章内所述传输层独立服务原语至ISO13400-2内所述IP非独立服务原语的映射。

表 5—服务原语至 DoIP_PDU 服务原语的映射

与传输层服务原语的会话 (第 6 章内所述的数据链路独立)	DoIP 网络层服务原语 (ISO 13400-2 内所述的数据链路不独立)
T_Data.indication	DoIP_Data.indication
T_Data.confirm	DoIP_Data.confirm
T_Data.request	DoIP_Data.request

9.6.2.2 为传输消息，T_PDU 至 DoIP_PDU 的映射

按照表6，将规定的用于请求传输诊断服务请求/响应的应用层协议数据单元映射至用于传输客户端/服务器内消息的通信层协议数据单元。

表 6—T_PDU 参数至 DoIP_PDU 参数的映射

T_PDU 参数 (第 6 章内所述的数据链路独立)	DoIP_PDU 参数 (ISO13400-2 所述的 IP 数据链路不独立)
T_Mtype	DoIP_Mtype
T_SA	DoIP_SA
T_TA	DoIP_TA
T_TAtype	DoIP_TAtype
T_AE	N/Ab
T_Data[]	<MessageData>
T_Length	<Length>
T_Result	<DoIP_Result>

- a DoIP 不支持远程诊断功能。
- b DoIP 不支持地址扩展。

9.7 数据链路层诊断实施要求

IP数据链路实施须符合ISO13400-3内所述要求。

10 UDSonK-Line

10.1 文件概述

图1说明了第5章、第6章、ISO 14230-1和ISO 14230-2的文档引用。本章仅使用了第5章中定义的诊断服务的子集（见表2）。

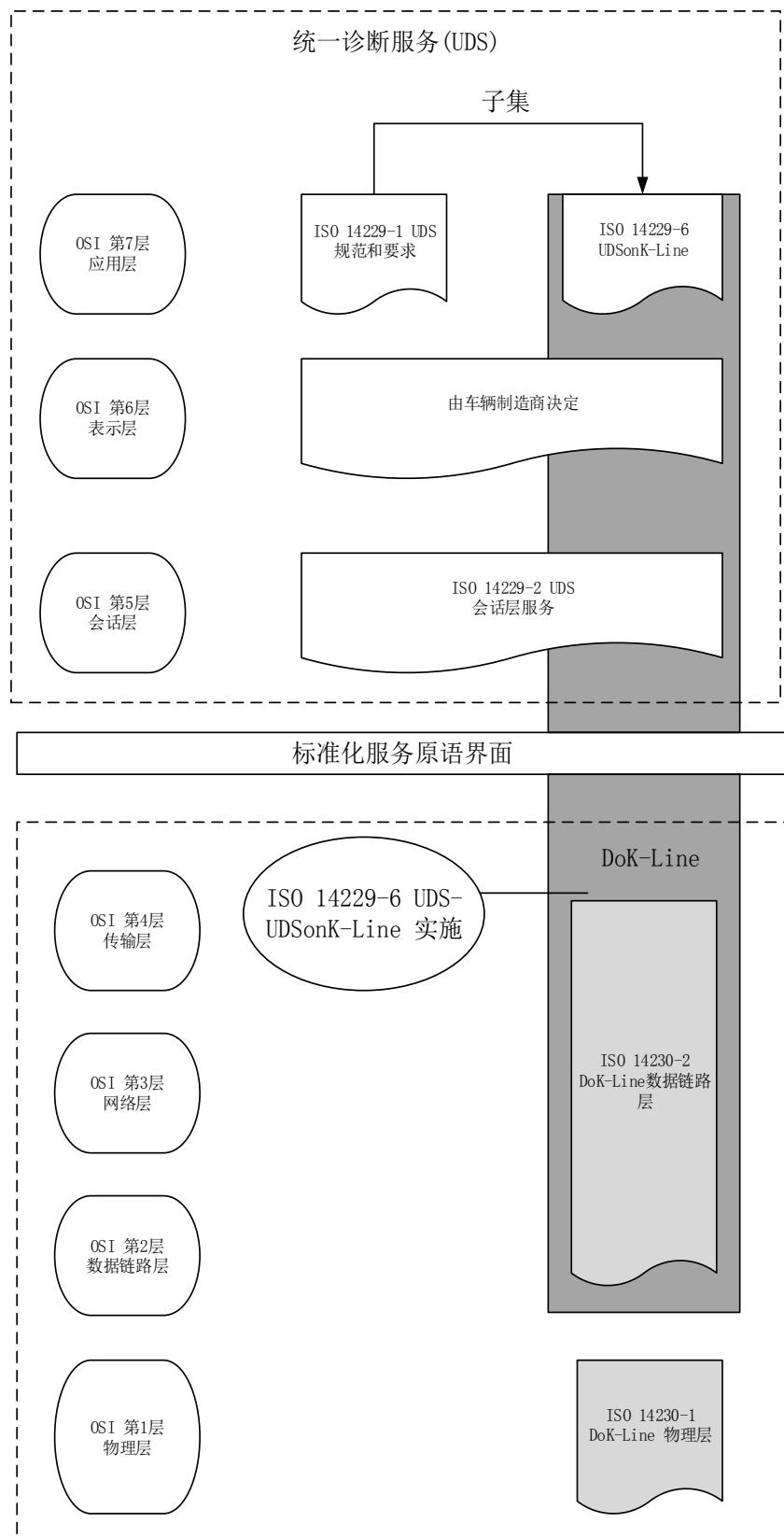


图27 OSI 模型的 UDSonK-Line 参考文件

10.2 K-Line 的统一诊断服务实施

10.2.1 概述

本章定义了第5章中定义的诊断服务如何应用到DoK-Line上。并对每个适用的服务，都定义了适用的功能和数据参数。

为确保DoK-Line和其他通信上应用层和会话层的统一实施，本章采用了第6章中规定的会话层协议并将重点放在必要的修改和联系上，以便依据ISO14230-2进行实施。

注1：子功能参数定义考虑了第5章中 suppressPosRspMsgIndicationBit 参数使用的最高有效位。

注2：系统设计员应负责保证：在客户端不需要响应信息（suppressPosRspMsgIndicationBit=TRUE ('1')）但服务器可能需要更多P2Server来处理请求信息的情况下，客户端能在后续请求之间留出足够的时间。

10.2.2 UDS on K-Line 服务概述

表8旨在列出所有适用于UDS on K-Line实施的统一诊断服务以供参考。表中汇总了所有适用服务。某些应用程序采用本章实施UDS on K-Line可能会限制可用服务的数量，并可能将这些服务划分为特定应用区域/诊断会话（默认会话、编程会话等）。

由于采用了数据链路层，故应限制所有诊断服务的数据长度（见7.1）。

注：某些诊断服务，如读取DTC信息（取决于予以报告的DTC数量）可能会超出信息长度限制。

注：不支持表8中未列出的服务。

图28 适用的第5章统一诊断服务和数据范围概述

诊断服务名称(第5章)	备注	本文件中的参考章节
诊断与通信管理功能单元		
DiagnosticSessionControl (诊断会话层控制)	无特定的K-Line要求	—
ECUReset (ECU复位)	无特定的K-Line要求	—
SecurityAccess (安全访问)	无特定的K-Line要求	—
TesterPresent (保持通信)	无特定的K-Line要求	—
AccessTimingParameter (访问定时参数)	K-Line特定要求	参见6.3.1
ControlDTCSetting (控制DTC设置)	无特定的K-Line要求	—
LinkControl (链路控制)	特定K-Line要求	参见6.3.2
数据传输功能单元		
ReadDataByIdentifier (通过标识符读取数据)	无特定的K-Line要求	—
ReadMemoryByAddress (通过地址读取内存)	无特定的K-Line要求	—
ReadScalingDataByIdentifier (通过标识符读取换算数据)	无特定的K-Line要求	—
DynamicallyDefineDataIdentifier (动态定义数据标识符)	特定K-Line要求	参见6.4
WriteDataByIdentifier (通过标识符写入数据)	无特定的K-Line要求	—
WriteMemoryByAddress (通过地址写入内存)	无特定的K-Line要求	—
存储数据传输功能单元		
ReadDTCInformation (读取DTC信息)	无特定的K-Line要求	—
ClearDiagnosticInformation (清除诊断信息)	无特定的K-Line要求	—
输入/输出控制功能单元		
InputOutputControlByIdentifier (通过标识符控制输入和输出)	无特定的K-Line要求	—
常规功能单元的远程激活		
RoutineControl (常规控制)	无特定的K-Line要求	—
上传/下载功能单元		
RequestDownload (请求下载)	无特定的K-Line要求	—
RequestUpload (请求上传)	无特定的K-Line要求	—
TransferData (传输数据)	无特定的K-Line要求	—
RequestTransferExit (请求终止传输)	无特定的K-Line要求	—

10.2.3 诊断与通信控制功能单元

10.2.3.1 访问定时参数 (0x83) 服务

ISO 14230-2 (DoK-Line) 中定义了此服务。第5章中定义的子功能参数不适用使用访问定时参数服务受限于数据链路层的定时参数 (P2Server_min、P2Server_max、P3Client_min、P3Client_max、P4Sender_min)。对于会话层定时参数，则应用于第6章中的定时参数P2Server_max和S3Client。

数据链路层应忽略ISO 14230-2中定义的客户请求数据记录中的参数P2Client_max和P3Client_max (见11.4)。

DoK-Line改变参数请求服务原语用于请求改变本地协议实体上的定时参数值。

参数S3Server和S3Client不是访问定时参数服务的一部分。

10.2.3.2 链路控制 (0x87) 服务

在DoK-Line上使用链路控制，以修改数据链路相关参数，使得诊断通信宽带最大化（如编程事件）。这会引起DoK-Line周期设计的改变（如更改波特率）。但是，具体的实施是由车辆制造商规定的，并应在单独的实施规范中进行说明。

在DoK-Line上实施此服务时，下列要求应适用：

步骤1：通信策略应遵守第5章中规定的要求。

步骤2：只有顺利完成步骤1，才应请求此步骤。在功能通信的情况下，当执行模式转换时（suppressPosRspMsgIndicationBit=TRUE），任何服务器均不应有响应，因为其中一个服务器可能已转换到新模式时其他服务器仍在进行中。

表9规定了在K-Line上实施这种服务可以使用的子功能参数。

表10 子功能参数定义

十六进制(位元6-0)	说明	Cvt	助记符
01	<p>Verify Mode Transition With Fixed Parameter（用固定参数验证模式转换）</p> <p>此参数用于验证是否能用链路控制模型标识器数据参数规定的预定义DoK-Line时序设计执行转换。</p>	M	VMTWFP
02	<p>Verify Mode Transition With Specific Parameter（用特定参数验证模式转换）</p> <p>链路记录的格式（用于使用特定参数验证模式转换）是由车辆制造商特定的。</p>	M	VMTWSP
03	<p>Transition Mode（转换模式）</p> <p>只有一个信号物理点对点连接激活时才允许更改波特率。第6章中规定的固定波特率适用。</p> <p>该子功能参数请求服务器将DoK-Line转换为前面验证消息已请求的时序设计。</p>	M	TM

10.2.4 数据传输功能单元

此服务不支持在K-Line上实现定期数据标识符。

10.3 应用层要求

10.3.1 应用层服务

本章基于客户端服务器的系统，使用第5章中定义的应用层服务，执行车载服务器的测试、检验、监测、诊断或编程等功能。

根据ISO 14230-2，将消息长度限制为255字节。两个方向均默认这一长度。

当检测到开始传输消息且消息长度适用时，消息缓冲区受会话层控制并由数据链路层发出请求。

10.3.2 应用层协议

本章采用第5章中定义的应用层协议。

10.3.3 应用层定时

10.3.3.1 概述

明确规定了应用层和会话层定时参数和如何将这些参数应用到客户端和服务器。

10.3.3.2 应用层定时参数值

应根据第6章在会话层处理P2Server_max。

P3Client/S3Client：ISO 14230-2中的P3Client_min定时值映射到第6章中的P3Client定时值。这仅关系到客户端。第6章中的P3Client_max定时值映射到本标准中的S3Client_max定时值。S3Client的监测由在会话层中执行。

10.4 表示层要求

表示层要求由车辆制造商负责确定。

10.5 会话层要求

会话层要求见第6章。

10.6 传输/网络层接口适配

10.6.1 一般信息

I本章使用第6章中定义的网络层服务，来传输和接受诊断消息。本节定义了数据链路独立传输/网络层协议数据单元（T_PDU）在K-Line数据链路特定网络层协议数据单元（N_PDU）上的映射。

注：传输/网络层服务用于执行应用层和诊断会话管理定时。

10.6.2 DoK-Line 传输/网络层接口适配

10.6.2.1 数据链路独立服务原语至 K-Line 数据链路相关服务原语的映射

表4规定了ISO14230-2 DoK-Line第2部分：数据链路层服务与第6章会话层服务之间的映射接口。
表 4-T_PDU 服务原语至 DL_PDU 服务原语的映射

与传输/网络层服务原语的会话（第6章内规定的数据独立链路）	DoK-Line 数据链路层服务原语（ISO 14230-2内规定的数据相关链路）
T_Data.indication	DoK-Line_Data.indication
T_DataSOM.indication	DoK-Line_Data_FB.indication
T_Data.confirm	DoK-Line_Data.confirm
T_Data.request	DoK-Line_Data.request

DoK-Line_Data_FB.indication服务原语映射到T_DataSOM.indication上通常表示K-Line数据链路开始接收消息。

10.6.2.2 T_PDU 至 DL_PDU 的消息传输映射

对于为请求传输诊断服务请求/响应而定义的应用层协议数据单元参数，根据表5将其映射到用于客户端/服务器中传输消息的网络层协议数据单元参数上。

T_PDU 参数（根据第6章内规定的数据独立链路）	N_PDU 参数（根据ISO 14230-2内规定的DoK-Line 数据相关链路）
T_Mtype	Mtype
T_SA	DoK-Line_SA
T_TA	DoK-Line_TA
T_TAtype	DoK-Line_TAtype
T_AE	不支持 K-Line
T_Data[]	<消息数据>
T_Length	<长度>
T_Result	<Result_Dok-Line>

网络层确认消息（DoK-Line_Data.con）传输成功后转发给应用层，因为应用层需在收到这一消息后开始执行这些动作，一旦传输请求/响应消息（ECUReset、BaudrateChange等）后这些动作应被立即执行。

网络层与OSI高层之间的地址映射不一定能准确复制数据链路层上编码的地址值，因此，具体取决于实施方案。

映射定义见ISO14230-2。排放相关的OBD数值见ISO14230-4。

10.7 数据链路层诊断实施要求

10.7.1 一般信息

本章应用了ISO 14230-2中定义的数据链路层规范，来传输和接收诊断消息。

10.7.2 数据分割

不支持DoK_数据分割（见 ISO 14230-2）。

10.7.3 会话处理

第6章中规定了依赖于S3Client的会话处理。本章描述了在S3Client发生超时时，如何使数据链路层与会话层同步。S3Client超时由会话层处理。数据链路层利用P3Client_max超时。

S3Server超时与P3Client_max超时同步，即，如果出现S3Server超时，那么P3Client_max超时应同样适用。另外，P3Client_max超时还适用于默认会话。

在P3Client_max超时的情况下，串行通信链路应关闭，在接收下一请求之前须检测到新初始化阶段（见ISO 14230-2）。

注：下列情况要求协议初始化阶段：

- 服务器开机后，客户端发出第一个请求；
- 默认会话中为 P3Client_max 超时；
- 非默认会话中 P3Client_max 超时（与 S3Server 超时相关，S3Server 超时只与第 6 章有联系，不适用于数据链路层）。

注：客户端须监测其数据链路层中的P3Client，来执行协议初始化。

注：S3Server超时和P3Client_max超时的同步是通过P3Client_max相关的定时要求实现的（见11.4）。

10.7.4 定时

P2Server_min应在数据链路层（ISO14230-2）进行处理。

P3Client_max/S3Client：

——对于串行数据链路上的通信，默认会话与非默认会话之间的超时处理没有区别。因此，S3Client 超时同样适用于经过 P3Client_max 超时的默认会话。

——P3Client_max 时间限制为 5 000 ms。

注：P3Client_max不能通过应用层服务（如AccessTimingParameters）修改。考虑到这项限制条件，默认会话的超时监测（只针对DoK-Line）可以通过数据链路层完成，而不对应用层相关的服务原语造成影响。

10.7.5 通信协议的初始化，启动和停止

初始化序列（5-BAUD-INIT初始化序列或通过WakeUpPattern和StartCommunication快速初始化）和ISO 14230-2中定义的StopCommunication服务完全在数据链路层处理。

如果数据链路层收到StopCommunication请求，其应呼叫可用的会话层接口（ChangeParameter.request）将诊断会话更改为默认会话。另外，数据链路层应停止监测默认会话中的P3Client超时。

物理/功能寻址：物理和功能寻址的处理在数据链路层执行。在串行数据链路上无法将有功能编址的TesterPresent消息与任何其他物理编址消息并行发送。这就要求客户端在与任何其他消息相同的时序中发送激活逻辑消息，并观察相应的数据链路层定时值（P3Client）。

10.7.6 错误处理

为了遵守ISO 14230-2中有关错误处理和P3Client_max超时正确处理的要求，在下列错误情况下，须调用带有参数（_NOT_OK）的L_Data.indication：

- P4Sender_max 超时，
- 请求错误（错误数据头/校验和）
- 错误来源或目标地址。

只要未检测到有效初始模式，便不得发出这些指示。

10.8 非易失性服务器内存编程过程

非易失性服务器内存编程过程参见第5章。无特定的适用K-LINE要求。

11 局域互连网络(UDSonLIN)上的 UDS

11.1 文件概述

图1显示了第5章、第6章、ISO 17987(所有部分)的文档引用情况。本章仅用到第5章中定义的诊断服务的一个子集（见表3）

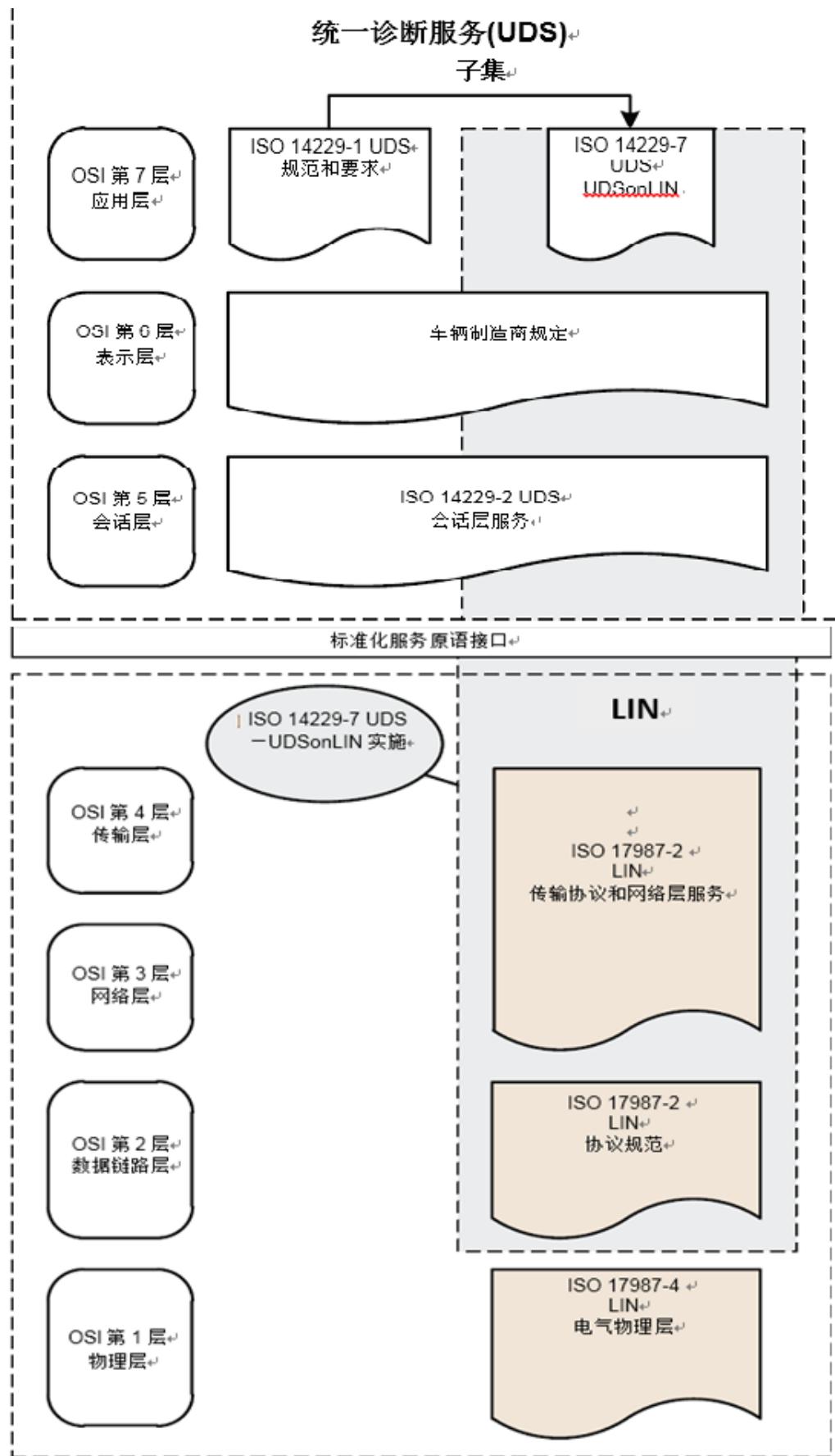


图 1—UDSonLIN 文档引用情况与 OSI 模型的对照

11.2 UDSonLIN 实现要求

11.2.1 通用

本条说明了如何对LIN实施如第5章所定义的诊断服务。

为了保证ISO 17987和其他通信的应用层和会话层的共同实现，本章使用了第6章中定义的会话层协议，并专注于必要的修订和接口，以将其应用于ISO 17987系列。

在子功能的参数定义中，假定最高有效位是用于第5章中定义的抑制肯定响应消息指示位参数。

根据ISO17987系列，车辆制造商有责任设置主节点和从节点来交换UDSonLIN信息。

11.2.2 诊断类的定义

11.2.2.1 概述

从节点的体系结构、诊断通信性能和传输协议需求通过将诊断服务功能分为三个诊断类来满足。

因此诊断类根据其诊断功能和复杂性的级别分配给每个从节点。

11.2.2.2 诊断I类

智能和简单的设备，如智能传感器和执行器，不需要或很少的诊断功能。执行器控制、传感器读取和故障存储器处理由主节点使用信号承载帧完成。因此，这些任务不需要支持特定的诊断。故障指示总是基于信号。

11.2.2.3 诊断II类

诊断II类从节点类似于诊断I类从节点，但它规定支持节点标识。扩展节点识别通常是车辆制造商所需要的。测试仪或主节点使用第5章诊断服务来请求扩展节点标识信息。执行器控制、传感器读取和故障存储器处理由主节点使用信号承载帧完成。因此这些任务不需要支持特定的诊断。故障指示总是基于信号。

11.2.2.4 诊断III类

诊断III类从节点是具有增强应用功能的设备，通常执行其自己的本地信息处理(例如功能控制器，本地传感器/执行器回路)。从节点执行超出基本传感器/执行器功能的任务，因此需要支持扩展的诊断。直接执行器控制和原始传感器数据通常不与主节点交换，因此不包括在信号承载帧中。用于I/O控制、传感器值读取和参数配置(超出节点配置)的第5章诊断服务是必需的。

诊断III类从节点具有内部故障存储器，以及相关的读取和清除服务。可选地从节点的重新编程(闪存/NVRAM重新编程)是可能的。需实现引导加载程序和必要的诊断服务来解锁设备启动下载和传输数据等。

诊断II类和诊断III类之间的主要区别是诊断II类的LIN主节点和LIN从节点之间的诊断能力分配，而诊断III类的LIN从节点，LIN主节点中没有实现LIN从节点的诊断应用功能。

11.2.3 LIN节点要求

11.2.3.1 主节点要求

11.2.3.1.1 主报文路由

主节点通常是高性能ECU，并且在大多数实现中支持第5章诊断服务。主节点和外部测试设备通过主干总线(如ISO 11898)连接。主节点应接收来自主干总线的从节点的所有诊断请求，并将它们路由到适当的LIN集群。来自从节点的响应应通过主节点路由回主干总线。

所有发送到从节点的诊断请求和响应报文都可以在网络层路由(即没有应用层路由)。主节点应执行LIN传输协议(参见ISO 17987-2)以及用于主干总线的传输协议(例如ISO 15765-2)。

11.2.3.1.2 主节点故障管理、传感器读取、I/O控制

诊断I类和诊断II类从节点(参见ISO 17987-3)通过信号承载帧提供基于信号的故障信息和传感器、I/O访问。LIN主节点负责处理从节点基于信号的故障及相关的DTCs。LIN主节点将UDS请求直接提供给客户端/测试仪，并充当诊断应用层网关。UDS服务提供对LIN总线上的传感器/执行器信号的访问。

诊断III类从节点(参见ISO 17987-3)是独立的诊断实体。LIN主节点不为其诊断III类从节点的诊断能力实施诊断服务。

11.2.3.2 从节点要求

从节点通常是不涉及复杂数据通信的电子设备。此外，他们对分发诊断数据的需求很低。然而，大多数从节点应传输简单的诊断信息，如信号承载帧中的错误指示。

虽然诊断和节点配置服务使用相同的帧ID，即0x3C(主请求帧)和0x3D(从响应帧)，但配置和诊断使用不同的服务。节点配置可以由主节点独立执行，而诊断服务总是根据外部或内部测试设备的请求路由。两种情

况都使用相同的节点地址(NAD)和传输协议，但配置总是通过单帧(SF)执行。只有从节点具有NAD。NAD也用作诊断从响应帧中的源地址。

注：物理节点和逻辑节点之间存在一对多映射，并且使用NAD进行寻址。

11.2.4 基于信号的诊断

11.2.4.1 从节点实现

基于信号的诊断由从节点(诊断 I 和 II 类)实现，这些从节点不实现故障存储器和诊断协议，而是直接从外部测试工具访问该故障存储器。

有两种类型的故障传输通过信号承载帧。

- a) 类型 1 故障信息由从节点定期传输并编码到现有信号中(例如，用于指示具体故障情况的信号范围的上限)。具体的和由车辆制造商定义的类型 1 故障使用案例，不属于本国际标准的一部分。
- b) 对于不产生周期性传输信号的组件，类型 2 故障信息不会周期性传输(例如，从节点内部故障)。额外的基于信号的故障传输应被用于类型 2 故障(例如，如果一个从节点能够局部检测到没有通过信号承载帧中的相关信号传输的故障)。

每个从节点应通过信号承载帧将从节点监控的故障状态信息传输给主节点。状态信息应包含从节点组件的当前故障状态。信号应支持表2中定义的状态。

表11 基于信号的故障状态

描述
无可用测试结果， 默认值， 初始化值
测试结果：失败
测试结果：通过

如果从节点实现多个独立功能，则可以为每个功能分配状态信号。在这种情况下，应用程序只能禁用失败的功能。

在LIN描述文件(LDF)的状态管理中设置故障状态信号。

11.2.4.2 主节点实现

应为每个导致主节点中单独记录 DTC 的故障分配故障状态信号。

此信息用于指示主节点应用程序的一个组件出现故障，然后主节点可以存储关联的 DTC。每个可更换部件应有一个信号，以简化车辆的维修和保养。

11.2.5 支持工具套件

工具套件支持LIN节点中诊断服务的实现，使用ISO 17987-2中定义的传输协议和网络层服务，可以由工具套件支持，与ISO 17987-2、ISO 17987-3和ISO 17987-5中规定的要求兼容。

ISO 17987-5中规定了支持“c”编程语言的标准化API。

11.3 LIN 上的统一诊断服务实现

11.3.1 UDSSonLIN 服务概述

表3的目的是参考所有的统一诊断服务，因为它们适用于UDSSonLIN的实现。该表包含所有适用服务的总和。某些应用程序使用本章实现UDSSonLIN，可能会限制可用服务的数量，并在某些应用程序区域/诊断会话(默认会话、编程会话等)中对它们进行分类。由于数据链路层应用，所有诊断服务的数据长度都受到限制(参见8.1)。

注：ISO 17987系列支持LIN从节点的不同诊断类。指定在LIN从节点中实施第5章中哪些诊断服务是车辆制造商的责任。

图29 可适用的第5章统一诊断服务和数据范围概述

诊断服务名称 (参见第5章)	说明	本国际标准中的参考
诊断和通信管理功能单元		
诊断会话控制	LIN—无特定要求	—
ECU 重启	LIN—无特定要求	—
安全访问	LIN—无特定要求	—
通信控制	LIN—有特定要求	参见 7.2.1
测试仪在线	LIN—无特定要求	—
基于事件响应	LIN—有特定要求	参见 7.2.2
控制 DTC 设置	LIN—无特定要求	—
数据传输控制单元		
按标识符读取数据	LIN—无特定要求	—
按标识符写数据	LIN—无特定要求	—
存储数据传输功能单元		
清除诊断信息	LIN—无特定要求	—
读取 DTC 信息	LIN—无特定要求	—
输入/输出控制功能单元		
按标识符输入输出控制	LIN—无特定要求	—
常规功能单元的远程激活		
例程控制	LIN—无特定要求	—
上传/下载功能单元		
请求下载	LIN—无特定要求	—
请求上传	LIN—无特定要求	—
传输数据	LIN—无特定要求	—
请求传输终止	LIN—无特定要求	—
请求文件传输	LIN—无特定要求	—

注：在表3中未列出的服务不支持。

11.3.2 诊断和通信控制功能单元

11.3.2.1 通信控制(0x28)服务

表4定义了特定于LIN的通信控制(0x28)服务要求。

表12 LIN 特定通信控制(0x28)服务要求

通信控制要求	要求描述
主节点_CC.Req1	LIN 主节点应实现第5章中规定的通信控制服务，以允许 LIN 簇“正常通信”报文类型的特定激活和取消激活(参见表5)。 注启用和禁用的仅用于诊断模式的命令是诊断服务—通信控制(参见第5章)。该服务的参数为通信类型，设置为“正常通信报文”。

表5定义了通信控制(0x28)服务和相关调度。

表13 通信控制(0x28)服务和相关调度

正常通信	主动调度
启用	交错诊断 LIN 簇的请求通过诊断通信控制服务的参数“节点标识号”有效调度。
禁用	仅用于诊断 LIN 簇的请求通过诊断通信控制服务的参数“节点标识号”有效调度。

11.3.2.2 基于事件响应(0x86)服务

表6定义了LIN特定诊断服务要求。

表14 LIN 特定基于事件响应(0x86)服务要求

基于事件响应要求	要求描述
主节点_CC.Req1	主节点应提供一个响应报头，以使从节点能够在发生指定事件时提供响应服务(参见第5章)。 在从节点的诊断需要基于事件响应服务时，LIN的规划就需要考虑到这一点。

11.4 应用层要求

11.4.1 应用层服务

本章使用第5章中定义的应用层服务，用于基于客户端-服务器的系统，以执行例如测试、检查、监控、诊断或者车载服务器编程的功能。

UDSonLIN报文的报文长度在ISO 17987-2中规定。

报文缓冲区由会话层控制，由数据链路层在检测到报文开始且报文长度可用时请求。

主节点应根据指定的数据长度与其所有从节点同步接收缓冲区的最大数量，以存储所有传输的数据。

某些诊断服务例如读取DTC信息可能超过报文长度限制(取决于要报告的DTC数量)。在这种情况下，否定响应处理适用于第5章中规定的每个相关服务。

11.4.2 应用层协议

本章使用第5章中定义的应用层协议。

11.4.3 应用层时序

11.4.3.1 通用

子条款指定应用程序和会话层时序参数，以及这些参数如何应用于客户端和服务器。

11.4.3.2 应用层时序参数值

LIN主节点和从节点应用程序诊断报文交互的时序要求应根据ISO 17987-2实施。

P2_{服务器_最大}应根据第6章在会话层处理。

由于排放相关系统的时间性能要求，不建议将排放相关的主要ECU连接到LIN簇，如果LIN从节点对排放相关的通信有任何影响，即为排放相关的通信提供排放相关的数据。

车辆制造商有责任确保在客户端不需要响应报文的情况下[抑制肯定响应消息指示位=真('1')]，服务器可能需要超过P2_{服务器}时间来处理请求消息，客户端应在与后续请求之间插入足够的时间。

根据实现的选择，有两种可能性：

——镜像到LIN的请求可以在主节点接收到第一帧后立即发送，并且

——主节点收到所有请求并将其转发给从节点

主节点有责任使用专用的调度表或时序表向从节点提供所需的数据并接收从节点的响应。

注：在LIN上提供第一帧的时间取决于时序表(使用0x3C帧标识符)。

注：由于ISO 17987-4中的传输协议定义，LIN上的连续帧数量可能不同于客户端/主端的连续帧数量(± 1)。

图2通过DoCAN主干总线展示UDSonLIN的时序图。

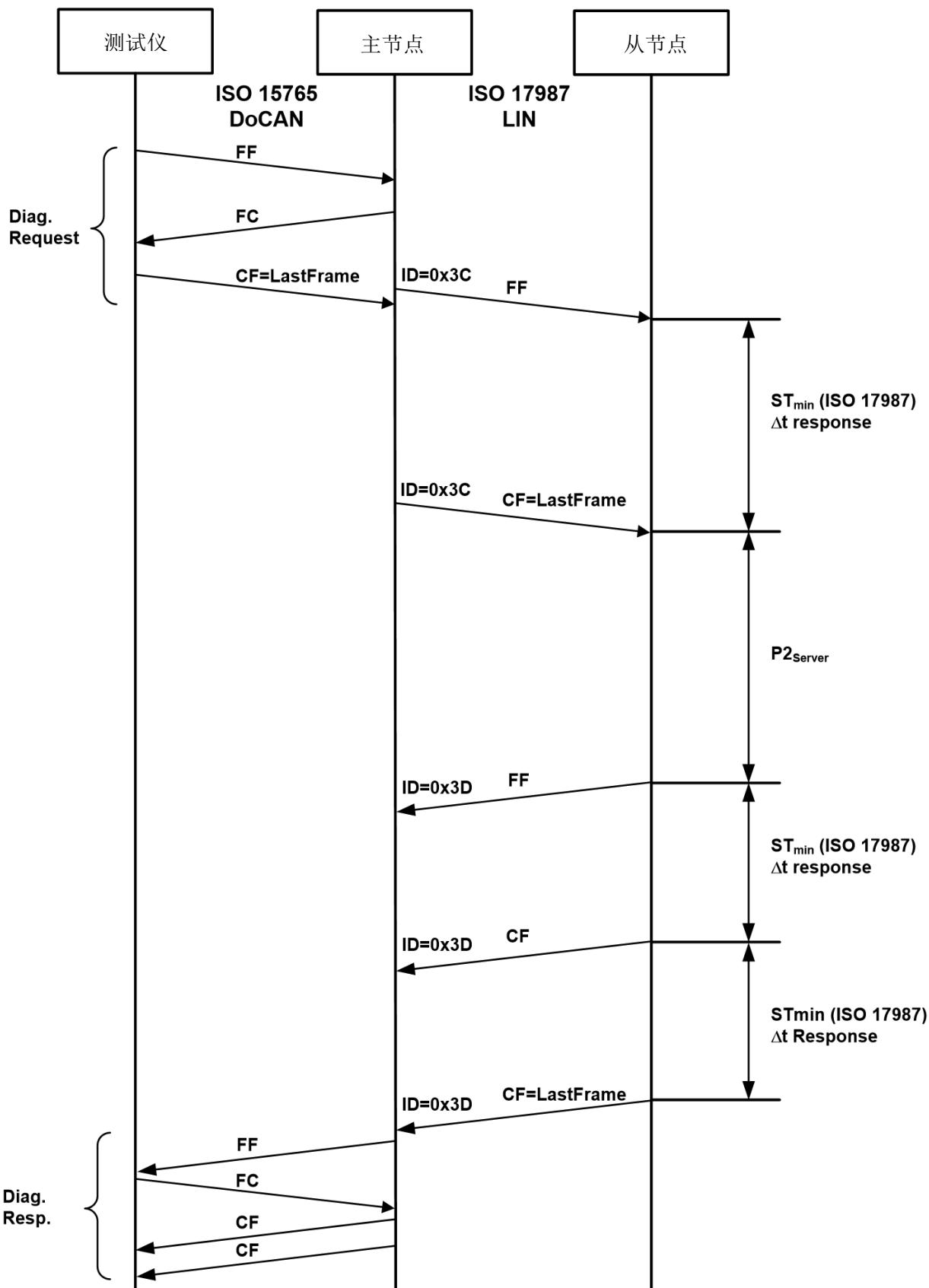


图30 通过 DoCAN 主干总线的 UDSonLIN 时序图

11.5 表示层要求

表示层要求由车辆制造商负责。

11.6 会话层要求

在第6章中指定了会话层要求。

11.7 传输/网络层适配接口

11.7.1 通用信息

本章利用第6章中定义的网络层服务来传输和接收诊断报文。该条款定义了数据链路独立传输/网络层协议数据单元(T_PDU)到LIN数据链路特定网络层协议数据单元(N_PDU)的映射。

完整的传输协议层实现需要支持基于多帧的报文传输(参见ISO 17987-2)。

注：传输/网络层服务用于执行应用层和诊断会话管理定时。

11.7.2 LIN 传输/网络层适配接口

11.7.2.1 数据链路独立服务原语到 LIN 数据链路相关服务原语的映射

表7指定了ISO 17987-2和第6章中定义的用于传输和接收诊断报文的会话层服务之间的映射接口。

表15 T_PDU 服务原语到网络 N_PDU 服务原语的映射

会话传输/网络层服务原语（独立于第6章的数据链路）	LIN 网络层服务原语（依赖于 ISO 17987-2 数据链路）
T_数据.指示	N_US 数据.指示
T_数据 SOM.指示	N_US 数据 FF.指示
T_数据.确认	N_US 数据.确认
T_数据.请求	N_US 数据.请求

N_US数据FF.指示服务原语映射到T_数据SOM.指示通常表示开始接收LIN数据链路上的多帧报文。

11.7.2.2 将 T_PDU 映射到 N_PDU 以进行报文传输

用于发起诊断服务请求/响应而定义的应用层协议数据单元的参数根据表8映射到网络层协议数据单元的参数用于在客户端/服务器中的传输报文。

表16 T_PDU 参数到网络 N_PDU 参数的映射

T_PDU 参数 (独立于第6章的数据链路)	N_PDU 参数 (依赖于 ISO 17987-2 的 LIN 数据链路)
T_报文类型	N_报文类型
T_源地址	不适用
T_目标地址	N_目标地址
T_目标地址类型	N_目标地址类型
T_AET_扩展地址	不适用
T_数据[]	<报文数据>
T_长度	<数据长度>
T_结果	<N_结果>

网络层确认成功传输的报文应在发送请求/响应报文(ECU重启、波特率变化等)后，立即执行(N_USData.con)转发到应用层，因为在应用层中启动这些操作需要它。

在网络层和OSI更高层之间的映射的地址不必和数据链路层的编译的地址的值一样，因这一地址取决于具体的实施方案。

在ISO 17987-2中定义的映射。

11.8 网络层诊断实施要求

11.8.1 LIN 从节点网络层要求

本章根据诊断类规定了LIN从节点网络层要求(参见表9)。

表17 LIN 从节点根据诊断类的网络层要求

诊断类要求	要求描述
从节点_诊断类 I .要求 1	没有为符合诊断 I 类的 LIN 从节点定义特定的网络层要求。
从节点_诊断类 II .要求 1	每个诊断 II 类 LIN 从节点应执行 ISO 17987-2 中规定的网络层要求。
从节点_诊断类 II .要求 2	在系统设计期间，应为每个诊断 II 类 LIN 从节点分配一个固定节点地址(NAD)。 注对于在网络通信启动期间使用动态 NAD 分配的 LIN 簇，这意味着 NAD 分配完成后，

	在系统设计期间,所有LIN从节点都按照指定分配了它们的NADs。
从节点_诊断类II.要求3	每个诊断II类的LIN从节点应忽略在LIN簇上广播(功能)寻址NAD 0x7E的诊断通信。
从节点_诊断类III.要求1	每个诊断III类的LIN从节点应执行ISO 17987-2中规定的网络层要求。
从节点_诊断类III.要求2	在系统设计期间,应为每个诊断III类LIN从节点分配一个固定节点地址(NAD)。 注:对于在网络通信启动期间使用动态NAD分配的LIN簇,这意味着NAD分配完成后,在系统设计期间,所有LIN从节点都按照指定分配了它们的NADs。
从节点_诊断类III.要求3	每个诊断III类的LIN从节点应支持在LIN簇上广播(功能)寻址NAD 0x7E的诊断通信。

在连接到LIN簇的多个UDS LIN从属服务器的情况下,由于通过帧标识符3C₁₆(60_d主请求帧)限制LIN上的诊断通信,因此不可能检索功能寻址请求的响应。

在连接到LIN簇的多个UDS LIN从属服务器的情况下,由于通过帧标识符3D₁₆(61_d从属响应帧)限制LIN上的诊断通信,因此无法发送并行物理请求。

11.8.2 LIN 主节点网络层要求

11.8.2.1 网络地址要求

本章规定了主节点网络层的要求(参见表10)。

表18 LIN 主节点网络层要求

诊断类要求	要求描述
主节点_诊断类III.要求1	每个诊断III类LIN主节点应执行ISO 17987-2中规定的网络层要求。
主节点_诊断类III.要求2	如果在LIN簇上使用动态NAD分配,则LIN主节点应确保在通信启动后,根据表9从节点_II.要求2和从节点_III.要求2,所有LIN从节点都有指定的NADs。完成NAD分配所需的启动配置时间应记录在主节点的诊断规范中。 注:这意味着外部诊断测试设备和LIN从节点之间的诊断通信可能是不可能的,直到LIN主节点完成了NAD分配序列。
主节点_诊断类III.要求3	连接到LIN主节点的LIN簇在诊断服务通信控制时使用被分配的子网号码0x1。

11.8.2.2 功能寻址的使用

如果客户端不了解LIN架构,则外部测试设备(客户端到LIN主节点)使用功能寻址。

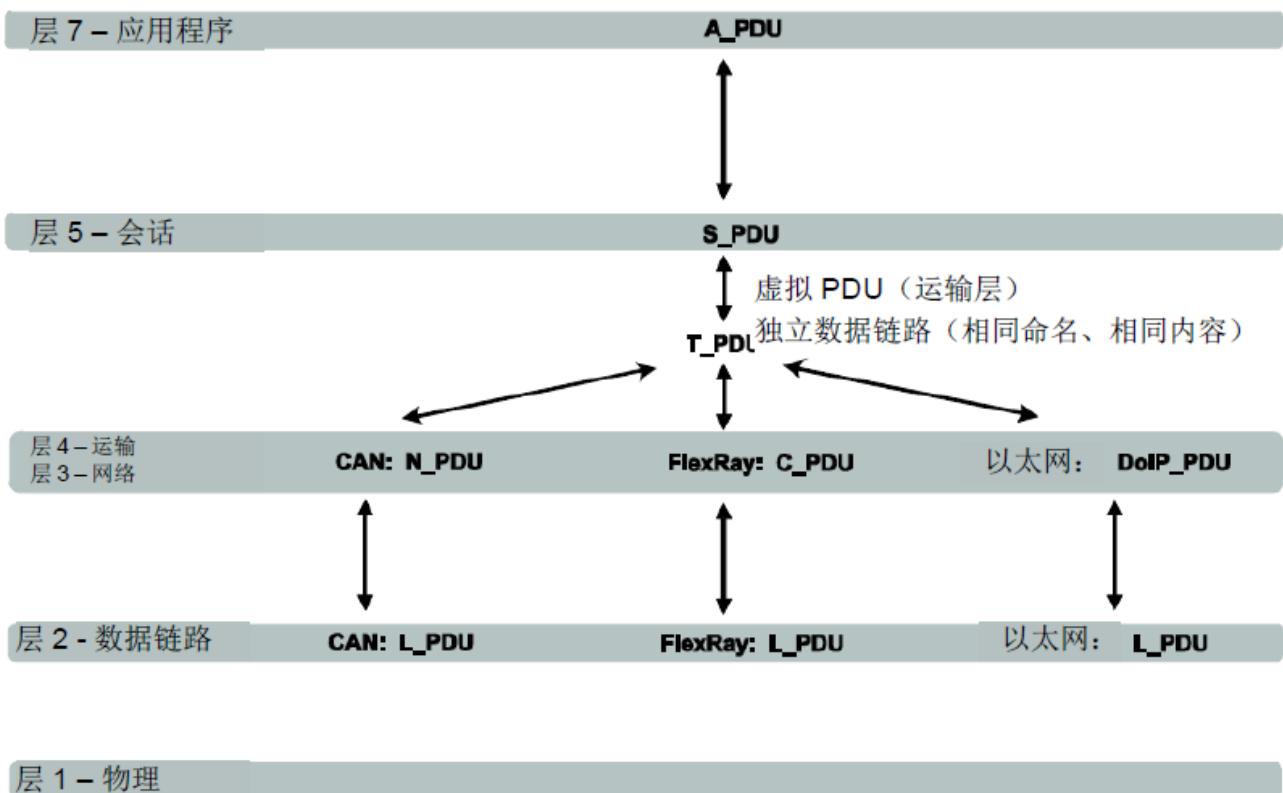
11.9 数据链路层诊断实施要求

本章利用ISO17987-3中定义的数据链路层规范来传输和接收诊断报文。

附录 A
(规范性附录)

通信协议的 T_PDU 虚拟 PDU 接口

图A.1为统一诊断服务PDU和任一通信协议之间的T_PDU（虚拟PDU）接口。



图A.1 任一通信协议的 T_PDU 虚拟 PDU 接口

附录 B

(资料性附录)

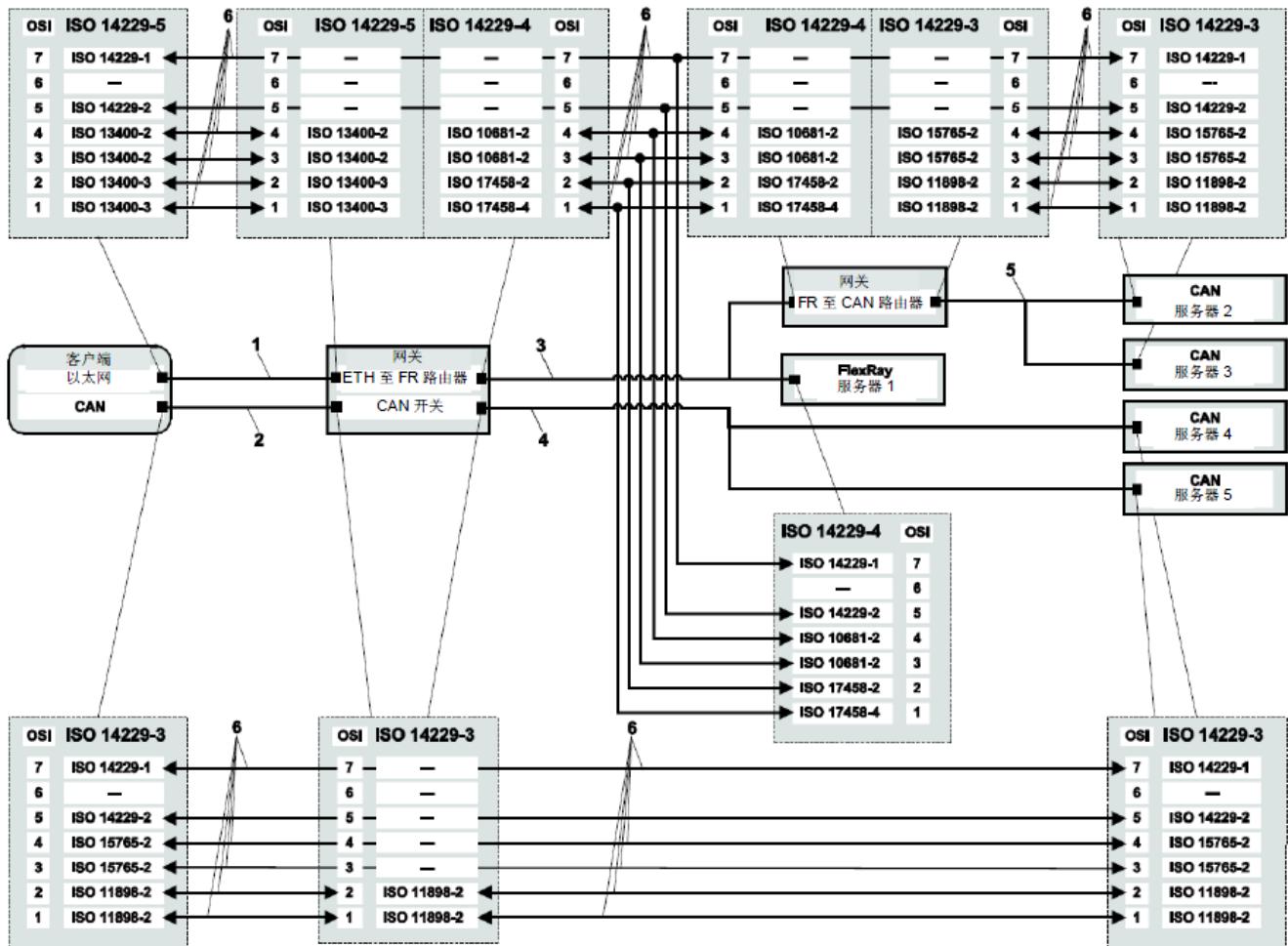
车辆诊断 OSI 层结构示例

B.1 车辆诊断OSI层网关示例

图B.1为带不同网络技术和两个网关实例的车辆诊断网络结构的示例。网关装置中启用的“以太网至FlexRay路由器”是在OSI层3和4传输PDU的网络设备。“CAN开关”是在OSI层2传输PDU的网络设备。

框架（虚线）中提供了须由相应接口支持的不同协议的ISO/OSI层模型。

数据链路密钥1和2均可用于访问车辆诊断数据。数据链路密钥3至5是车载通信网络的示例，其中数据链路密钥5位于第二个网关（FlexRay至CAN路由器）的后面。



图例

- 1 诊断以太网(ETH);
 - 2 诊断CAN;
 - 3 FlexRay通信系统(FR);
 - 4 CAN;
 - 5 CAN;
 - 6 逻辑连接;
- 不可用

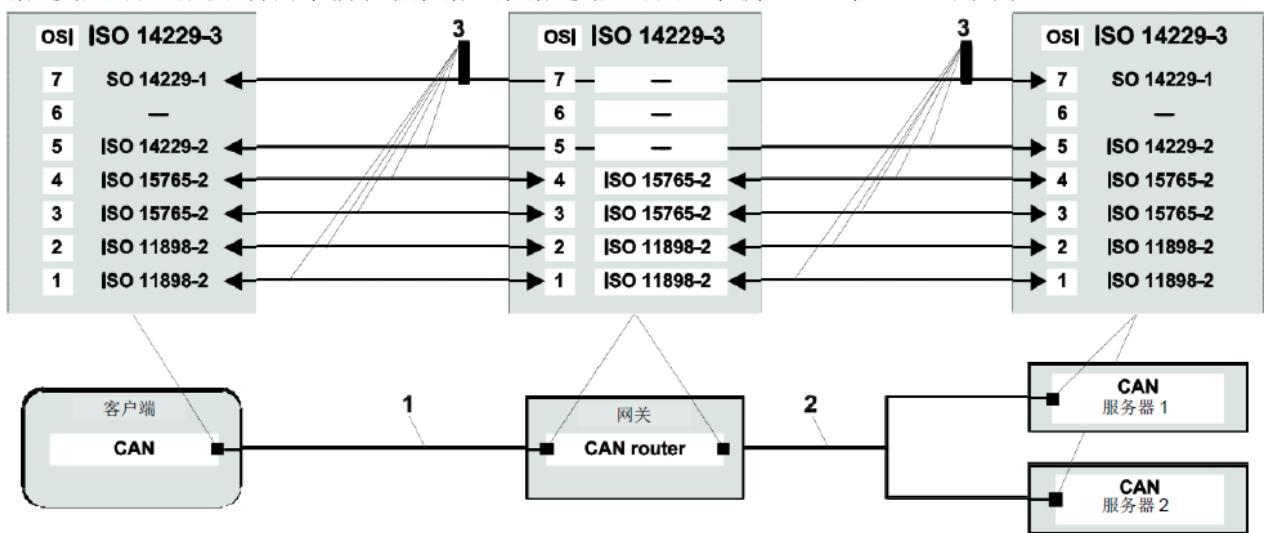
图 B.1 车辆诊断 OSI 层网关示例

B.2 车辆诊断OSI层CAN路由器示例

图B.2为网关中启用带“CAN路由器”的车辆诊断网络结构示例。网关装置中启用的“CAN路由器”是在OSI层3和4传输PDU的网络设备。

框架（虚线）中提供了须由相应接口支持的不同协议的ISO/OSI层模型。

数据链路密钥1可用于访问车辆诊断数据。数据链路密钥2是车载CAN通信网络的示例。



图例

- 1 诊断 CAN
- 2 CAN
- 3 逻辑链接
- 不适用

图B.2 车辆诊断OSI层CAN路由器示例

B.3 车辆诊断OSI层CAN开关示例

图B.3为网关中启用带“CAN开关”的车辆诊断网络结构。网关装置中启用的“CAN开关”是在OSI层2传输PDU的网络设备。

框架（虚线）中提供了须由相应接口支持的不同协议的ISO/OSI层模型。

数据链路密钥1可用于访问车辆诊断数据。数据链路密钥2是车载CAN通信网络的示例。

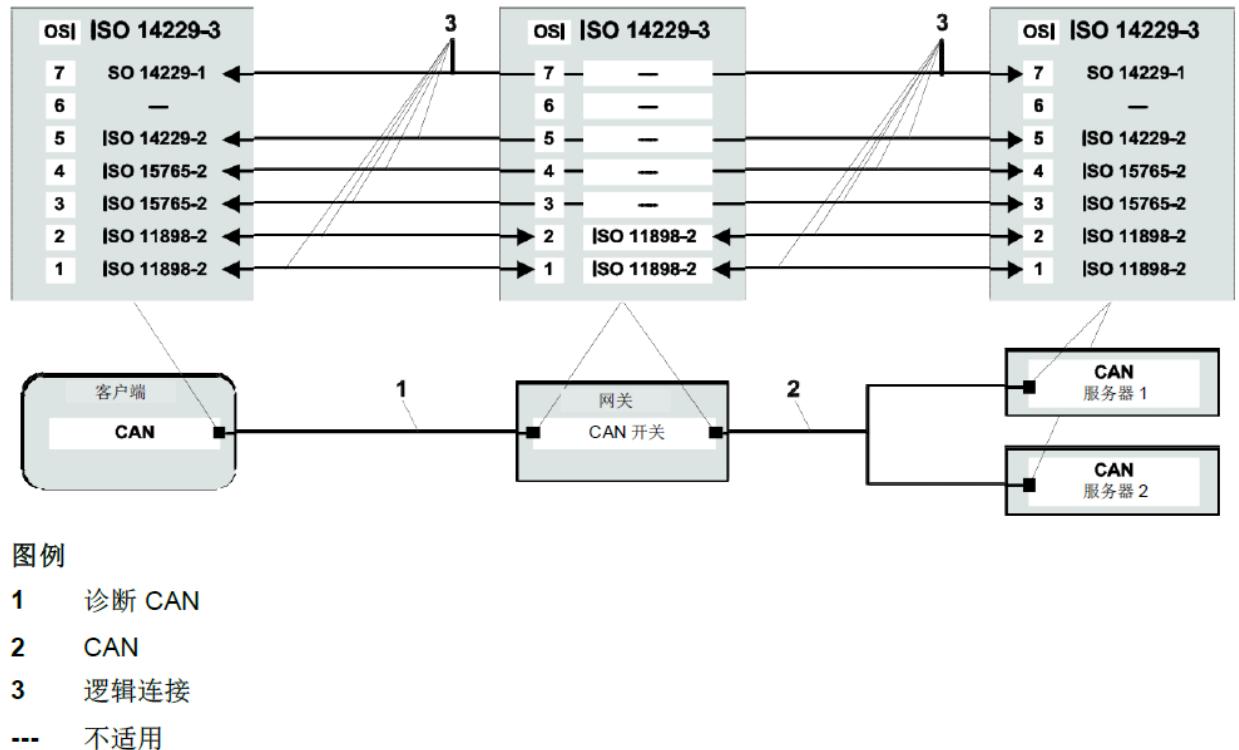


图 B.3 车辆诊断 OSI 层 CAN 开关示例

参 考 文 献

- [1] IEEE 802.3 IEEE Standard for Information technology — Telecommunications and information exchange between systems – Local and metropolitan area networks — Specific requirements — Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications
- [2] ISO/IEC 10731 Information technology—Open Systems Interconnection—Basic Reference Model—Conventions for the definition of OSI services
- [3] ISO 11898-1 Road vehicles—Controller area network(CAN)—Part 1:Data link layer and physical signalling
- [4] ISO 11898-2 Road vehicles—Controller area network(CAN)—Part 2:High-speed medium access unit
- [5] ISO 11898-3 Road vehicles—Controller area network(CAN)—Part 3:Low-speed, fault-tolerant
- [6] ISO 11898-5 Road vehicles — Controller area network (CAN) — Part 5: High-speed medium access unit with low-power mode
- [7] ISO/IEC 7498-1 Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model
- [8] ISO 10681-1 Road vehicles—Communication on FlexRay—Part 1: General information and use case definition
- [9] ISO 10681-2 Road vehicles—Communication on FlexRay—Part 2: Communication layer services
- [10] ISO 13400(all parts) Road vehicles—Communication between test equipment and vehicles over Internet Protocol
- [12] ISO 14230-1 Road vehicles—Diagnostic communication over K-Line (DOK-Line)—Part 1: Physical layer
- [13] ISO 14230-2 Road vehicles—Diagnostic communication over K-Line (DOK-Line) —Part 2: Data link layer
- [14] ISO 15031-5 Road vehicles—Communication between vehicle and external equipment for emissionsrelated diagnostics — Part 5: Emissions-related diagnostic services
- [15] ISO 15031-6 Road vehicles — Communication between vehicle and external equipment for emissionsrelated diagnostics — Part 6: Diagnostic trouble code definitions
- [16] ISO 15765-2 Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2: Transport protocol and network layer services
- [17] ISO 15765-4 Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) —Part 4: Requirements for emissions-related systems
- [18] ISO 17458-2 Road vehicles—FlexRay communications system—Part 2:Data link layer specification
- [19] ISO 17458-4 Road vehicles—FlexRay communications system—Part 4: Electrical physical layer specification
- [20]ISO 17987 (all parts) Road vehicles — Local Interconnect Network (LIN)
- [21]SAE J1930-DA Digital Annex of Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviations, and Acronyms Web Tool Spreadsheet

- [22] SAE J1939-73:2010 Recommended Practice for a Serial Control and Communication Vehicle Network—Application layer — Diagnostics
- [23] SAE J1939 Companion Spreadsheet (CS1939)
- [24] SAE J1979-DA Digital Annex of E/E Diagnostic Test Modes
- [25] SAE J2012-DA Digital Annex of Diagnostic Trouble Code Definitions and Failure Type Byte Definitions
- [26] ISO 27145 (all parts) Road vehicles—Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements