

МЕЖДУНАРОДНЫЙ
СТАНДАРТ

ИСО
14229-2

Первое
издание 2013-02-15

Автотранспортные средства — Единые
диагностические службы (УДС) —

Часть 2:

Сервисы сеансового уровня

Транспортные средства — Службы диагностических комплексов (SDU) —

Часть 2: Последовательность предоставления услуг



Номер ссылки
ИСО 14229-2:2013(Е)

© ИСО 2013

ИСО 14229-2:2013(E)



ДОКУМЕНТ, ЗАЩИЩЕННЫЙ АВТОРСКИМ ПРАВОМ

© ИСО 2013

Все права защищены. Если не указано иное, никакая часть данной публикации не может быть воспроизведена или использована в любой форме или любыми средствами, электронными или механическими, включая фотокопирование и микрофильмирование, без письменного разрешения либо от ИСО по указанному ниже адресу, либо от органа-члена ИСО в стране запрашивающей.

Бюро авторских прав
ISO Case postale 56 • CH-1211 Женева 20 Тел. +
41 22 749 01 11 Факс + 41 22 749 09 47
Электронная почта copyright@iso.org Web
www.iso.org

Опубликовано в Швейцарии

Содержание

Страница

1	Сфера.....	1
2	Нормативные ссылки.....	1
3	Термины, определения и сокращения.....	1
3.1 3.2	термины	2
4	Соглашения.....	2
5	Обзор документа.....	3
6	Сервисы сеансового уровня.....	4
6,1	сведения.....	4
6,2	сервисных примитивов сеансового уровня.....	6
6,3	данных сеанса.....	7
7	Определение временного параметра.....	9
7,1	сображения по времени применения.....	9
7,2	времени приложения – defaultSession.....	10
7,3	отклика.....	15
7,4	отклика.....	16
7,5	сеанса по умолчанию.....	17
7,6 7,7	сервера.....	19
	ошибок.....	20
8	Обработка интронизации во время связи.....	21
	связью.....	21
	с сообщениями запроса клиента.....	36
	коммуникация.....	21
	функциональность.....	21
	Минимальное время между.....	21
	Физическая.....	21
Приложение А (обязательное) Интерфейс T_PDU		43
Приложение В (информативное) Примеры архитектуры уровня OSI для диагностики автомобиля.....		44
Приложение С (информативное) Примеры архитектуры уровня OSI для диагностики автомобиля.....		45
Приложение D (информативное) Примеры архитектуры уровня OSI для диагностики автомобиля.....		46
Библиография.....		47

ИСО 14229-2:2013(E)

Предисловие

ISO (Международная организация по стандартизации) — всемирная федерация национальных органов по стандартизации (органов-членов ISO). Работа по подготовке международных стандартов обычно осуществляется техническими комитетами ИСО. Каждый орган-член, заинтересованный в предмете, для которого был создан технический комитет, имеет право быть представленным в этом комитете. Международные организации, правительственные и неправительственные, взаимодействуют с ИСО также принимают участие в работе. ISO тесно сотрудничает с Международной электротехнической комиссией (IEC) по всем вопросам электротехнической стандартизации.

Международные стандарты разрабатываются в соответствии с правилами, изложенными в Директивах ИСО/МЭК, Часть 2.

Основной задачей технических комитетов является подготовка международных стандартов. Проекты международных стандартов, принятые техническими комитетами, рассылаются организациям-членам для голосования. Публикация в качестве международного стандарта требует одобрения не менее чем 75 % организаций-членов, принимающих участие в голосовании.

Обращаем внимание на возможность того, что некоторые элементы этого документа могут быть предметом патентных прав. ISO не несет ответственности за идентификацию каких-либо или всех таких патентных прав.

ISO 14229-2 был подготовлен Техническим комитетом ISO/TC 22, Дорожные транспортные средства, Подкомитетом SC 3, Электрическое и электронное оборудование.

ISO 14229 состоит из следующих частей под общим заголовком Дорожные транспортные средства. Унифицированные диагностические услуги. (УДС):

- Часть 1. Спецификация и требования
- Часть 2. Услуги сеансового уровня
- Часть 3: Унифицированные диагностические услуги по реализации CAN (UDSonCAN)
- Часть 4: Унифицированные диагностические сервисы для реализации FlexRay (UDSonFR)
- Часть 5: Унифицированные диагностические услуги по реализации интернет-протокола (UDSonIP)
- Часть 6: Унифицированные диагностические услуги при реализации K-Line (UDSonK-Line)

Готовится следующая часть:

- Часть 7: Унифицированные диагностические услуги для реализации локальной межсетевой сети (UDSonLIN)

Названия будущих частей будут оформлены следующим образом:

- Часть n: Унифицированные диагностические услуги по ... внедрению (UDSON...)

Вступление

Стандарт ИСО 14229 был разработан для определения общих требований к диагностическим системам, которые не зависят от особенностей последовательного канала передачи данных.

Для достижения этого ИСО 14229 основан на базовой эталонной модели взаимодействия открытых систем (OSI) в соответствии с ИСО 7498-1 и ИСО/IEC 10731, которая структурирует системы связи на семь уровней.

При сопоставлении на этой модели сервисы, используемые диагностическим тестером (клиентом) и электронным блоком управления (ЭБУ, сервером), разбиваются на следующие слои в соответствии с таблицей 1:

- прикладной уровень (уровень 7), унифицированные диагностические услуги, указанные в ИСО 14229-1, ИСО 14229-3 UDSONCAN, ИСО 14229-4 UDSONFR, ИСО 14229-5 UDSONIP, ИСО 14229-6 UDSONK-Line, ИСО 14229-7 UDSONLIN, дополнительные стандарты и ИСО 27145-3 WWH-OBD.
- Уровень представления (уровень 6), зависящий от производителя транспортного средства, ИСО 27145-2 WWH-OBD.
- услуги сеансового уровня (уровень 5), указанные в данной части ИСО 14229.
- услуги транспортного уровня (уровень 4), указанные в ИСО 15765-2 DoCAN, ИСО 10681-2 Связь на FlexRay, ИСО 13400-2 DoIP, ИСО 27145-4 WWH-OBD.
- услуги сетевого уровня (уровень 3), указанные в ИСО 15765-2 DoCAN, ИСО 10681-2 Связь на FlexRay, ИСО 13400-2 DoIP, ИСО 27145-4 WWH-OBD.
- уровень канала передачи данных (уровень 2), определенный в ИСО 11898-1, ИСО 11898-2, ИСО 17458-2, ИСО 13400-3, IEEE 802.3, ISO 14230-2 и другие стандарты, ИСО 27145-4 WWH-OBD.
- физический уровень (уровень 1), определенный в ИСО 11898-1, ИСО 11898-2, ИСО 17458-4, ИСО 13400-3, IEEE 802.3, ISO 14230-1, другие стандарты, ИСО 27145-4 WWH-OBD.

Таблица 1 — Пример спецификации диагностики/программирования, применимых к уровням OSI

Применимость	ОС И семейный слой	Расширенные услуги и диагностики					WWH-БД
Семейство в соответствии с ИСО/МЭК 7498-1 и ИСО/МЭК 10731	Приложение (уровень 7)	ИСО 14229-1, ИСО 14229-3 УДСОНКАН, ИСО 14229-4 УДСОНФР, ИСО 14229-5 UDSONIP, ИСО 14229-6 UDSONK-Line, ИСО 14229-7 UDSONLIN, дополнительные стандарты					ИСО 27145-3
	Презентация (слой 6)	зависит от производителя автомобиля					ИСО 27145-2
	Сеанс (уровень 5)	ИСО 14229-2					
	Транспорт (уровень 4)	ИСО 15765-2	ИСО 10681-2	ИСО 13400-2	Неприменимый	дополнительные стандарты	ИСО 27145-4
	Сеть (слой 3)					дополнительные стандарты	
	Канал передачи данных (уровень 2)	ИСО 11898-1, ИСО 11898-2	ИСО 17458-2, ИСО 17458-4	ИСО 13400-3, ИИЭР 802.3	ИСО 14230-2	дополнительные стандарты	
	Физический (уровень 1)				ИСО 14230-1	дополнительные стандарты	

ИСО 14229-2:2013(Е)

Автотранспортные средства. Единые диагностические службы (УДС) — Часть 2: Сервисные сервисы уровня

1 Область применения

Эта часть ИСО 14229 определяет независимые от канала передачи данных требования услуг сервисного уровня.

Эта часть ИСО 14229 определяет общие службы сервисного уровня для обеспечения независимости между унифицированными диагностическими службами (ISO 14229-1) и всеми транспортными протоколами и службами сетевого уровня (например, ISO 15765-2 DoCAN, ISO 10681-2 Связь на FlexRay, ISO 13400 DoIP), ISO 14230-2 DoK-Line и др.)

Эта часть ИСО 14229 определяет общий интерфейс сервисных примитивов между уровнем 4 OSI (транспорт) и уровнем 5 (сервис) через так называемые сервисные примитивы запроса/подтверждения/индикации. Этот интерфейс позволяет беспреткливо внедрять унифицированные диагностические службы (UDS) ISO 14229-1 с любым протоколом связи под названием «DoXYZ / CoXYZ», например ISO 15765 DoCAN — диагностическая связь через сеть контроллеров, ISO 13400 DoIP, связь ISO 10681 через FlexRay, ISO 14230. DoK-Line.

ISO 15031 (OBD, с вьезами) и ISO 27145 (WWH-OBD) поддерживают стандартизированный интерфейс сервисного примитива.

2 Нормативные ссылки

Для применения настоящего документа необходимы следующие ссылочные документы. Для датированных ссылок применяется только итируемое издание. Для недатированных ссылок применяется последнее издание ссылочного документа (включая любые поправки).

ISO 14229-1, Транспорт дорожный. Унифицированные диагностические услуги (UDS). Часть 1. Спецификация и требования

3 Термины, определения и сокращения

3.1 Термины и определения

Для целей настоящего документа применяются следующие термины и определения.

3.1.1

шлюз.

Сетевое устройство, которое передает PDU на разные уровни OSI.

ПРИМЕЧАНИЕ Сетевое устройство, которое обеспечивает связь между сетями модулей управления, которые используют разные протоколы связи, разные скорости связи и т. д. Это включает, помимо прочего, функции шлюза, такие как мост, коммутатор, маршрутизатор или маршрутизация прикладного уровня.

ИС О 14229-2:2013(Е)

3.1.2

маршрутизатор: Сетевое устройство, которое передает PDU на уровне 3 и 4 модели OSI.

3.1.3

коммутатор: Сетевое устройство, которое передает PDU на уровне 2 модели OSI.

3.2 Сокращенные термины

CDD	общий словарь данных
CMD	общий словарь сообщений
ДСК	контроль сеанса диагностики
ЭБУ	электронный блок управления
ОСИ	взаимодействие открытых систем
S_AE	расширение адреса сеансового уровня
S_SA	исходный адрес сеансового уровня
S_Данные	имя службы передачи данных сеансового уровня
СИ	идентификатор услуг
СОМ	начало сообщения
S_Мтип	тип сообщения сеансового уровня
S_PDU	блок данных протокола сеансового уровня
S_TA	целевой адрес сеансового уровня
S_TAtype	тип целевого адреса сеансового уровня

4 Конвенции

Эта часть ISO 14229 руководствуется соглашениями, обещаемыми в соглашениях об обслуживании OSI (ISO 10731:1994), поскольку они применяются как диагностическим услугам. Эти соглашения определяют взаимодействие между пользователем службы и поставщиком службы. Информация передается между пользователем службы и поставщиком службы с помощью примитивов службы, которые могут передавать параметры.

5 Обзор документа

На рис. 1 показаны реализации ISO 14229-2 для различных протоколов.

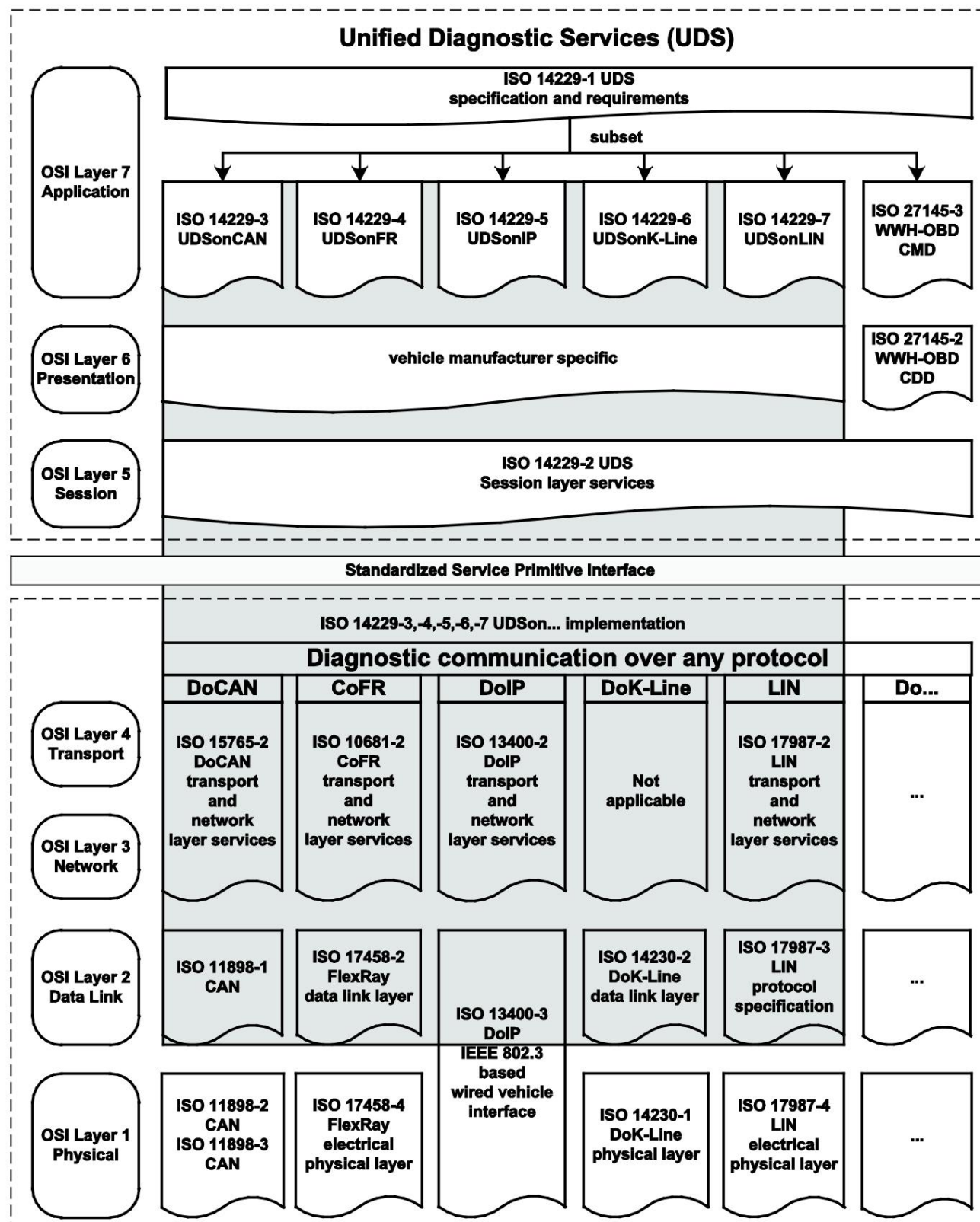


Рисунок 1 — Реализация ссылки на документ UDS в соответствии с моделью OSI

ИС О 14229-2:2013(Е)

6 сервисов с еанс овог оу уровня

6.1 Общие

Сервисный интерфейс определяет набор сервисов, необходимых для доставки функций, предлагаемых с еанс овым уровнем, т. е. передача/прием данных и установка параметров протокола.

Все сервисы с еанс овог оу уровня имеют одинаковую общую структуру. Прimitives с лужбы определяют, как пользователь с лужбы (например, диагностическое приложение) взаимодействует с поставщиком с лужбы (например, с еанс овый уровень). Для определения сервисов указываются три типа сервисных примитивов:

- примитив запроса услуг и S_Data.request, используемый более высоким прикладным уровнем для передачи управляющей информации или данных, которые должны быть переданы на с еанс овый уровень (т. е. поставщик услуг запрашивается пользователем услуг для обработки управляющей информации или для передачи данных);
- примитив индикации услуг и S_Data.indication, используемый с еанс овым уровнем для передачи информации о состоянии и полученных данных на более высокий прикладной уровень (т. е. пользователь услуг информируется поставщиком услуг о внутреннем событии с еанс овог оу уровня или запросе услуг). пользователя услуг и объекта равноправного уровня протокола);
- примитив подтверждения услуг и S_Data.confirm, используемый с еанс овым уровнем для передачи информации о состоянии прикладному уровню (т. е. пользователь услуг информируется поставщиком услуг о результате предыдущего запроса услуг и пользователя услуг);

Все сервисы с еанс овог оу уровня имеют одинаковый общий формат. Служебные примитивы записываются в виде:

```
service_name.type (параметр А,  
                  параметр В, параметр С  
                  [, параметр Х, ...])
```

Где:

- "имя_с_лужбы" — имя с лужбы (например, S_Data),
- "тип" указывает тип сервисного примитива (например, запрос, индикация, подтверждение),
- "параметр А,..." представляет собой S_PDU (блок данных протокола с еанс овог оу уровня) в виде списка значений, переданных сервисным примитивом (например, адресная информация, данные, длина, результат),
- «параметр А, параметр В, параметр С» являются обязательными параметрами, которые должны быть включены во все услуги звонки,
- «[параметр Х]» является необязательным параметром, который включается, если выполняются определенные условия

На рис. 2 показаны сервисные примитивы сеансового уровня для сообщения с одним кадром.

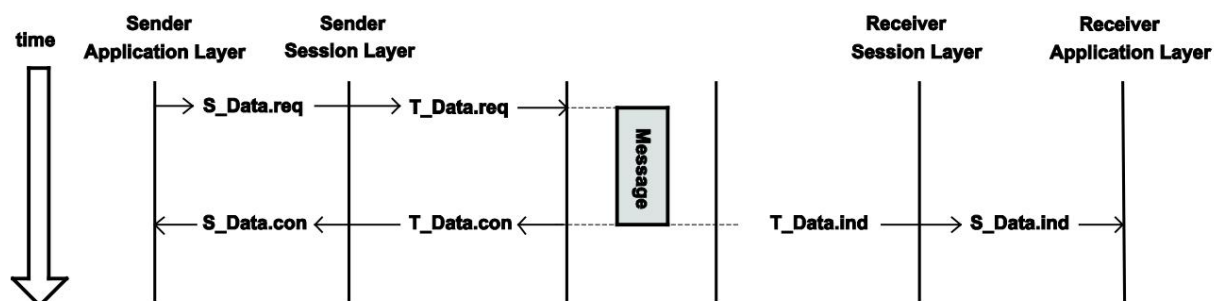


Рисунок 2 — Сервисные примитивы сеансового уровня — однокадровое сообщение

На рис. 3 показаны сервисные примитивы сеансового уровня для сообщения с несколькими кадрами, если транспортный/сетевой уровень поддерживает интерфейс **T_DataSOM.ind**.

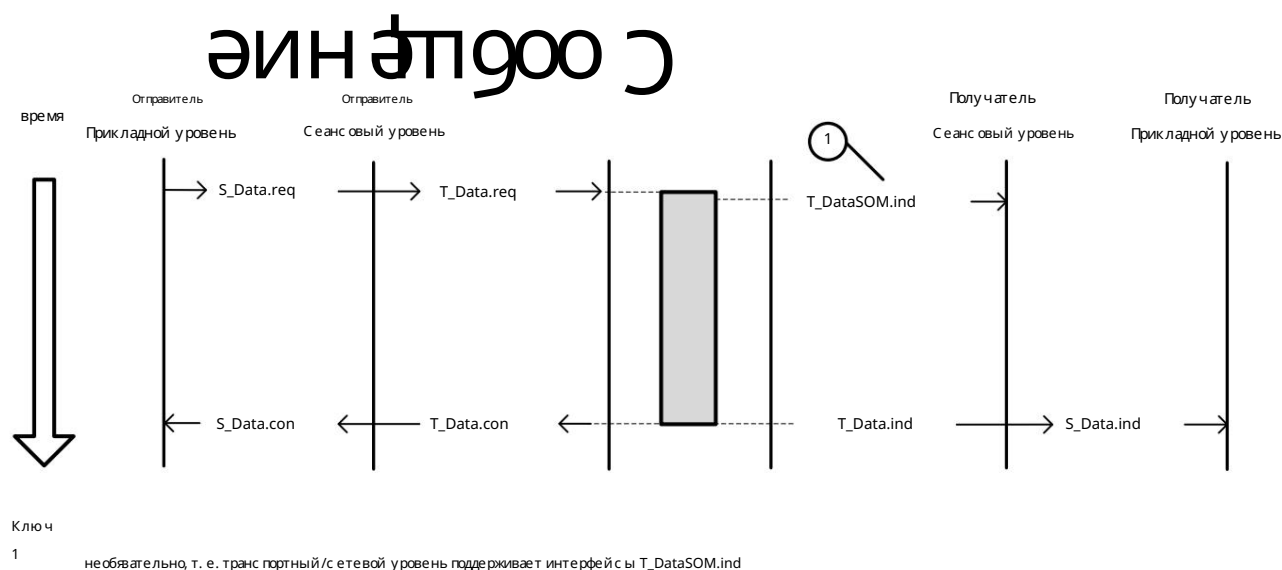


Рисунок 3 — Сервисные примитивы сеансового уровня — сообщение с несколькими кадрами

Следует различать следующие сценарии общения

а) физическое общение во время

- 1) сеанс по умолчанию и
- 2) сеанс не по умолчанию — требуется обработка сеанса;

б) функциональное общение во время

- 1) сеанс по умолчанию и
- 2) сеанс не по умолчанию — требуется обработка сеанса.

ИСО 14229-2:2013(E)

Во всех случаях должна быть рассмотрена возможность запроса расширенного окна времени ответа сервером посредством отрицательного ответного сообщения, включая отрицательный код ответа 0x78. Услуг и транспортный протокол сетевого уровня, определенные в различных стандартах ISO (например, ISO 15765-2 DoCAN или ISO 10681-2 CoFR), используют для выполнения их ронизации и управления диагностическими сеансами на клиенте и сервере.

6.2 Спецификация сервисных примитивов сеансового уровня

6.2.1 Общие положения

Чтобы описать функцию сеансового уровня, необходимо рассмотреть услуги, предоставляемые более высоким уровням, и внутренние операции сеансового уровня.

6.2.2 S_Data.request

Сервисный примитив запрашивает передачу S_Data с количеством байтов S_Length от отправителя получателю равноправных объектов, идентифицированных адресной информацией в S_SA, S_TA, S_TAtype и S_AE. Каждый раз, когда вызывается служба S_Data.request, сеансовый уровень должен синхронизировать о завершении (или сбое) передачи с сообщением пользователю службы посредством выдачи вызова службы S_Data.confirm.

S_Data.request

```
(S_Mtype,
  S_SA, S_TA,
  S_TAtype,
  [S_AE], S_Data
  [Данные №1,
   Данные №2, ..., S_Length]          Данные#n ],
```

6.2.3 S_Data.confirm

Служба S_Data.confirm выдается сеансовым уровнем. Примитив службы подтверждает завершение службы S_Data.request, идентифицированной адресной информацией об адресе в S_SA, S_TA, S_TAtype и S_AE. Параметр S_Result представляет статус запроса на обслуживание.

S_Data.confirm

```
(
  S_Мтип,
  C_CA,
  C_TA,
  S_TAтип,
  [S_AE],
  S_Результат )
```

6.2.4 S_Data.indication

Служба S_Data.indication выдается сеансовым уровнем. Служебный примитив указывает события S_Result и представляет S_Data с S_Length байтами, полученными от однорангового объекта протокола, идентифицированного адресной информацией в S_SA, S_TA, S_TAtype, на оседный верхний уровень.

Параметры S_Data и S_Length действительны только в том случае, если S_Result равен S_OK.

S_Data.indication (S_Mtype, S_SA,
S_TA, S_Tatype,
[S_AE],
S_Data
[Данные №1,
Данные №2,
... , Данные №n],
S_Длина,
S_Результат)

6.3 Спецификация блока данных сессии

6.3.1 S_Mtype, тип сообщения сессии

Тип перечисление

Ассортимент: диагностика, дистанционная диагностика

Описание:

Параметр Mtype должен использоваться для идентификации типа и диапазона параметров адресной информации, включенных в сервисный вызов. Эта часть ISO 14229 определяет диапазон из двух значений для этого параметра. Цель состоит в том, чтобы пользователи документа могли расширить диапазон значений, указав другие типы и комбинации параметров адресной информации, которые будут использоваться с протоколом транспортного уровня, указанным в этом документе. Для каждого старого нового диапазона адресной информации должно быть указано новое значение параметра Mtype, чтобы идентифицировать новую адресную информацию.

- Если S_Mtype = диагностика, то адресная информация должна состоять из параметров S_SA, S_TA и S_Tatype.

- Если S_Mtype = дистанционная диагностика, то адресная информация должна состоять из параметров S_SA, S_TA, S_Tatype и S_AE.

6.3.2 S_SA, адрес источника сессии

Тип 2-байтовое беззнаковое целое число

Диапазон: 0x0000 – 0xFFFF

Описание:

Параметр S_SA должен использоваться для кодирования отправляющего объекта протокола сессии. Параметр S_SA должен использоваться для кодирования идентификаторов клиента и сервера.

6.3.3 S_TA, целевой адрес сессии

Тип 2-байтовое беззнаковое целое число

Диапазон: 0x0000 – 0xFFFF

Описание:

Параметр S_TA должен использоваться для кодирования принимающего объекта протокола сессии. Параметр S_TA должен использоваться для кодирования идентификаторов клиента и сервера.

ИСО 14229-2:2013(Е)

6.3.4 S_Tatype, тип целевого адреса сеансового уровня

Тип перечисление

Диапазон: физический, функциональный

Описание:

Параметр S_Tatype является конфигурационным атрибутом параметра S_TA. Он должен использоваться для кодирования модели связи, используемой взаимодействующими равноправными объектами уровня связи. Определены две модели связи: связь «1 к 1», называемая физической адресацией, и связь «1 к n», называемая функциональной адресацией.

- Физическая адресация (связь 1-к-1) должна поддерживаться для всех типов сообщений сеансового уровня.

- Должна поддерживаться функциональная адресация (связь 1-к-n). Требования протокола сетевого уровня могут ограничивать использование функциональной адресации (например, SingleFrame на канальном уровне CAN).

6.3.5 S_AE, расширение адреса сеансового уровня (необязательный параметр)

Тип 2-байтовое беззнаковое целое число

Диапазон: 0x0000 - 0xFFFF

Описание:

Параметр S_AE используется для расширения диапазона адресов для больших сетей и для кодирования как отправляющих, так и принимающих объектов протокола сетевого уровня подсетей, отличных от локальной сети, в которой происходит связь. S_AE является частью адресной информации только в том случае, если Mtype установлен на удаленную диагностику.

6.3.6 S_Длина

Тип 4 байта

Диапазон: 0x0000 0000 – 0xFFFF FFFF

Описание: Этот параметр включает длину передаваемых /принимаемых данных.

6.3.7 S_Данные

Тип строка байтов

Диапазон: неограниченный

Описание: Этот параметр включает в себя все данные, которыми должны обмениваться объекты более высокого уровня.

6.3.8 S_Результат

Тип перечисление

Диапазон: C_OK, C_NOK

Описание: Этот параметр содержит состояние, связанное с результатом выполнения служб.

6.3.9 Преобразование S_PDU в T_PDU и наоборот для передачи сообщений

Параметры блока данных протокола сеансового уровня, определенные для запроса передачи запроса/ответа диагностической услуги, отображаются следующим образом на параметры блока данных протокола транспортного/сеансового уровня для передачи сообщения на клиенте/сервере.

Параметры блока данных протокола транспортного/сеансового уровня, определенные для приема сообщения, преобразуются следующим образом в параметры блока данных протокола сеансового уровня для индикации приема диагностического ответа/запроса.

Подтверждение транспортного/сеансового уровня успешной передачи сообщения (T_Data.con) передается приложению, поскольку оно необходимо приложению для запуска действий, которые должны быть выполнены сразу после передачи сообщения запроса/ответа. (например, ECUREset, BaudrateChange и т. д.).

Индикация транспортного/сеансового уровня для приема StartOfMessage T_PDU (T_DataSOM.ind) не направляется на прикладной уровень, поскольку она используется только на сеансовом уровне для выполнения ретрансляции сеансового уровня (см. пункт 7). Следовательно, отображение T_DataSOM.ind T_PDU на S_PDU не определено.

Таблица 2 определяет отображение S_PDU с сеансового уровня на T_PDU транспортного/сеансового уровня и наоборот.

Таблица 2 — Отображение S_PDU с сеансового уровня на T_PDU транспортного/сеансового уровня и наоборот

Параметр S_PDU (Протокол с сеансового уровня Блок данных)	Описание	Параметр T_PDU (Транспортный/сеансовый уровень Блок данных протокола)	Описание
S_Мтип	Сеансовый уровень Тип сообщения	T_Мтип	Транспорт/Сеть с логическим Типом сообщения
S_SA	Сеансовый уровень Источник Адрес	T_SA	Источник транспортного/сеансового уровня Адрес
S_TA	Цель сеансового уровня Адрес	T_TA	Целевой транспортный/сеансовый уровень Адрес
S_TAтип	Цель сеансового уровня Тип адреса	T_TAtype	Целевой транспортный/сеансовый уровень Тип адреса
S_AE ^a	Адрес сеансового уровня Расширение	T_AEa	Транспорт/Сеть с логическим Расширением адреса
S_Данные[1] – S_Данные[n]	Данные сеансового уровня	T_Data[1] – T_Data[n]	Транспорт/Сеть с логическими Данными Приложения
S_Длина	Данные сеансового уровня Длина	T_длина	Данные транспортного/сеансового уровня Длина
S_результат	Сеансовый уровень Результат	T_Result	Транспортный/сеансовый уровень Результат
^a Если Mtype = диагностическая, то адресная информация должна состоять из параметров SA, TA и TAtype. Если Mtype = удаленная диагностическая, то адресная информация должна состоять из параметров SA, TA, TAtype и AE.			

7 Определение временных параметров

7.1 Общие сообщения по времени применения

7.1.1 Сервер

Сервер использует реализацию единого прикладного таймера (P2Server), которая запускается (запускается и останавливается) примитивным интерфейсом службы T_Data (T_Data.ind, T_Data.con, T_Data.req).

ИСО 14229-2:2013(E)

Таймер приложения P2Server запускается с значением параметра P2Server_max/P2*Server_max. И параметры, и значения указаны в этой части ISO 14229 (с м. определение в таблице 3 и значение параметра в таблице 4).

Временной параметр P4Server — это время между приемом запроса (T_Data.indication) и началом передачи окончательного ответа (T_Data.request). Окончательный ответ — это положительный ответ или отрицательный ответ, отличный от кода отрицательного ответа 0x78 «requestCorrectlyReceived-ResponsePending». В случае запроса на планирование периодических ответов окончательным ответом считается первоначальный положительный или отрицательный ответ USDТ, указывающий на принятие или неприятие запроса на планирование периодических ответов. P4Server — это требование к производительности. P4Server_max — максимальное значение P4Server. Если P4Server_max совпадает с P2Server_max, это означает, что отрицательный ответ с кодом отрицательного ответа 0x78 не разрешен для этой службы или данных.

Эти требования применимы только к службам, поддерживаемым сервером/блоком управления. Не поддерживаемые службы все равно должны использовать значение P4Server_max, равное P2Server_max (т. е. NRC 0x78 не допускается).

7.1.2 Клиент

Клиент использует реализацию одного прикладного таймера (PClient), которая запускается (запускается перезапускается и останавливается) интерфейсом примитива службы T_Data (T_Data.con, T_DataSOM.ind, T_Data.ind).

Таймер приложения PClient все равно запускается с P2Client_max/P2*Client_max для всех протоколов, которые в принципе поддерживают служебный примитив T_DataSOM.ind (например, DoCAN). Таймер приложения PClient все равно запускается с значением параметра P6Client_max/P6*Client_max для всех протоколов, которые в принципе поддерживают только служебный примитив T_Data.ind (например, DoIP).

Таймер приложения PClient запускается всякий раз, когда уровень клиента с кода приложения получает сервисный примитив T_Data.con. В зависимости от типа протокола (с T_DataSOM.ind или без T_DataSOM.ind) он запускается с значением параметра P2Client_max или P6Client_max.

В зависимости от типа протокола (с T_DataSOM.ind или без T_DataSOM.ind) таймер приложения PClient останавливается, когда приложение получает сервисный примитив T_DataSOM.ind или T_Data.ind.

Для протоколов, поддерживающих T_DataSOM.ind, клиентское приложение проверяет правильность синхронизации приложения, сравнивая фактический таймер приложения PClient с значением параметра P2Client_max. Если T_DataSOM.ind или T_Data.ind получены, когда значение PClient меньше или равно P2Client_max, синхронизация соответствует требованиям, установленным настоящим стандартом. Если .ind не получен, в то время как PClient меньше или равен P2Client_max, обнаруживается состояние ошибки. Это должно быть отмечено для прикладного уровня с помощью параметров, включенных либо в сервисный примитив T_DataSOM.ind, либо в сервисный примитив T_Data.ind.

Для протоколов, которые поддерживают T_Data.ind, только клиентское приложение проверяет правильность синхронизации приложения, сравнивая фактический таймер приложения PClient с значением параметра P6Client_max. Если T_Data.ind получен, когда PClient меньше или равен P6Client_max, синхронизация соответствует требованиям, установленным этой частью ISO 14229. Если .ind не получен, когда PClient меньше или равен P6Client_max, обнаруживается состояние ошибки. Это должно быть отмечено прикладным уровнем с помощью параметров, включенных в сервисный примитив T_Data.ind.

Все параметры и значения указаны в этой части ISO 14229 (с м. определение в таблице 3 и значения параметров в таблице 4).

7.2 Определения параметров времени приложения — defaultSession

Сервер все равно должен запускать defaultSession при включении. Если никакой другой диагностический сеанс не запущен, то defaultSession будет выполняться до тех пор, пока сервер включен. Определения параметров синхронизации для defaultSession должны соответствовать таблице 3, а определения значений должны соответствовать таблице 4.

Таблица 3 — Определения параметров с их ронизацией с общений для defaultSession

Параметр времени	Описание	Тип
P2	Параметр P2 определяется как наихудший случай задержки передачи с сообщения, зависящий от конструкции сети транспортного средства, такой как задержки, вносимые шлюзами, и арбитраж, зависящий от загрузки шины. Значение P2 делится на время передачи запроса к адресуемого серверу/ЭБУ (P2request) и, если протокол поддерживает T_DataSOM.ind, до начала передачи ответа, указанного T_DataSOM.ind или T_Data.ind, если ответ представляет собой одноканальное сообщение (например, ISO 15765 DoCAN).	Требования к производительности
P6	Параметр P6 определяется как наихудший случай задержки передачи с сообщения, зависящий от конструкции сети транспортного средства, такой как задержки, вносимые шлюзами, и арбитраж, зависящий от загрузки шины. Значение P6 делится на время передачи запроса на адресный сервер/блок управления (P6request) и времени передачи полного ответа клиенту/тестеру (P6response). P6 не зависит от того, поддерживает ли протокол T_DataSOM.ind (например, ISO 15765 DoCAN) или не поддерживает T_DataSOM.ind (например, ISO 13400 DoIP).	Требования к производительности
P2сервер	Требование к производительности для запроса с сервера с ответным сообщением после получения сообщения запроса (указывается через T_Data.ind).	Требования к производительности
P2Клиент	Время ожидания клиента после успешной передачи с сообщения запроса (указывается через T_Data.con) для начала входящих ответных сообщений (указывается через T_DataSOM.ind многоканального сообщения или T_Data.ind одноканального сообщения).	Значение перезагрузки таймера
P6Клиент	Время ожидания клиента после успешной передачи с сообщения запроса (указывается через T_Data.con) для полного приема соответствующего ответного сообщения (указывается через T_Data.ind), например, ISO 13400 DoIP.	Значение перезагрузки таймера
P2*сервер	Требование к производительности для запроса с сервера с ответным сообщением после передачи отрицательного ответного сообщения (указанного через T_Data.con) с отрицательным кодом ответа 0x78 (улучшенное время ответа).	Требования к производительности
P2*Клиент	Увеличенный тайм-аут для клиента после получения отрицательного ответа с кодом отрицательного ответа 0x78 (указывается через T_Data.ind) для начала входящих ответных сообщений (указывается через T_DataSOM.ind многоканального сообщения или T_Data.ind с сообщения SingleFrame).	Значение перезагрузки таймера
P6*Клиент	Увеличенный тайм-аут для клиента после получения отрицательного ответного сообщения с отрицательным кодом ответа 0x78 (указывается через T_Data.ind) для полного приема соответствующих ответных сообщений (указывается через T_Data.ind), например ISO 13400 DoIP.	Значение перезагрузки таймера
P3Client_Phys	Минимальное время, в течение которого клиент ожидает после успешной передачи с сообщения запроса с физическим адресом (указывается через T_Data.con) без ответа, прежде чем он может передать следующему сообщению запроса с физическим адресом (см. рисунок 19).	Значение перезагрузки таймера
P3Client_Func	Минимальное время ожидания клиента после успешной передачи с сообщения запроса с функциональной адресацией (указывается через T_Data.con), прежде чем он может передать следующему сообщению запроса с функциональной адресацией в случае, если ответ не требуется или запрошенные данные поддерживаются только подмножеством серверов с функциональной адресацией (см. 8.3).	Значение перезагрузки таймера
P4сервер	Это время между приемом запроса (T_Data.indication) и началом передачи окончательного ответа (T_Data.request) на стороне сервера.	Требования к производительности

Каждый сервер/ЭБУ должен иметь возможность обрабатывать новое сообщение запроса сразу же после успешной передачи ответного сообщения (T_Data.con) из предыдущего сообщения запроса. Исключение из этого требования может быть представлено изготовителем транспортного средства для некоторых случаев использования, когда серверу/ЭБУ требуется дополнительное время после выполнения предыдущего запроса на обслуживание, например с лужбы EcuReset.

ИСО 14229-2:2013(E)

Таблица 4 — Определения значений параметров их ронизации с общений для defaultSession

Параметр времени	Минимум	Максимум
P2	0 мс	конкретное значение производителя транспортного средства P2запрос + P2ответ
P6	0 мс	конкретное значение производителя транспортного средства P6запрос + P6ответ
P2сервер	0 мс	конкретное значение сервера рекомендуемое значение: 50 мс
P2Клиент	P2Server_max + P2max	--- а)
P6Клиент	P2Server_max + P6max	--- а)
P2*сервер	б) 0 мс	конкретное значение сервера рекомендуемое значение: 5 000 мс
P2*Клиент	P2*Server_max + P2ответ ---	в)
P6*Клиент	P2*Server_max + P6ответ ---	в)
P3Client_Phys	P2Server_max + P2max	--- г)
P3Client_Func	P2Server_max + P2max	--- г)
P4сервер	P2сервер	Как определено законодательством для случаев использования диагностики OBD. Специальное значение производителя автомобиля для расширенного диагностического использования.
<p>а) Максимальное время, в течение которого клиент ожидает ответа с общения, определяется клиентом при условии, что P2Client / P6Client больше указанного минимального значения P2Client / P6Client.</p> <p>б) Во время расширенного времени ответа минимальное время между передачей последовательных отрицательных сообщений (каждое с отрицательным кодом ответа 0x78) должно составлять $0,3 * P2 * Server_max$, чтобы избежать перегрузки канала данными ненужными отрицательными сообщениями с кодом ответа 0x78.</p> <p>с) Максимальное значение, которое клиент использует для P2*Client / P6*Client, определяется клиентом при условии, что оно больше указанного минимального значения P2*Клиент / P6*Клиент.</p> <p>д) Максимальное время, в течение которого клиент ожидает передачи следующего сообщения запроса, определяется клиентом, при условии, что для сеансы по умолчанию, с их ронизация S3Server остается активной на сервере(ах).</p>		

Параметр P2/ P6 учитывает любые задержки, зависящие от конструкции и сети системы, такие как задержки, вызванные шлюзами и полосой пропускания шины, плюс запас прочности (например, 50 % наихудшего случая). На сценарий наихудшего случая (время передачи, необходимое для одного «цикла» от клиента к серверу и обратно от сервера к клиенту), основанный на конструкции и системе, влияет

- количество задействованных шлюзов,

- время передачи кадра (скорость передачи),

- использование автобуса и

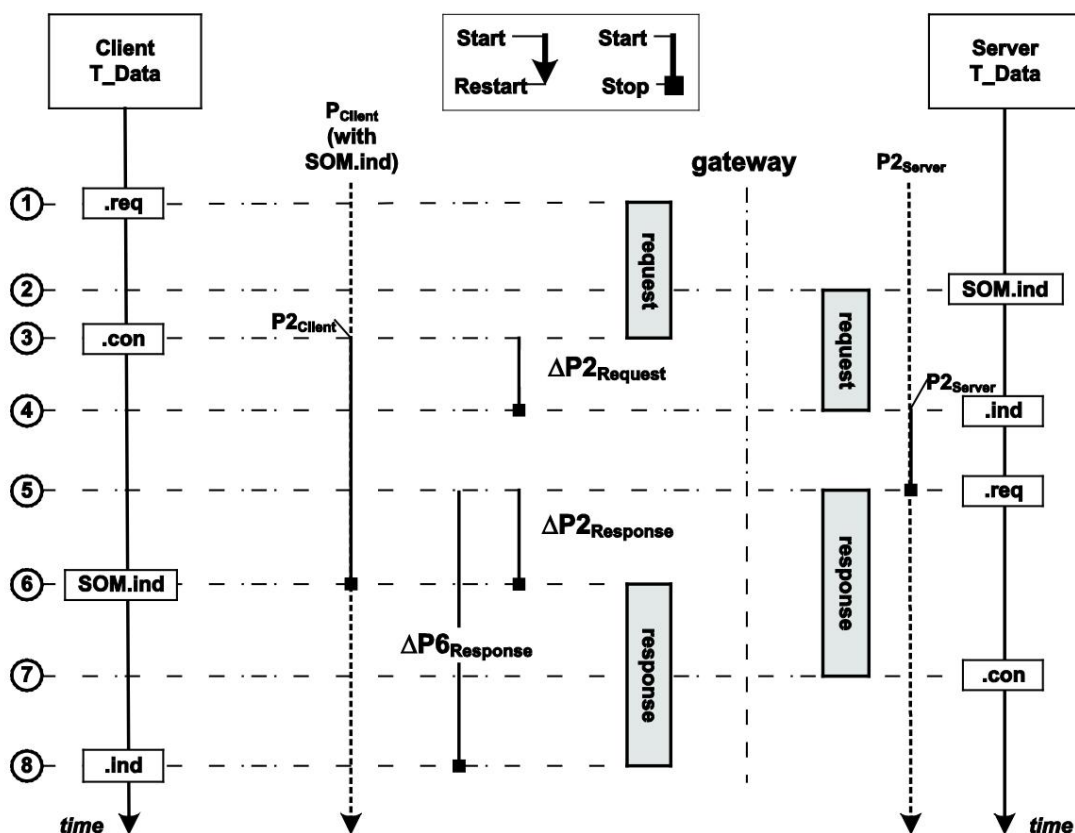
- метод реализации драйвера устройства (опрос или прерывание) и время обработки на транспортном/сетевом уровне.

Значение P2/ P6 делится на время передачи запроса к адресуемому серверу и время передачи ответа клиенту:

$$P2 = P2\text{запрос} + P2\text{ответ}$$

$$P6 = P6\text{запрос} + P6\text{ответ}$$

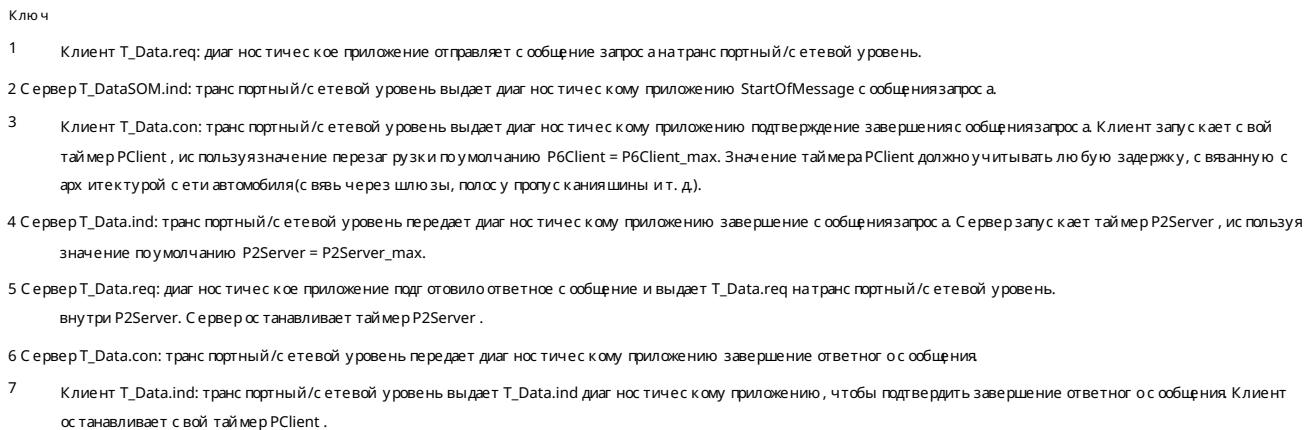
На Рисунке 4 и Рисунке 5 показан пример того, как можно составить P2 и P6.



Ключ

- 1 Клиент T_Data.req: диагностическое приложение отправляет сообщение запроса на транспортный/сетевой уровень.
- 2 Сервер T_Data.SOM.ind: транспортный/сетевой уровень выдает диагностическому приложению StartOfMessage с сообщением запроса.
- 3 Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения с сообщением запроса. Клиент запускает свой таймер P2_Client, используя значение перезагрузки по умолчанию P2_Client = P2_Client_max. Значение таймера P2_Client должно учитывать любую задержку, связанную с архитектурой сети автомобиля (связь через шлюзы, полосу пропускания шины и т. д.).
- 4 Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщением запроса. Сервер запускает таймер P2_Server, используя значение по умолчанию P2_Server = P2_Server_max.
- 5 Сервер T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный/сетевой уровень. внутри P2_Server. Сервер останавливает таймер P2_Server.
- 6 Клиент T_Data.SOM.ind: транспортный/сетевой уровень выдает диагностическому приложению StartOfMessage ответного сообщения. Клиент останавливает свой таймер P2_Client.
- 7 Сервер T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
- 8 Клиент T_Data.ind: транспортный/сетевой уровень выдает T_Data.ind диагностическому приложению, чтобы подтвердить завершение ответа с сообщением.

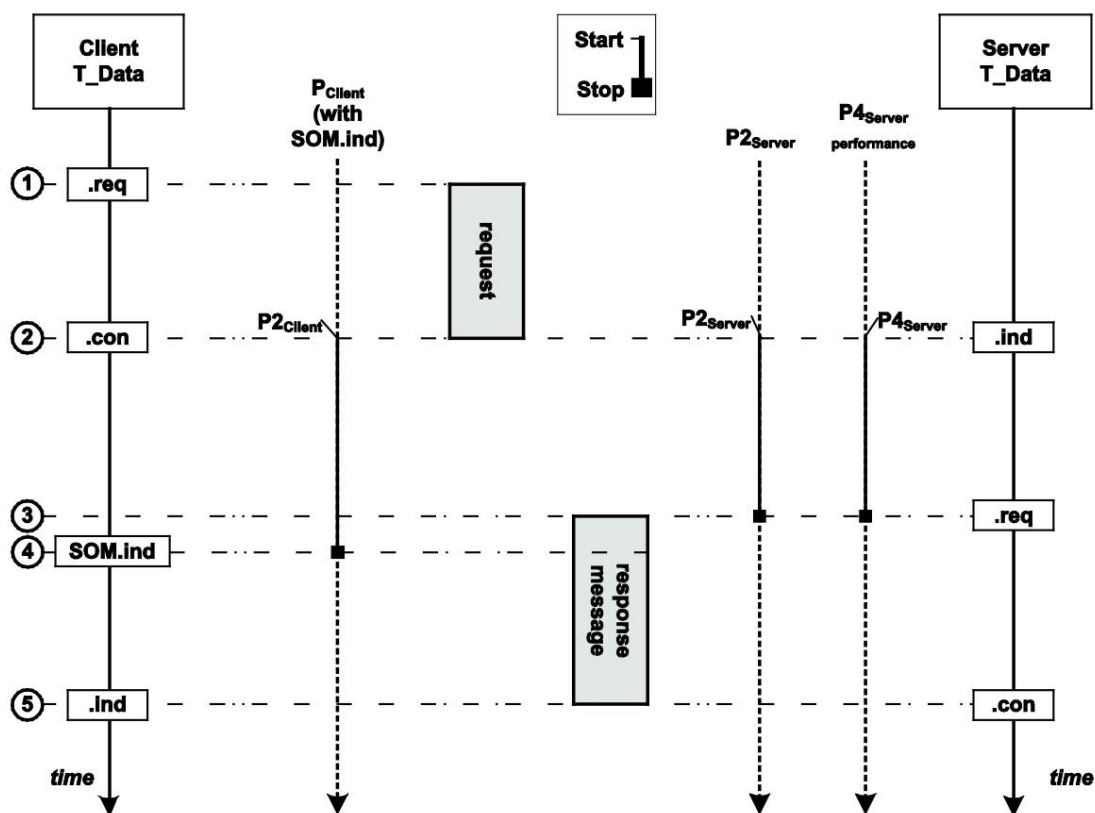
Рисунок 4 — Пример для P2 и P6 — Ответное сообщение с SOM.ind



ПРИМЕЧАНИЕ. Для простоты описания временных параметров на всех следующих рисунках предполагается, что клиент и сервер расположены в одной сети. Все описания и рисунки представлены в последовательном порядке во времени.

7.3 Пример для P4Server без улучшения времени отклика

На рис. 6 показан пример, где P4Server = P2Server. В этом сценарии параметр времени ответа сервера указывает, что отрицательные ответы, включая NRC 0x78, не допускаются.



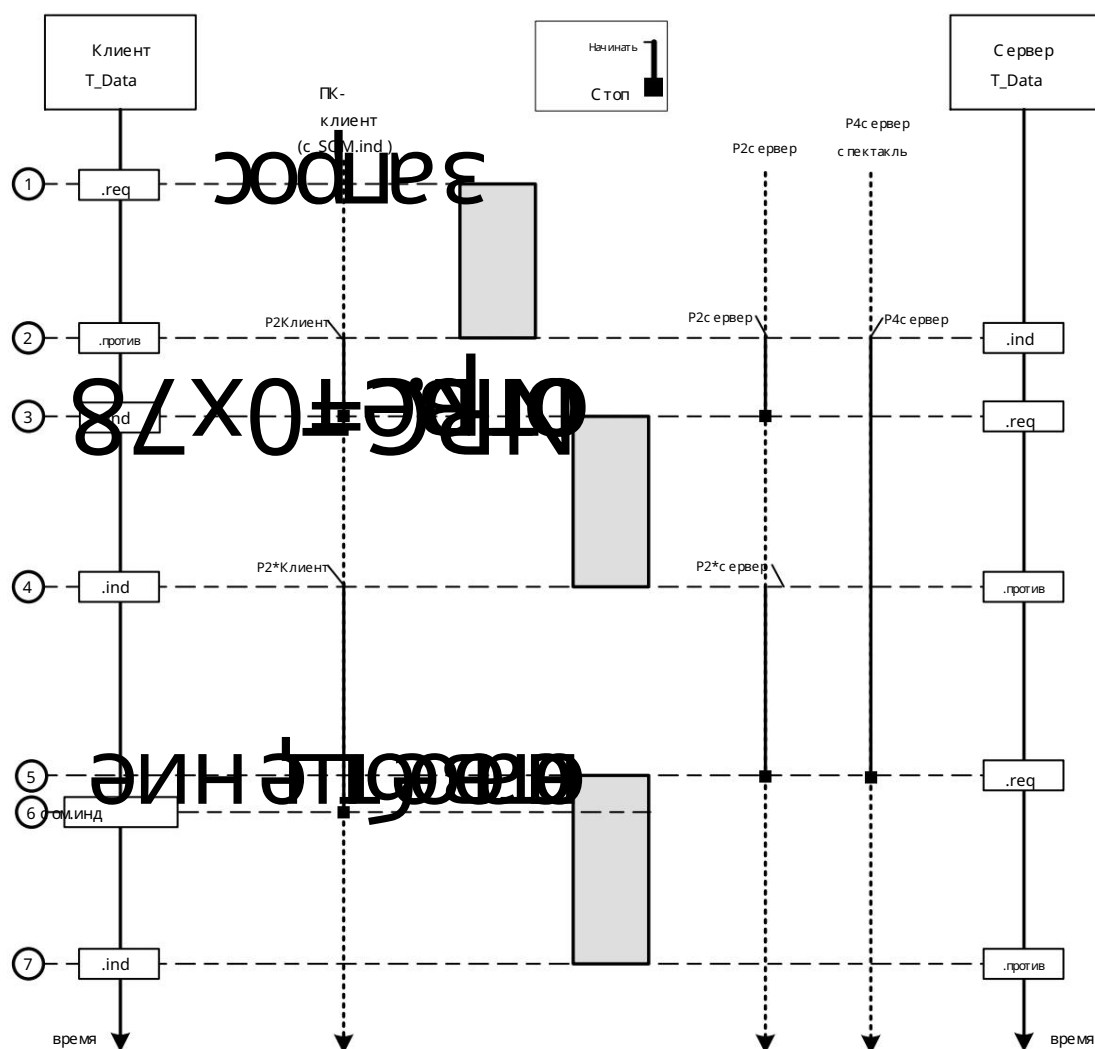
Ключ

- 1 Клиент T_Data.req: диагностическое приложение отправляет с сообщение запроса на транспортный/сетевого уровень.
- 2 Сервер T_Data.ind: транспортный/сетевого уровень передает диагностическому приложению завершение с сообщения запроса. Сервер запускает таймер P2Server, используя значение по умолчанию P2Server = P2Server_max, и таймер P4Server, используя значение по умолчанию P4Server = P4Server_max. Если для определенного T_Data.ind применяется P4Server = P2Server, сервер должен гарантировать, что окончательный положительный или отрицательный ответ будет получен до истечения времени таймера P2Server (т.е. никакие отрицательные ответы с NRC 0x78 не допускаются).
Клиент T_Data.con: транспортный/сетевого уровень выдает диагностическому приложению подтверждение завершения с сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети автомобиля (связь через шлюзы, полосы пропускания шины и т.д.).
- 3 Сервер T_Data.req: диагностическое приложение подготовило ответное с сообщение и отправляет T_Data.req на транспортный/сетевого уровень в P2Server. Сервер останавливает таймер P2Server.
Таймер производительности P4Server остановлен.
- 4 Клиент T_Data.SOM.ind: транспортный/сетевого уровень передает диагностическому приложению получение с сообщения StartOfMessage. Клиент останавливает таймер PClient.
- 5 Сервер T_Data.con: транспортный/сетевого уровень передает диагностическому приложению завершение ответного с сообщения.
Клиент T_Data.ind: транспортный/сетевого уровень передает диагностическому приложению завершение ответного с сообщения.

Рисунок 6 — Пример для P4Server без улучшения времени отклика

7.4 Пример для P4Server с улучшенным временем отклика

На рис. 7 показан пример, где $P4Server > P2Server$. В этом сценарии параметр времени ответа сервера указывает, что отрицательные ответы, включая NRC 0x78, разрешены, пока не превышено значение P4Server.



Ключ

- 1 Клиент T_Data.req: диагностическое приложение отправляет сообщение запроса на транспортный/сетевой уровень.
- 2 Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение сообщения запроса. Сервер запускает таймер P2Server, используя значение по умолчанию P2Server = P2Server_max, и таймер P4Server, используя значение по умолчанию P4Server = P4Server_max.
Если для определенного T_Data.ind применяется P4Server > P2Server, сервер должен гарантировать, что окончательный положительный или отрицательный ответ будет запущен до истечения таймера P2Server (т.е. разрешены отрицательные ответы с NRC 0x78 до тех пор, пока P4Server не превышен).
Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети автомобиля (связь через шлюзы, полосу пропускания шины и т.д.).
- 3 Сервер T_Data.req: диагностическое приложение не имеет готовности к положительному ответу на сообщение и выдает отрицательное сообщение с NRC = 0x78 с помощью T_Data.req на транспортный/сетевой уровень в P2Server. Сервер останавливает таймер P2Server.
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению получение ответа на сообщение. Клиент останавливает таймер PClient.
- 4 Сервер T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответа на сообщение. Сервер запускает таймер P2Server, используя значение P2*Server = P2*Server_max (улучшенное время по умолчанию).

Если сервер по-прежнему не может предоставить запрошенную информацию в расширенном P2*Server, то сервер может отправить дополнительное сообщение с отрицательным ответом, включая код отрицательного ответа 0x78. Это заставляет клиент перезапустить свой таймер PClient, используя расширенное значение перезагрузки P2*Client. Для простоты на рисунке показано только одно сообщение отрицательного ответа с кодом отрицательного ответа 0x78.

Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению получение ответного сообщения. Клиент запускает таймер PClient с означением P2*Client = P2*Client_max (улучшенное время по умолчанию).

- 5 Сервер T_Data.req: диагностическое приложение подготовило ответное сообщение и отправляет T_Data.req на транспортный/сетевой уровень в P2Server. Сервер останавливает таймер P2Server.
- Таймер производительности P4Server остановлен.
- 6 Клиент T_Data.SOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение сообщения StartOfMessage. Клиент останавливает таймер PClient.
- 7 Сервер T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
- Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.

Рисунок 7 — Пример для P4Server с улучшенным временем отклика

7.5 Определения параметров времени сеанса для сеанса не по умолчанию

Когда запускается диагностический сеанс, отличный от defaultSession, требуется обработка сеанса, которая достигается с помощью временных параметров сеанса этого уровня, приведенных в таблице 5.

Таблица 5 — Определения параметров времени сеанса для сеанса не по умолчанию

Параметр времени	Описание	Тип	Рекомендуемая перезагрузка РС	Тайм-аут РС
S3Клиент	Время между функционально адресуемыми сообщениями запроса TesterPresent (0x3E), передаваемыми клиентом, чтобы поддерживать сеанс диагностики, отличный от defaultSession, активным на нескольких серверах (функциональная связь), или максимальное время между физическими передаваемыми сообщениями запроса на один сервер (физическая связь). Значение тайм-аута S3Client включает время ожидания сообщения по сети (задержки шлюза и т. д.).	Значение перезагрузки таймера	2 000	<S3Сервер
S3Сервер	Время, в течение которого сервер должен поддерживать сеанс диагностики, отличный от defaultSession, активным, не получая никаких сообщений с запросом на диагностику. Допуск S3Server составляет -0 мс, +200 мс.	Значение перезагрузки таймера	Н/Д	5 000

Определения и значения параметров с индексацией, указанные в таблицах 3 и 4, также действительны для сеанса не по умолчанию. Кроме того, сервер может изменить свои временные параметры прикладного уровня P2Server и P2*Server при переходе в сеанс не по умолчанию, чтобы достичь определенной производительности или компенсировать ограничения, которые могут применяться во время сеанса диагностики не по умолчанию. Применимые параметры времени для диагностического сеанса не по умолчанию с общими сообщениями положительного ответа DiagnosticSessionControl в случае, когда ответ требуется передать или он должен быть известен клиенту заранее, если не требуется передача ответа. Когда клиент функционально запускает нестандартный сеанс, он должен адаптироваться к временным параметрам отвечающих серверов.

Таблица 5 определяет условия запуска/перезапуска таймера S3Client/S3Server клиентом и сервером. Для клиента периодически передаваемое функционально адресованное сообщение запроса TesterPresent (0x3E) должно отличаться от последовательно передаваемого сообщения запроса TesterPresent (0x3E) с физической адресацией, которое передается только в случае отсуствия любого другого сообщения запроса диагностики. Для сервера нет необходимости различать такую обработку TesterPresent (0x3E). Кроме того, в таблице 5 показано, что обработка таймера S3Server основана на сервисных примитивах транспортного/сетевого уровня, что означает

Во время сеанса не по умолчанию для клиента и сервера должны применяться дополнительные требования к ресурсам таймера, указанные в таблице 6.

Параметр времени	Действие	Физическая функция взаимодействия с использованием функционально адресуемого периодического передаваемого сообщения запроса TesterPresent.	Только физическая связь с использованием последовательно передаваемого запроса TesterPresent с физической адресацией.
S3Клиент	Персональный запуск	T_Data.con, указывающий на завершение сообщения запроса DiagnosticSessionControl (0x10). Это верно только в том случае, если тип сеанса не является сеансом по умолчанию.	T_Data.con, указывающий на завершение сообщения запроса DiagnosticSessionControl (0x10), если ответ не требуется
			T_Data.ind, указывающий на получение ответного сообщения DiagnosticSessionControl (0x10) в случае, если требуется ответ.
	Последующий запуск	T_Data.con, указывающий на завершение функционально адресованного сообщения запроса DiagnosticSessionControl (0x10), которое должно быть передано в течение 10 секунд после приема сообщения TesterPresent.	T_Data.con, который указывает на завершение любого сообщения запроса, если ответ не требуется
			T_Data.con, указывающий на ошибку во время передачи одноадресного или многоадресного сообщения запроса.
			T_Data.ind, указывающий на получение любого ответного сообщения в случае, если требуется ответ.
			T_Data.ind, указывающий на ошибку во время приема многоадресного ответного сообщения
S3Сервер	Персональный запуск	T_Data.con, указывающий на завершение передачи положительного ответного сообщения DiagnosticSessionControl для перехода от сеанса по умолчанию к сеансу не по умолчанию, если требуется ответное сообщение.	
		Успешное выполнение запрошенного действия службы DiagnosticSessionControl (0x10) для перехода из сеанса по умолчанию в сеанс не по умолчанию, в случае, если ответное сообщение не требуется/разрешено.	
	Последующие остановки	T_DataSOM.ind, который указывает на начало сообщения запроса мультикадра, или T_Data.ind, который указывает на прием любого сообщения запроса SingleFrame. Если defaultSession активен, таймер S3Server отключен.	
	Последующий запуск	T_Data.con, указывающий на завершение любого ответного сообщения, завершающего выполнение службы (окончательное ответное сообщение), в случае, если требуется/разрешена передача ответного сообщения (включая положительные и отрицательные ответные сообщения). Отрицательный ответ с кодом отрицательного ответа 0x78 не перезапускает таймер S3Server.	
		Завершение запрошенного действия (завершение усугубления) в случае, если ответное сообщение (положительное и отрицательное) не требуется/разрешено.	
		T_Data.ind, указывающий на ошибку во время приема сообщения запроса мультикадра.	
		Для получения дополнительной информации об обработке S3Server на сервере, когда сервер запрашивает для передачи незапрашиваемого ответного сообщения, такого как периодические данные или ответы, основанные на событии, см. конкретные документы по реализации канала передачи данных ISO 14229.	

7.6 Требования к ресурсам таймера клиента и сервера

Ресурс таймера, необходимый клиенту и серверу для выполнения приведенных выше требований к времени во время сеанса по умолчанию и любого сеанса по умолчанию, должен соответствовать списку в Таблице 7 и Таблице 8.

Таблица 7 определяет требования к ресурсам таймера во время defaultSession и не-defaultSession.

Таблица 7 — Требования к ресурсам таймера во время defaultSession

Параметр времени	Клиент	Сервер
ПК-клиент	Для каждого логического канала связи (физической и функциональной связи) требуется один таймер, например, для каждой двухточечной связи требуется отдельный канал связи.	Н/Д
P2c сервер	Н/Д	Для расширенного времени ответа требуется один таймер, чтобы гарантировать, что последующие сообщения отрицательным ответом с кодом отрицательного ответа 0x78 будут переданы до истечения срока действия P2*Server.
P3Client_Phys	Для каждого логического физического канала связи требуется один таймер.	Н/Д
P3Client_Func	Для каждого логического функционального канала связи требуется один таймер.	Н/Д

Таблица 8 определяет дополнительные требования к ресурсам таймера во время сеанса, отличного от defaultSession.

Таблица 8 — Требования к дополнительным ресурсам таймера во время сеанса, отличного от defaultSession

Параметр времени	Клиент	Сервер
S3Клиент	Один таймер требуется при использовании периодических передаваемых сообщений запроса TesterPresent (0x3E) с функциональной адресацией, чтобы удерживать серверы в сеансе, отличном от defaultSession. Нет необходимости в дополнительных таймерах для активированных диагностических сеансов.	Н/Д
	Один таймер требуется для каждого канала связи точка-точка при использовании последовательно передаваемых сообщений запроса TesterPresent (0x3E) с физической адресацией, чтобы удерживать один сервер в сеансе, отличном от defaultSession, в случае отсутствия другого сообщения запроса диагностики.	
S3Сервер	Н/Д	На сервере требуется один таймер, поскольку одновременно на одном сервере может быть активен только один сеанс диагностики.

ИСО 14229-2:2013(E)

7.7 Обработка ошибок

Обработка ошибок для прикладного уровня управления сеансом, которые должны выполняться клиентом и сервером во время физической и функциональной связи, должна соответствовать Таблице 9 и Таблице 10, в отношении которых предполагается, что клиент и сервер реализуют приложение. и синхронизация сеансового уровня с соответствием этой частью ISO 14229.

Таблица 9 — Рекомендации по общей обработке ошибок клиента

Фаза общения	Тип ошибки клиента	Работа с клиентами	
		Физическая связь	Функциональная коммуникация
Запросить передачу	T_Data.con с транспортно-сетевым уровнем отрицательным значением результата	Клиент должен повторить последний запрос по истечении времени P3Client_Phys, следуя указанию за индикацией ошибки. Перезапустите S3Client в случае физического обращения и последовательной передачи TesterPresent (поскольку S3Client был остановлен на основе передачи сообщения запроса).	Клиент должен повторить последний запрос по истечении времени P3Client_Func, следуя указанию за индикацией ошибки.
ПК-клиент ПК-клиент	Тайм-аут	Клиент должен повторить последний запрос. Перезапустите S3Client в случае физического обращения и последовательной передачи TesterPresent (поскольку S3Client был остановлен на основе передачи сообщения запроса).	Если клиент не знает количество ответивших серверов, это указывает на то, что дальнейших ответных сообщений не ожидается. Повтор сообщения запроса не требуется.
			Клиент должен полностью получать все сообщения ответа, которые находятся в процессе выполнения, до тех пор, пока он не сможет продолжить дальнейшие запросы. Если клиенту известно количество отвечающих серверов, это означает, что не все ожидаемые серверы ответили. Клиент должен повторить запрос после того, как он полностью получил любое ответное сообщение, которое обрабатывается в момент времени, когда происходит тайм-аут.
Прием ответа	T_Data.ind с транспортным уровнем отрицательным значением результата	Клиент должен повторить последний запрос. Перезапустите S3Client в случае физического обращения и последовательной передачи TesterPresent (поскольку S3Client был остановлен на основе передачи сообщения запроса).	Клиент должен повторить последний запрос после того, как он полностью получил любое ответное сообщение, которое обрабатывается в момент времени, когда была указана ошибка.
Определенная обработка ошибок клиента должна выполняться не более двух раз, что означает, что наилучший случай передачи запросов на обслуживание равен трем.			

Таблица 10 определяет обработку ошибок сервера.

Таблица 10 — Обработка ошибок сервера

Фаза общения	Тип ошибки сервера	Обработка сервера
Запросить прием	T_Data.ind с транспортным уровнем отрицательным значением	Перезапустите таймер S3Server (поскольку он был остановлен на основании полученного ранее указания StartOfMessage).

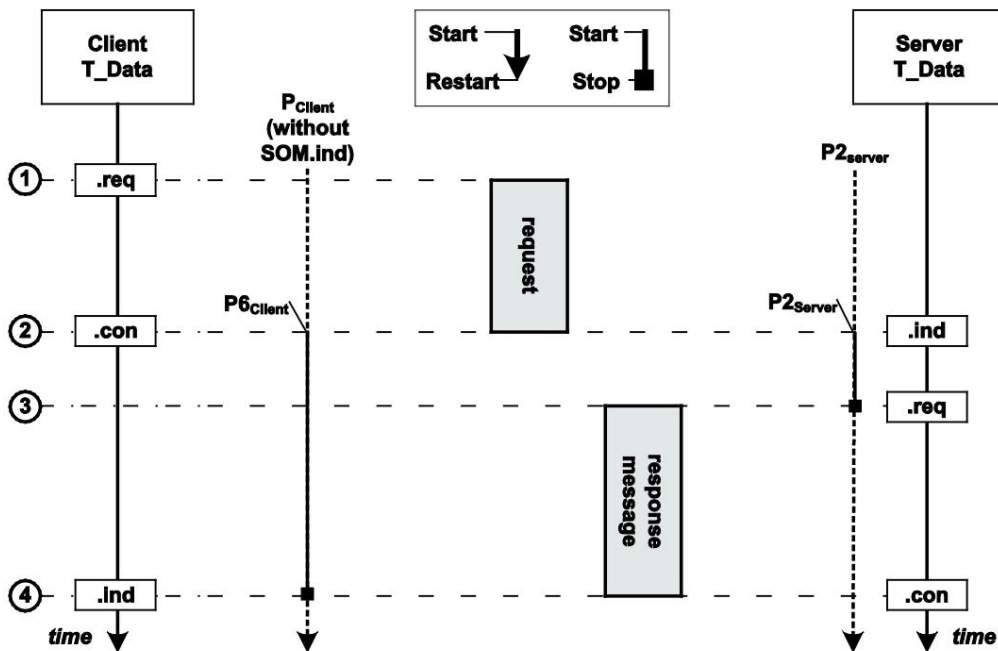
Фаза общения	Тип ошибки с сервера	Обработка с сервера
	значение результата.	сервер должен игнорировать запрос.
Передача ответа	T_Data.con с транспортно / сетевым уровнем отрицательным значением результата.	Перезапустите таймер S3Server (поскольку он был установлен на последнее полученное сообщение с запросом). Сервер не должен выполнять повторную передачу ответного сообщения.

8 Обработка инхронизации во время связи

8.1 Физическая связь

8.1.1 Физическая связь во время defaultSession — без SOM.ind

На рис. 8 графически показана обработка инхронизации на клиенте и сервере для сообщения запроса с физической адресацией без SOM.ind во время сеанса по умолчанию.



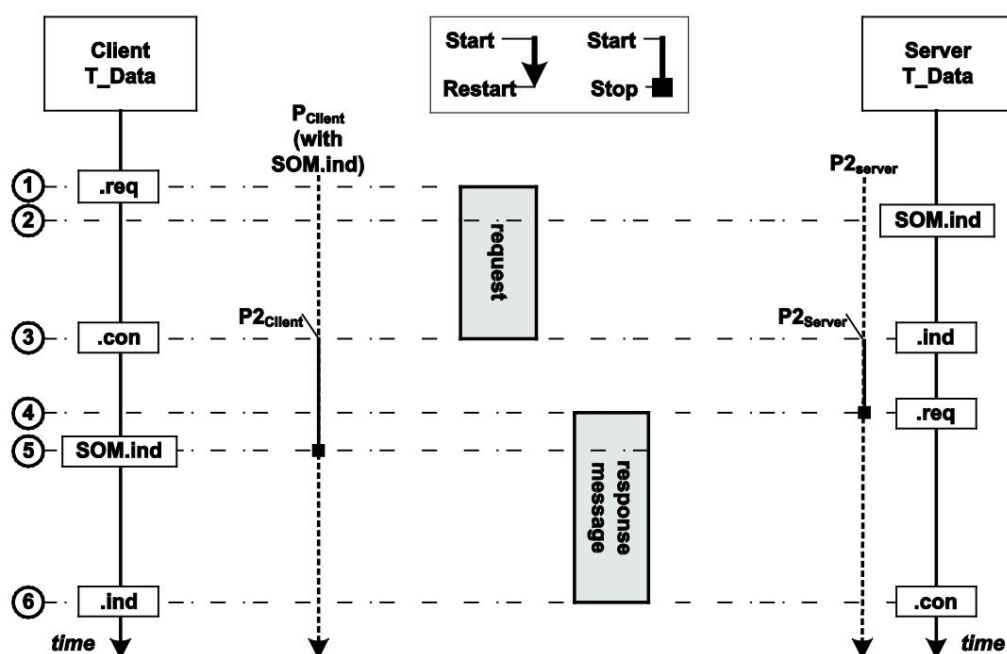
Ключ

- 1 Клиент T_Data.req: диагностическое приложение отправляет сообщение запроса на транспортный / сетевой уровень.
- 2 Сервер T_Data.ind: транспортный / сетевой уровень передает диагностическому приложению завершение с сообщением запроса. Сервер запускает таймер P2Server, используя значение по умолчанию P2Server = P2Server_max.
Клиент T_Data.con: транспортный / сетевой уровень выдает диагностическому приложению подтверждение завершения с сообщением запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P6Client = P6Client_max. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети автомобиля (связь через шлюзы, полосу пропускания шины и т. д.).
- 3 Сервер T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный / сетевой уровень.
внутри P2Server. Сервер останавливает таймер P2Server.
- 4 Сервер T_Data.con: транспортный / сетевой уровень передает диагностическому приложению завершение ответного сообщения.
Клиент T_Data.ind: транспортный / сетевой уровень передает диагностическому приложению завершение ответного сообщения. Клиент останавливает свой таймер PClient.

Рисунок 8 — Физическая связь во время сеанса по умолчанию — без SOM.ind

8.1.2 Физическая связь во время defaultSession — с SOM.ind

На рис. 9 графически показана обработка информации на клиенте и сервере для сообщения запроса с физической адресацией с помощью SOM.ind во время сеанса по умолчанию.



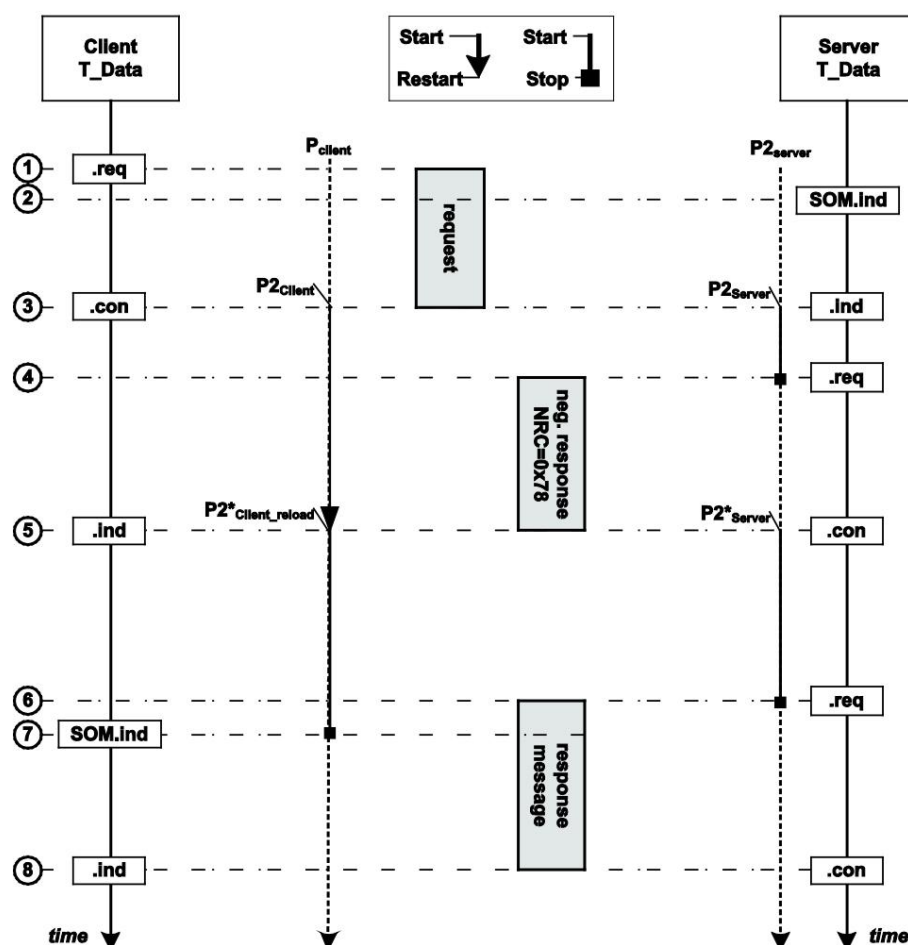
Ключ

- 1 Клиент T_Data.req: диагностическое приложение отправляет с сообщение запроса на транспортный/сетевой уровень.
- 2 Сервер T_Data.SOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение StartOfMessage запроса с сообщение.
- 3 Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса. Сервер запускает таймер P2_Server, используя значение по умолчанию P2_Server = P2_Server_max.
Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения с сообщения запроса. Клиент запускает свой таймер P_Client, используя значение перезагрузки по умолчанию P_Client = P_Client_max. Значение таймера P_Client должно учитывать любую задержку, связанную с архитектурой сети автомобиля (связь через шлюзы, полосу пропускания шины и т.д.). Для простоты на рисунке предполагается, что клиент и сервер находятся в одной сети.
- 4 Сервер T_Data.req: диагностическое приложение подготовило ответное с сообщение и выдает T_Data.req на транспортный/сетевой уровень внутри P2_Server. Сервер останавливает таймер P2_Server.
- 5 Клиент T_Data.SOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение с сообщения StartOfMessage. Клиент останавливается таймер P_Client.
- 6 Сервер T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения.
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения.

Рисунок 9 — Физическая связь во время сеанса по умолчанию — с SOM.ind

8.1.3 Физическая связь во время defaultSession с улучшенным временем отклика

На рис. 10 графически показана обработка времени в клиенте и сервере для сообщения запроса с физической адресацией во время сеанса по умолчанию и запроса с сервера на расширенное время ответа (обработка отрицательного кода ответа 0x78).



Ключ

- 1 Клиент T_Data.req: диагностическое приложение отправляет с сообщение запроса на транспортный/сетевой уровень.
- 2 Сервер T_Data.SOM.ind: транспортный/сетевой уровень выдает диагностическому приложению получение с сообщения StartOfMessage с сообщения запроса, если транспортный/сетевой уровень поддерживает интерфейс с T_Data.SOM.ind. Сервер T_Data.ind: транспортный/сетевой уровень выдает диагностическому приложению завершение, с сообщения запроса. Сервер запускает таймер P2Server, используя значение по умолчанию P2Server = P2Server_max.
- 3

Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения с сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.

- 4 Сервер T_Data.req: диагностическое приложение не имеет ответа положительного ответа с сообщения и выдает отрицательное ответное с сообщением NRC = 0x78 с помощью T_Data.req на транспортный/сетевой уровень в P2Server. Сервер останавливает таймер P2Server.
- 5 Сервер T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения. Сервер запускает таймер P2Server, используя значение P2Server = P2*Server_max (расширенное время по умолчанию).

Если сервер по-прежнему не может предоставить запрошенную информацию в расширенном P2*Server, то сервер может отправить дополнительное с сообщением с отрицательным ответом, включая код отрицательного ответа 0x78. Это заставляет клиента перезапустить свой таймер PClient, используя расширенное значение перезагрузки P2*Client. Для простоты на рисунке показано только одно сообщение отрицательного ответа с кодом отрицательного ответа 0x78.

Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению получение ответного с сообщения. Клиент останавливает таймер PClient и перезагружает его с обозначением P2*Client = P2*Client_max (расширенное время по умолчанию).

- 6 Сервер T_Data.req: диагностическое приложение выдает ответное с сообщением (положительный или отрицательный ответ, кроме кода отрицательного ответа 0x78) на транспортный/сетевой уровень. Сервер останавливает таймер P2Server.
- 7 Клиент T_Data.SOM.ind: транспортный/сетевой уровень выдает диагностическому приложению прием StartOfMessage ответного с сообщения, если транспортный/сетевой уровень поддерживает интерфейс с T_Data.SOM.ind. Клиент останавливает таймер PClient.
- 8 Сервер T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения.

Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению получение ответного с сообщения. При получении этого указания клиент останавливает свой таймер PClient, если транспортный/сетевой протокол не поддерживает интерфейс с T_Data.SOM.ind.

Рисунок 10 — Физическая связь во время сеанса по умолчанию — улучшенное время отклика

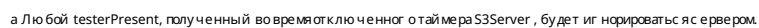
ИСО 14229-2:2013(Е)

8.1.4 Физическая связь во время сеанса по умолчанию

8.1.4.1 Функционально адресованное сообщение TesterPresent (0x3E)

На рис. 11 графически показана обработка синхронизации на клиенте и сервере при выполнении физической связи во время сеанса, отличного от сеанса по умолчанию (например, сеанса программирования) и использования функционально адресованного, периодически передаваемого сообщения запроса TesterPresent (0x3E), которое не требует ответного сообщения от сервера.

Обработка синхронизации PClient и P2Server идентична обработке, описанной в 8.1.2. Единственным исключением является то, что значения перезагрузки на стороне клиента и результирующее время, когда сервер должен отправить свое окончательное время ответа, могут отличаться. Это основано на переходе в сеанс, отличный от сеанса по умолчанию, где могут применяться другие параметры синхронизации PClient (с помощью DiagnosticSessionControl (0x10) в ISO 14229-1 для получения подробной информации о том, как параметры времени сообщаются клиенту).



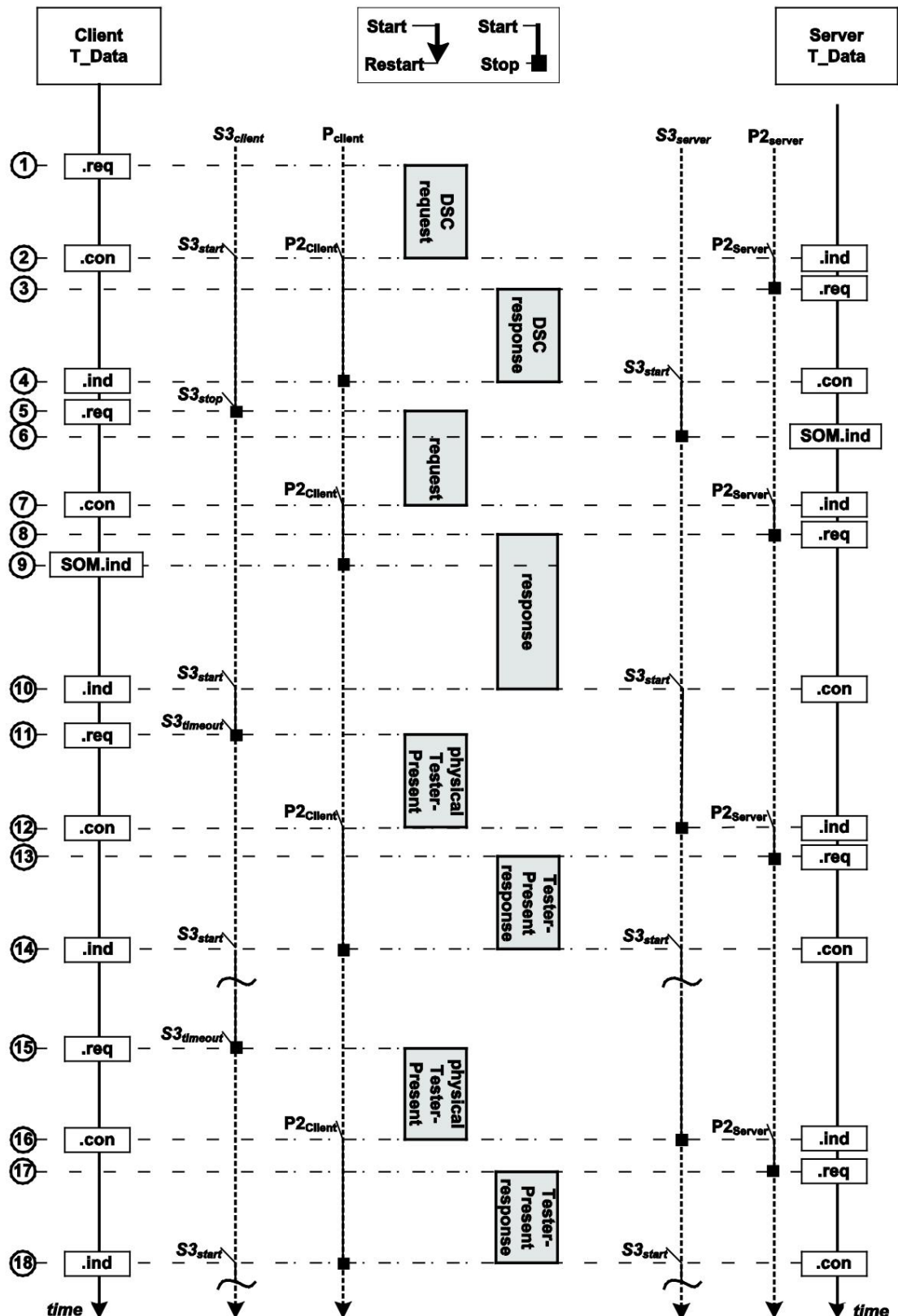
ИСО 14229-2:2013(E)

- 1 Клиент T_Data.req: диагностическое приложение отправляет с сообщение запроса DiagnosticSessionControl (0x10) на транспортный/сетевой уровень.
Транспортный/сетевой уровень передает с сообщение запроса на сервер.
- 2 Клиент T_Data.con: транспортный/сетевой уровень передает диагностическому приложению получение с сообщения запроса DiagnosticSessionControl (0x10).
Теперь применяется время отклика PClient, как описано в 8.1.2. Сгенерированный T_Data.con в клиенте вызывает запуск таймера S3Client (таймера с еанса).
Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.
Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса DiagnosticSessionControl (0x10).
Теперь применяется время ответа P2Server, как описано в 8.1.2.
- 3 Сервер T_Data.req: диагностическое приложение выдает положительное ответное с сообщение DiagnosticSessionControl (0x10) на транспортный/сетевой уровень. Для приведенного рисунка предполагается, что клиент требует ответа от сервера.
- 4 Сервер T_Data.con: по завершении передачи ответного с сообщения сообщается на сервере через T_Data.con. Теперь сервер запускает с вой таймер S3Server, который сокращает активированный нестандартный с еанс активным до тех пор, пока не истечет время ожидания. Клиент несет ответственность за то, чтобы таймер S3Server был сброшен до истечения времени ожидания, чтобы сервер оставался в с еансе, отличном от с еанса по умолчанию.
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения. Клиент останавливает таймер PClient.
- 5 Клиент T_Data.req: диагностическое приложение отправляет новое с сообщение запроса на транспортный/сетевой уровень.
- 6 Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса. Каждый раз, когда сервер обрабатывает какую-либо диагностическую службу, он останавливает с вой таймер S3Server. Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.
Клиент T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса.
- 7 Сервер T_Data.req: диагностическое приложение выдает положительное ответное с сообщение на транспорт/сеть.
- 8 Клиент T_DataSOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение StartOfMessage, если интерфейс T_DataSOM.ind поддерживается транспортным/сетевым уровнем. Клиент останавливает таймер PClient.
- 9 Client T_Data.req: после запуска таймера S3Client в клиенте это вызывает передачу с сообщения запроса TesterPresent (0x3E) с функциональной адресацией, которое не требует ответного с сообщения каждый раз, когда таймер S3Client истекает.
- 10 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с воет транспорт/сетевой уровень клиент еще раз запускает с вой таймер S3Client. Это означает, что с сообщение запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.
Сервер T_Data.ind: любое с сообщение запроса TesterPresent (0x3E), полученное во время обработки другого с сообщения запроса, может быть проигнорировано сервером, поскольку он уже установил с вой таймер S3Server и перезапустит его после полной обработки выполняемой службы.
- 11 Клиент T_Data.ind: после того, как транспортный/сетевой уровень отправляет диагностическому приложению завершение ответного с сообщения
Сервер T_Data.con: когда служба диагностики полностью обработана, сервер перезапускает с вой таймер S3Server. Это означает, что любой диагностический сервис, включая TesterPresent (0x3E), сбрасывает таймер S3Server. Диагностическая служба считается работающей в любое время между началом приема с сообщения запроса (прием T_DataSOM.ind или T_Data.ind) и завершением передачи окончательного ответного с сообщения, когда требуется ответное с сообщение, или завершение любого действия, вызванного запросом, когда ответное с сообщение не требуется (до наступления момента времени, который может вызвать запуск с ответного с сообщения).
- 12 Client T_Data.req: после запуска таймера S3Client в клиенте это вызывает передачу с сообщения запроса TesterPresent (0x3E) с функциональной адресацией, которое не требует ответного с сообщения каждый раз, когда таймер S3Client истекает.
- 13 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с воет транспорт/сетевой уровень клиент еще раз запускает с вой таймер S3Client. Это означает, что с сообщение запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.
Сервер T_Data.ind: любое с сообщение запроса TesterPresent (0x3E), полученное во время активированного таймера S3Server, перезапускает таймер S3Server.
- 14 Client T_Data.req: как только на клиенте запускается таймер S3Client, это вызывает передачу функционально адресованного с сообщения запроса TesterPresent (0x3E), которое не требует ответного с сообщения каждый раз, когда истекает время ожидания таймера S3Client.
- 15 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с воет транспорт/сетевой уровень клиент еще раз запускает с вой таймер S3Client. Это означает, что с сообщение запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.
Сервер T_Data.ind: любое с сообщение запроса TesterPresent (0x3E), полученное во время активированного таймера S3Server, перезапускает таймер S3Server.

Рисунок 11 — Физическая связь во время с еанса не по умолчанию — функционально адресовано TesterPresent

8.1.4.2 Сообщение TesterPresent (0x3E) с физическим адресом

На рис. 12 графически показана обработка с интронизацией в клиенте и на сервере при выполнении физической связи во время сеанса, отличного от сеанса по умолчанию (например, ProgrammingSession), и использовании сообщения запроса TesterPresent (0x3E) с физической адресацией, которое требует ответного сообщения от сервера для сохранения диагностики. Сеанс активен в случае отсутствия какой-либо другой диагностической службы.



ИСО 14229-2:2013(E)

Ключ

- 1 Клиент T_Data.req: диагностическое приложение отправляет с сообщение запроса DiagnosticSessionControl (0x10) на транспортный/сетевой уровень.
Транспортный/сетевой уровень передает с сообщение запроса на сервер.
- 2 Клиент T_Data.con: транспортный/сетевой уровень передает диагностическому приложению получение с сообщения запроса DiagnosticSessionControl (0x10). Теперь применяется время отклика PClient, как описано в 8.1.2. Сгенерированный T_Data.con в клиенте вызывает запуск таймера S3Client (таймера сеанса). Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.
Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса DiagnosticSessionControl (0x10). Теперь применяется время ответа P2Server, как описано в 8.1.2.
- 3 Сервер T_Data.req: диагностическое приложение выдает положительное ответное с сообщение DiagnosticSessionControl (0x10) на транспортный/сетевой уровень. Для приведенного рисунка предполагается, что клиент требует ответа от сервера.
- 4 Сервер T_Data.con: по завершении передачи ответного с сообщения с сервера через T_Data.con. Теперь сервер запускает с вой таймер S3Server, который запускает активированный нестандартный сеанс активным до тех пор, пока не истечет время ожидания. Клиент несет ответственность за то, чтобы таймер S3Server был сброшен до истечения времени ожидания, чтобы сервер оставался в сеансе, отличном от сеанса по умолчанию.
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения. Клиент останавливает таймер PClient.
- 5 Клиент T_Data.req: диагностическое приложение отправляет новое с сообщение запроса на транспортный/сетевой уровень. В каждый раз, когда клиент передает с сообщение запроса на сервер (включая с сообщение TesterPresent (0x3E) с физическим адресом), он останавливает с вой таймер S3Client.
- 6 Сервер T_Data.SOM.ind: транспортный/сетевой уровень выдает диагностическому приложению прием с сообщения StartOfMessage, если интерфейс T_Data.SOM.ind поддерживается транспортным/сетевым уровнем. Прием StartOfMessage с сообщения запроса останавливает таймер S3Server на сервере.
- 7 Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса.
Клиент T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса. В случае, когда клиенту не требуется ответное с сообщение, он должен запустить с вой таймер S3Client, когда получит подтверждение завершения с сообщения запроса, которое указывается через T_Data.con. В этом случае сервер запускает с вой таймер S3Server после завершения запрошенного действия. Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.
- 8 Сервер T_Data.req: диагностическое приложение выдает положительное ответное с сообщение на транспорт/сеть.
- 9 Клиент T_Data.SOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение StartOfMessage, если интерфейс T_Data.SOM.ind поддерживается транспортным/сетевым уровнем. Клиент останавливает таймер PClient.
- 10 Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения.
Сгенерированный T_Data.ind в клиенте вызывает запуск таймера S3Client (таймера сеанса).
Сервер T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения. Теперь сервер запускает с вой таймер S3Server.
- 11 Клиент T_Data.req: в случае, если клиент не отправил с сообщение с диагностическим запросом до истечения времени ожидания S3Client, тогда тайм-аут таймера S3Client заставляет клиента передать с сообщение запроса TesterPresent (0x3E) с физическим адресом.
- 12 Сервер T_Data.ind: прием с сообщения запроса TesterPresent (0x3E) указывается на сервере через T_Data.ind. Это заставляет сервер останавливать с вой таймер S3Server.
Теперь применяется время отклика, как описано в 8.1.2.
Клиент T_Data.con: завершение с сообщения запроса TesterPresent (0x3E) указывается в клиенте через T_Data.con. Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.
- 13 Сервер T_Data.req: диагностическое приложение выдает ответное с сообщение TesterPresent (0x3E) на транспорт/сеть.
- 14 Клиент T_Data.ind: завершение ответного с сообщения TesterPresent (0x3E) указывается в клиенте через T_Data.ind, который заставляет клиента запустить с вой таймер S3Client.
Сервер T_Data.con: завершение ответного с сообщения TesterPresent (0x3E) указывается на сервере через T_Data.con, что заставляет сервер запустить с вой S3Server. В случае, когда клиенту не требуется ответное с сообщение, он должен запустить с вой таймер S3Client, когда получит подтверждение завершения с сообщения запроса TesterPresent (0x3E), которое указывается через T_Data.con. Сервер запустит с вой таймер S3Server, когда завершит запрошенное действие. Для простоты на рисунке показано, что требуется ответ. Клиент останавливает таймер PClient.
- 15 Клиент T_Data.req: в случае, если клиент не отправил с сообщение с диагностическим запросом до истечения времени ожидания S3Client, тогда тайм-аут таймера S3Client заставляет клиента передать с сообщение запроса TesterPresent (0x3E) с физическим адресом.
- 16 Сервер T_Data.ind: прием с сообщения запроса TesterPresent (0x3E) указывается на сервере через T_Data.ind. Это заставляет сервер останавливать с вой таймер S3Server.
Теперь применяется время отклика, как описано в 8.1.2.
Клиент T_Data.con: завершение с сообщения запроса TesterPresent (0x3E) указывается в клиенте через T_Data.con. Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.
- 17 Сервер T_Data.req: диагностическое приложение выдает ответное с сообщение TesterPresent (0x3E) на транспорт/сеть.

18 Клиент T_Data.ind: завершение ответного сообщения TesterPresent (0x3E) указывается яв клиенте через T_Data.ind, который заставляет клиент запустить с вой таймер S3Client.

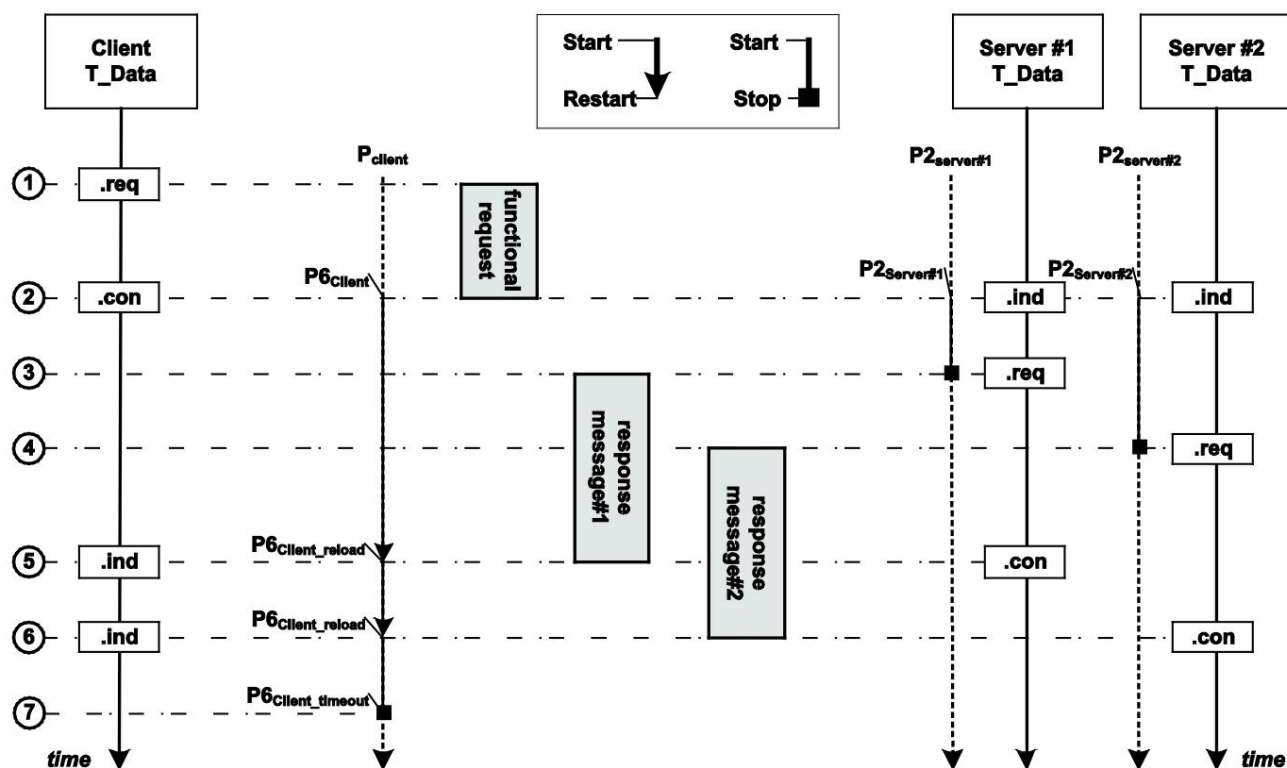
Сервер T_Data.con: завершение ответного сообщения TesterPresent (0x3E) указывается яв сервере через T_Data.con, что заставляет сервер запустить с вой S3Server. В случае, когда клиенту не требуется ответное сообщение, он должен запустить с вой таймер S3Client, когда получит подтверждение завершения сообщения запроса TesterPresent (0x3E), которое указывается я через T_Data.con. Сервер запустит с вой таймер S3Server, когда завершит запрошенное действие. Для простоты на рисунке показано, что требуется я ответ.

Рисунок 12 — Физическая связь во время сеанса по умолчанию — с физическим адресом TesterPresent

8.2 Функциональная связь

8.2.1 Функциональная связь во время defaultSession – без SOM.ind

На рис. 13 графически показана обработка синхронизации на клиенте и двух серверах для сообщения запроса с функциональной адресацией во время сеанса по умолчанию. С точки зрения сервера нет никакой разницы в обработке синхронизации по сравнению с сообщением запроса с физической адресацией, но клиент должен обрабатывать синхронизацию иначе, чем при физической связи.



Ключ

- 1 Клиент T_Data.req: диагностическое приложение выдает функционально адресованное сообщение запроса на транспортный/сетевой уровень.
- 2 Все серверы T_Data.ind: транспортный/сетевой уровень выдает диагностическому приложению завершение сообщения запроса. Все серверы запускают таймер P2Server, используя значение P2Server = P2Server_max.
Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения сообщения запроса. Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P6Client = P6Client_max. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети (например, связь через шлюзы, полосу пропускания шины и т. д.). Для простоты на рисунке предполагается, что клиент и сервера находятся в одной сети.
- 3 Сервер №1 T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный/сетевой слой внутри P2Server. Сервер останавливает таймер P2Server.

4 Сервер #2 T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный/сетевой уровень в P2Server_max. Сервер останавливает таймер P2Server.

5 Сервер #1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения

Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Это заставляет клиента перезапустить свой таймер PClient, используя значение перезагрузки по умолчанию P6Client_max.

6 Сервер #2 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения

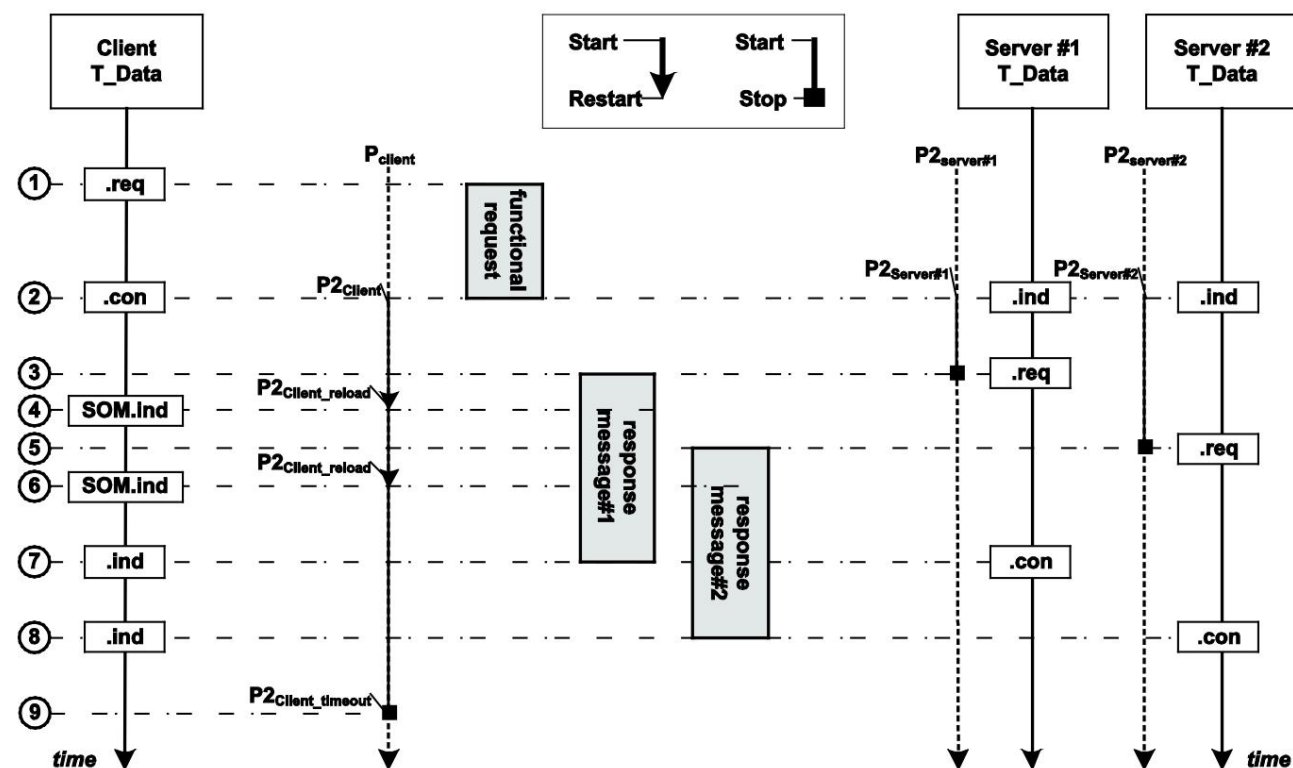
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Это заставляет клиента перезапустить свой таймер PClient, используя значение перезагрузки по умолчанию P6Client_max.

7 Клиент: это указание для клиента, что дальнейших ответных сообщений не ожидается и он может продолжить дальнейшие запросы.

Рисунок 13 — Функциональная связь во время сеанса по умолчанию — без SOM.ind

8.2.2 Функциональная связь во время defaultSession – с SOM.ind

На рис. 14 графически показана обработка синхронизации на клиенте и двух серверах для сообщения запроса с функциональной адресацией во время сеанса по умолчанию. С точки зрения сервера нет никакой разницы в обработке синхронизации по сравнению с сообщением запроса с физической адресацией, но клиент должен обрабатывать синхронизацию иначе, чем при физической связи.



Ключ

- 1 Клиент T_Data.req: диагностическое приложение выдает функционально адресованное сообщение запроса на транспортный/сетевой уровень.
- 2 Все серверы T_Data.ind: транспортный/сетевой уровень выдает диагностическому приложению завершение сообщения запроса. Все серверы запускают таймер P2Server, используя значение P2Server = P2Server_max.
Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети транспортного средства (например, связь через шлюзы, полосу пропускания шины и т. д.). Для простоты на рисунке предполагается, что клиент и серверы являются одной сетью.
- 3 Сервер #1 T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный/сетевой уровень внутри P2Server. Сервер #1 останавливает таймер P2Server.
- 4 Клиент T_DataSOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение сообщения StartOfMessage. Это вызовет

клиент для перезапуска с воер таймера PClient, используя значение перезагрузки по умолчанию PClient_max.

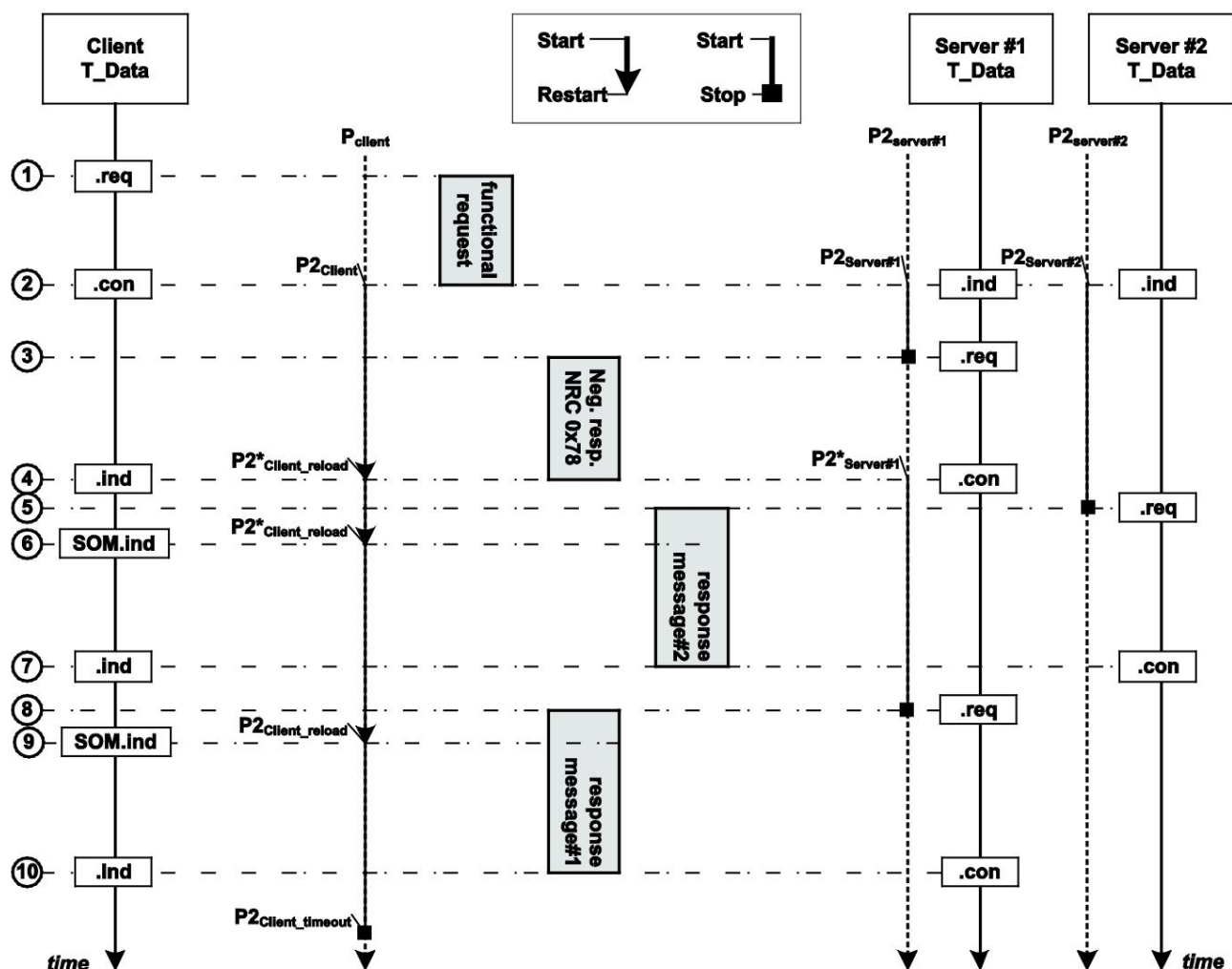
- 5 Сервер #2 T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный/сетевой уровень в P2Server_max. Сервер №2 останавливает таймер P2Server.
- 6 Клиент T_DataSOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение сообщения StartOfMessage. Это заставляет клиента перезапустить с воер таймер PClient, используя значение перезагрузки по умолчанию PClient_max.
- 7 Сервер #1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
- 8 Сервер #2 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
- 9 Клиент: это указание для клиента, что дальнейших ответных сообщений не ожидается и он может продолжить дальнейшие запросы.

Рисунок 14 — Функциональная связь во время сеанса по умолчанию — с SOM.ind

8.2.3 Функциональная связь во время defaultSession с улучшенным временем отклика — с SOM.ind

На рис. 15 графически показана обработка синхронизации на клиенте и двух серверах для сообщения запроса с функциональной адресацией во время сеанса по умолчанию, где один сервер запрашивает расширенную синхронизацию ответа посредством отрицательного ответного сообщения, включая отрицательный код ответа 0x78.

С точки зрения сервера нет никакой разницы в обработке синхронизации по сравнению с сообщением запроса с физической адресацией, которое требует увеличенного времени ответа, но клиент должен обрабатывать синхронизацию иначе, чем при физической связи.



Ключ

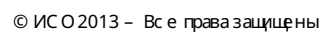
- 1 Клиент T_Data.req: диагностическое приложение выдает функционально адресованное сообщение запроса на транспортный/сетевой уровень.
- 2 Все серверы T_Data.ind: транспортный/сетевой уровень выдает диагностическому приложению завершение сообщения запроса. Все серверы запускают таймер P2Server, используя значение P2Server = P2Server_max.
Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max. Список ожидания NRC 0x78 пуст. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети транспортного средства (например, связь через шлюзы, полосу пропускания и т. д.). Для простоты на рисунке предполагается, что клиент и серверы находятся в одной сети.
- 3 Сервер №1 T_Data.req: диагностическое приложение не имеет готового положительного ответа на сообщение и выдает отрицательное ответное сообщение с NRC = 0x78 с помощью T_Data.req на транспортный/сетевой уровень в P2Server. Сервер №1 останавливает таймер P2Server. Если какой-либо из адресованных серверов не может предоставить запрошенную информацию в течение времени ответа P2Server, он может запросить расширенное окно времени ответа, отправив сообщение с отрицательным ответом, включая код отрицательного ответа 0x78.
- 4 Сервер №1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Сервер №1 запускает таймер P2Server, используя значение P2Server#1 = P2Server_max (улучшенное время по умолчанию).
Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению получение ответного сообщения. Клиент перезагружает таймер PClient с значением P2Client = P2Client_max (улучшенное время по умолчанию). Запись, содержащая адрес ответного сообщения, добавляется в список ожидания NRC 0x78.
- 5 Сервер №2 T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный/сетевой уровень. Сервер №2 останавливает таймер P2Server.
- 6 Клиент T_Data.SOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение сообщения StartOfMessage. Это заставляет клиента перезапустить свой таймер PClient, используя значение перезагрузки по умолчанию P2Client (расширенное время по умолчанию).
- 7 Сервер №2 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.

- Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
- 8 Сервер №1 T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспорт/сеть.
 Слой внутри P2*Server. Сервер №1 останавливает таймер P2Server.
- 9 Клиент T_DataSOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение сообщения StartOfMessage. Запись, соответствующая адресу ответного сообщения, удаляется из списка ожидающих обработки, и теперь этот список пуст. Это означает, что дальнейших ответных сообщений не ожидается. Это заставляет клиента перезапустить свой таймер PClient, используя значение перезагрузки по умолчанию P2Client.
- 10 Сервер #1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
- Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.

Рисунок 15 — Функциональная коммуникация во время сеанса по умолчанию — улучшенное время отклика — с SOM.ind

8.2.4 Функциональная связь во время сеанса по умолчанию — с SOM.ind

На рисунке 16 графически показана обработка информации в клиенте и двух серверах для сообщения запроса с функциональной адресацией во время сеанса, отличного от сеанса по умолчанию (например, сеанс программирования), где один сервер запрашивает расширенную информацию ответа посредством отрицательного ответного сообщения, включая шаг кода отрицательного ответа 0x78.



Ключ

а Любой testerPresent, полученный во время отключения от таймера S3Server, будет игнорироваться сервером.

1 Клиент T_Data.req: диагностическое приложение отправляет с сообщения запроса DiagnosticSessionControl (0x10) с функциональным адресом на транспортный/сетевой уровень. Транспортный/сетевой уровень передает с сообщения запроса на серверы.

2 Клиент T_Data.con: транспортный/сетевой уровень передает диагностическому приложению получение с сообщения запроса DiagnosticSessionControl (0x10). Теперь применяется время отклика PClient, как описано в 8.1.2 и 8.1.3. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети транспортно-осредств (например, с вью через шлюзы, полосу пропускания шины и т. д.). Для простоты на рисунке предполагается, что клиент и серверы расположены в одной сети. Сгенерированный T_Data.con в клиенте вызывает запуск таймера S3Client (таймера сеанса).

All Server T_Data.ind: транспортный/сетевой уровень выдает диагностическому приложению завершение с сообщения запроса DiagnosticSessionControl (0x10). Все серверы запускают таймер P2Server, используя значение P2Server = P2Server_max. Теперь применяется время ответа P2Server, как описано в 8.1.2 и 8.1.3.

3 Сервер №1 T_Data.req: диагностическое приложение выдает положительное ответное с сообщения DiagnosticSessionControl (0x10) на транспортный/сетевой уровень в P2Server_max. Сервер №1 останавливает таймер P2Server. Для приведенного рисунка предполагается, что клиент требует ответа от сервера.

4 Сервер №1 T_Data.con: завершение передачи положительного ответного с сообщения DiagnosticSessionControl (0x10) указывается на сервере №1 через T_Data.con. Теперь сервер №2 запускает с таймером S3Server, который поддерживает активированный сеанс по умолчанию активным до тех пор, пока не истечет время ожидания. Клиент несет ответственность за то, чтобы таймер S3Server был сброшен до истечения времени ожидания, чтобы сервер №1 оставался сеансом, отличным от сеанса по умолчанию.

Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения. Это заставляет клиента перезапустить с таймером PClient, используя значение перезагрузки по умолчанию P2Client.

5 Сервер №2 T_Data.req: диагностическое приложение выдает положительное ответное с сообщения DiagnosticSessionControl (0x10) на транспортный/сетевой уровень в P2Server. Сервер №2 останавливает таймер P2Server. Для приведенного рисунка предполагается, что клиент требует ответа от сервера.

6 Сервер №2 T_Data.con: завершение передачи положительного ответного с сообщения DiagnosticSessionControl (0x10) указывается на сервере №2 через T_Data.con. Теперь сервер №2 запускает с таймером S3Server, который поддерживает активированный сеанс по умолчанию активным до тех пор, пока не истечет время ожидания. Клиент несет ответственность за то, чтобы таймер S3Server был сброшен до истечения времени ожидания, чтобы сервер №2 оставался сеансом, отличным от сеанса по умолчанию.

Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения. Это заставляет клиента перезапустить с таймером PClient, используя значение перезагрузки по умолчанию P2Client.

7 Клиент T_Data.req: после запуска таймера S3Client в клиенте это вызывает передачу с сообщения запроса TesterPresent (0x3E) с функциональной адресацией, которое не требует ответного с сообщения каждый раз, когда истекает время ожидания таймера S3Client.

8 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с своего транспортного/сетевого уровня клиент еще раз запускает с таймером S3Client. Это означает, что с сообщения запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.

All Server T_Data.ind: любое с сообщения запроса TesterPresent (0x3E), полученное во время активированного таймера S3Server, перезагружает таймер S3Server.

9 Клиент: это указание для клиента о том, что дальнейших ответных с сообщений не ожидается и он может продолжить дальнейшие запросы.

10 Клиент T_Data.req: диагностическое приложение выдает функционально адресованное с сообщения запроса на транспортный/сетевой уровень.

11 Все серверы T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса. Все серверы запускают таймер P2Server, используя значение P2Server = P2Server_max. Каждый раз, когда сервер обрабатывает какую-либо диагностическую службу, он останавливает с таймером S3Server.

Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения с сообщения запроса. Клиент запускает с таймером PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max. Значение таймера PClient должно учитывать любую задержку, связанную с архитектурой сети транспортно-осредств (например, с вью через шлюзы, полосу пропускания шины и т. д.). Для простоты на рисунке предполагается, что клиент и серверы расположены в одной сети.

12 Сервер №1 T_Data.req: диагностическое приложение не имеет готового положительного ответного с сообщения и выдает отрицательное ответное с сообщения с NRC = 0x78 с помощью T_Data.req на транспортный/сетевой уровень в P2Server. Сервер №1 останавливает таймер P2Server. Если какой-либо из адресованных серверов не может предоставить запрошенную информацию в течение времени ответа P2Server, он может запросить расширенное окно времени ответа, отправив с сообщения с отрицательным ответом, включающим отрицательный ответ 0x78.

13 Сервер №1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного с сообщения. Сервер №1 запускает таймер P2Server, используя значение P2*Server = P2*Server_max (улучшенное время по умолчанию).

Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению получение ответного с сообщения. Клиент перезагружает таймер PClient с значением P2*Client = P2*Client_max (улучшенное время по умолчанию). Запись, содержащая адрес ответного с сообщения, добавляется в список ожидания NRC 0x78.

14 Сервер №2 T_Data.req: диагностическое приложение подготовило ответное с сообщения и выдает T_Data.req на транспортный/сетевой уровень внутри P2Server. Сервер №2 останавливает таймер P2Server.

ИСО 14229-2:2013(E)

- 15 Клиент T_DataSOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение с сообщения StartOfMessage. Это вызовет клиент для перезапуска своего таймера PClient, используя значение перезагрузки по умолчанию P2*Client (расширенное время по умолчанию).
- 16 Client T_Data.req: как только на клиенте запускается таймер S3Client, это вызывает передачу функционально адресованного сообщения запроса TesterPresent (0x3E), которое не требует ответа с сообщения каждый раз, когда истекает время ожидания таймера S3Client.
- 17 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с своего транспортного/сетевого уровня клиент еще раз запускает свой таймер S3Client. Это означает, что с сообщения запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.
- All Server T_Data.ind: любое сообщение запроса TesterPresent (0x3E), полученное во время активированного таймера S3Server, перезагружает таймер S3Server. Любое TesterPresent, полученный во время отключенного таймера S3Server, будет игнорироваться сервером(ами).
- 18 Клиент T_Data.ind: когда транспортный/сетевой уровень отправляет диагностическому приложению завершение ответа с сообщения.
- Сервер № 2 T_Data.con: когда служба диагностики полностью обработана, сервер № 2 перезапускает свой таймер S3Server. Это означает, что любой диагностический сервис, включая TesterPresent (0x3E), сбрасывает таймер S3Server. Диагностическая служба считывает работающий в любое время между началом приема сообщения запроса (прием T_DataSOM.ind или T_Data.ind) и завершением передачи окончательного ответа с сообщения, когда требуется ответное сообщение, или завершение любого действия, вызванного запросом, когда ответное сообщение не требуется (достигнут момент времени, который может вызвать запуск ответа с сообщения).
- 19 Сервер №1 T_Data.req: диагностическое приложение подготавливает ответное сообщение и выдает T_Data.req на транспорт/сеть.
- сервис внутри P2*Server. Сервер №1 останавливает таймер P2Server.
- 20 Клиент T_DataSOM.ind: транспортный/сетевой уровень передает диагностическому приложению получение с сообщения StartOfMessage. Запись, с ответствующий адрес у ответа с сообщения, удалится из списка ожидающих обработки, и теперь этот список пуст. Это означает, что дальнейших ответов с сообщений не ожидается. Это заставляет клиента перезапустить свой таймер PClient, используя значение перезагрузки по умолчанию P2Client.
- 21 Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответа с сообщения.
- Сервер №1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответа с сообщения. Теперь сервер №1 запускает свой таймер S3Server.
- 22 Client T_Data.req: каждый раз, когда таймер S3Client истекает, вызывает передачу с сообщения запроса TesterPresent (0x3E) с функциональной адресацией, которое не требует ответа с сообщения.
- 23 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с своего транспортного/сетевого уровня клиент еще раз запускает свой таймер S3Client. Это означает, что с сообщения запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.
- All Server T_Data.ind: любое сообщение запроса TesterPresent (0x3E), полученное во время активированного таймера S3Server, перезагружает таймер S3Server.
- 24 Client T_Data.req: после запуска таймера S3Client в клиенте это вызывает передачу с сообщения запроса TesterPresent (0x3E) с функциональной адресацией, которое не требует ответа с сообщения каждый раз, когда таймер S3Client истекает.
- 25 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с своего транспортного/сетевого уровня клиент еще раз запускает свой таймер S3Client. Это означает, что с сообщения запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.
- All Server T_Data.ind: любое сообщение запроса TesterPresent (0x3E), полученное во время активированного таймера S3Server, перезагружает таймер S3Server.

Рисунок 16 — Функциональная связь во время сеанса по умолчанию — с SOM.ind

Обработка с интронизацией PClient и P2Server идентична обработке, описанной в 8.1.2 и 8.1.3, за исключением того, что значения перезагрузки на стороне клиента и результирующее время, которое сервер должен отправить, могут отличаться. Это особенно на переходе в сеанс, отличный от сеанса по умолчанию, где могут применяться другие параметры с интронизацией PClient (с м. службу DiagnosticSessionControl (0x10) в ISO 14229-1 для получения подробной информации о том, как параметры времени общаются клиенту).

8.3 Минимальное время между сообщениями запроса клиента

Минимальное время между сообщениями запроса, передаваемыми клиентом, требуется для того, чтобы обеспечить интерпретацию служебных данных на основе опроса на сервере. Основываясь на обычной функциональности, сервер может обрабатывать сообщения диагностических запросов с скоростью планирования, зависящей от проекта (например, 10 мс). Время для планировщика интерпретации данных диагностической службы должно быть меньше, чем требование к производительности P2Server, чтобы соответствовать требованиям к серверу, как указано в 8.1.1, 8.1.2 и 8.1.3.

Параметр времени для минимального времени между сообщениями запроса делится на следующие два параметра времени.

- P3Client_Func : этот параметр синхронизации применяется к любому сообщению запроса с функциональной адресацией, поскольку может быть случаем, когда сервер не обязан отвечать на сообщения запроса с функциональной адресацией, если он не поддерживает запрошенные данные.
- P3Client_Phys : этот параметр синхронизации применяется к любому физическому адресу сообщения запроса, на которое сервер не должен передавать ответ (suppressPosRspMsgIndicationBit = TRUE).

В случае физической связи, когда ответ требуется серверу, клиент может передать следующий запрос сразу после полного приема последнего ответного сообщения, потому что сервер полностью ответил на запрос — это означает, что запрос полностью обрабатывается сервером.

Рисунок 17 графически изображает пример проблемы, которая может возникнуть во время функциональной связи, когда клиент передает следующий запрос сразу после того, как он определил, что все ожидаемые серверы ответили на предыдущее сообщение запроса.

Этот сценарий применяется не только к запросам с функциональной адресацией, но и к запросам с физической адресацией, когда клиент не хочет получать какое-либо ответное сообщение (suppressPosRspMsgIndicationBit = TRUE).

Для обработки описанных сценариев для клиента определяются минимальные значения времени P3Client_Phys и P3Client_Func между окончанием сообщения запроса с физической или функциональной адресацией и началом нового сообщения запроса с физической или функциональной адресацией.

- a) Значение P3Client_Phys будет идентично P2Server_max для физического адресованного сервера. Время применяется к любому физическому адресу сообщению запроса в любом диагностическом сеансе (сеансе по умолчанию и не по умолчанию) и в случае, если сервер не требует ответа.

Таймер P3Client_Phys запускается в клиенте каждый раз, когда физическое сообщение запроса без ответа успешно передается машине, что указывается через T_Data.con в клиенте.

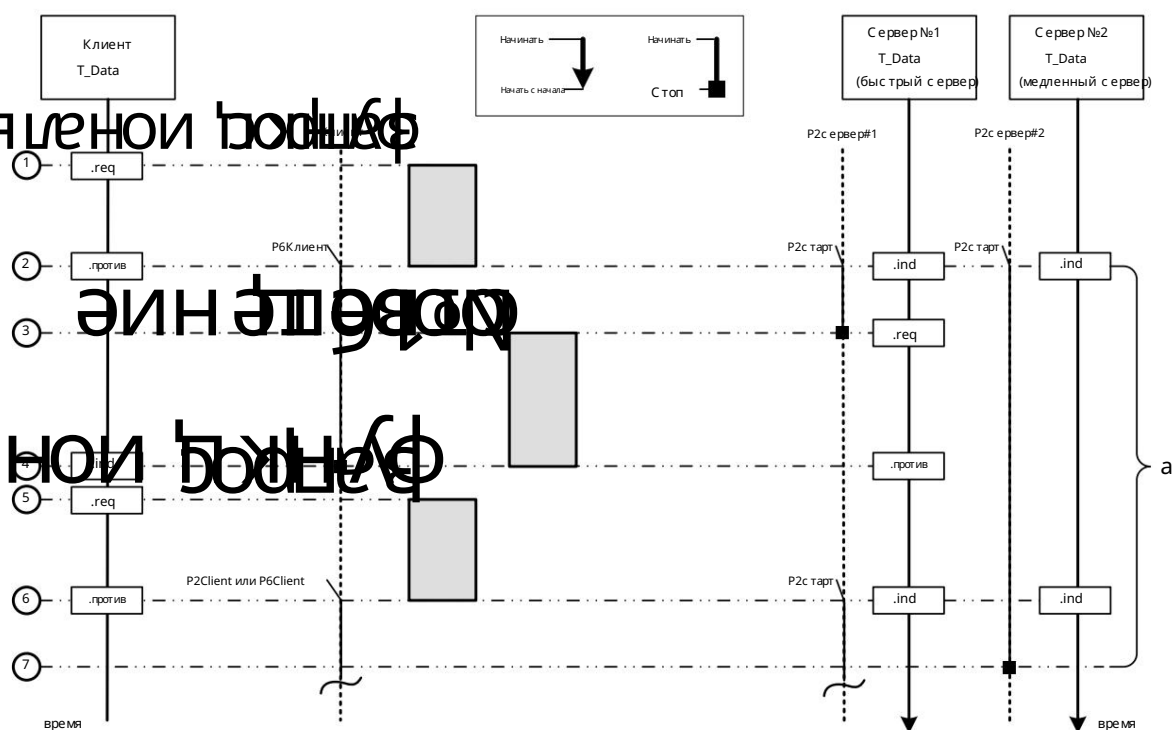
Когда клиент хочет передать новое сообщение запроса с физическим адресом после предыдущего запроса, который был полностью обработан, это разрешено только в том случае, если таймер P3Client_Phys больше не активен в то время, когда клиент хочет передать сообщение запроса с физическим адресом. В случае, если P3Client_Phys все еще будет активен в момент времени, когда клиент хотел бы передать новое сообщение запроса с физическим адресом, тогда передача должна быть отложена до истечения времени ожидания P3Client_Phys.

- b) Значение P3Client_Func будет максимальным (наихудшим) значением P2Server_max всех функционально адресованных серверов для любого сообщения запроса с функциональной адресацией в любом диагностическом сеансе (сеансе по умолчанию и не по умолчанию).

Таймер P3Client_Func запускается в клиенте каждый раз, когда функционально адресованное сообщение запроса с требуемым ответом или без ответа успешно передается машине, что указывается через T_Data.con в клиенте. Когда клиент хочет передать новое сообщение запроса с функциональной адресацией после предыдущего запроса, который был полностью обработан, это разрешено только в том случае, если таймер P3Client_Func больше не активен в то время, когда клиент хочет передать сообщение запроса с функциональной адресацией. В случае, если P3Client_Func все еще будет активен в момент времени, когда клиент хотел бы передать новое сообщение запроса с функциональной адресацией, тогда передача должна быть отложена до тех пор, пока не истечет время P3Client_Func.

ПРИМЕЧАНИЕ «Полностью обработано» означает, что либо ответ не получен, если ответ не требуется либо все ожидаемые ответы на функционально адресованный запрос получены, если известны отвечающие серверы и требуются ответы, либо произошел тайм-аут PClient, если отвечающие серверы неизвестны и требуются ответы.

Требование к серверу состоит в том, что он должен начинаться с своего ответного сообщения в P2Server. Это означает, что скорость интерпретации диагностических данных сервера должна быть меньше, чем P2Server_max.

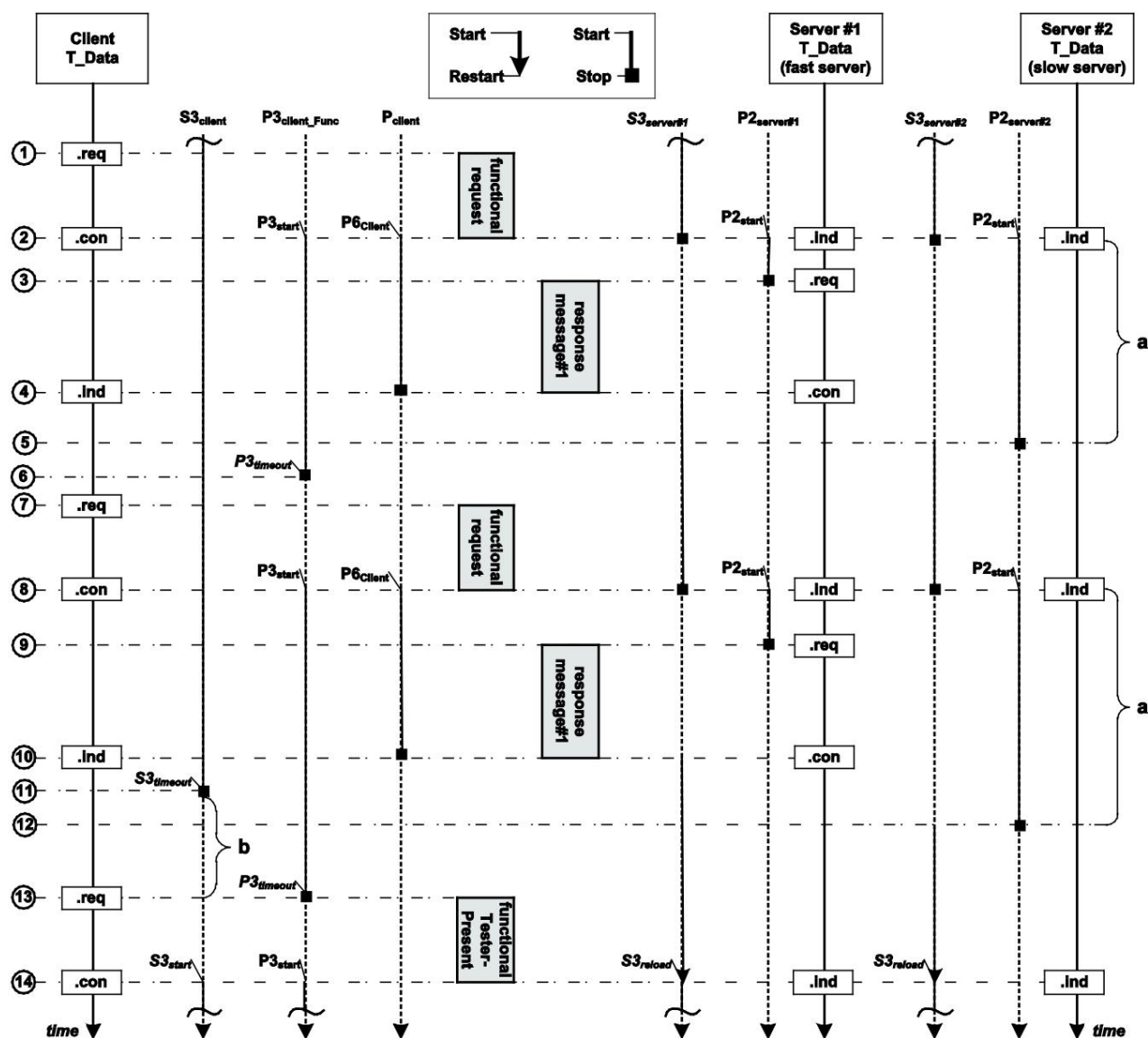


Ключ

- а Скорость интерпретации данных диагностической службы
- 1 сервера № 2. Клиент T_Data.req: диагностическое приложение отправляет с сообщение запроса на транспортный/сетевой уровень.
- 2 Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения с сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max.
- All Server T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса. Все серверы запускают таймер P2Server, используя значение по умолчанию P2Server = P2Server_max.
- 3 Сервер №1 T_Data.req: диагностическое приложение подготовило ответное с сообщение и отправляет T_Data.req на транспортный/сетевой уровень в P2Server. Сервер №1 останавливает таймер P2Server. Для сообщения запроса предполагается, что только сервер №1 поддерживает запрошенную информацию, что означает отсутствие ответа от сервера №2. Сервер №1 является быстрым сервером и может немедленно обработать полученное с сообщение запроса и передать свой ответ в P2Server.
- 4 Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Клиент ожидал только ответного сообщения от сервера №1, поэтому он останавливает свой таймер PClient.
- Сервер №1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения.
- 5 Клиент T_Data.req: диагностическое приложение отправляет с сообщение запроса на транспортный/сетевой уровень. Клиент отправит следующий запрос сразу после завершения всех ожидаемых ответных сообщений.
- 6 Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения с сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max (если T_DataSOM.ind поддерживается) или P6Client = P6Client_max.
- Сервер №1 T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса. Сервер №1 запускает таймер P2Server, используя значение по умолчанию P2Server = P2Server_max.
- Сервер №2 T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение с сообщения запроса. Сообщение запроса будет обработано только быстрым сервером №1, поскольку сервер №2 еще не обработал предыдущий запрос. Любое с сообщение запроса, полученное во время интерпретации данных диагностической службы сервера №2, будет игнорироваться сервером №2.
- 7 Сервер №2: Сервер №2 является медленным сервером и интерпретирует полученные запросы на периодической основе (скорость интерпретации данных диагностической службы). В худшем случае последняя проверка выполнения с сообщения запроса выполняется на приеме транспортным/сетевым уровнем с сообщения запроса с функциональной адресацией. Это будет означать, что запрос будет сохранен в буфере и обработан как можно раньше, когда планировщик в следующий раз проверит наличие входящего запроса. Когда сервер №2 обрабатывает запрос, он определяет, что ему не нужно отвечать, поскольку он не поддерживает запрошенную информацию (например, сервер не отправляет с сообщение об отсутствии ответа, поскольку отрицательный код ответа requestOutOfRange всегда подавляется в худшем функционально адресованного запроса). Как показано на рисунке, это произойдет после завершения ответного сообщения с сервера №1 и даже после завершения следующего с сообщения запроса, переданного клиентом.

Рисунок 17 — Пример критической проблемы при слишком ранней передаче следующего запроса

На рис. 18 графически показана обработка синхронизации P3Client_Func для клиента (на основе сценария связи, показанного на рис. 16). Кроме того, на рис. 18 показана обработка сообщения запроса TesterPresent (0x3E) с функциональной адресацией в клиенте в случае, когда таймер P3Client_Func все еще активен, когда истекает время ожидания S3Client (запрос будет отложен до истечения времени ожидания P3Client_Func).



Ключ

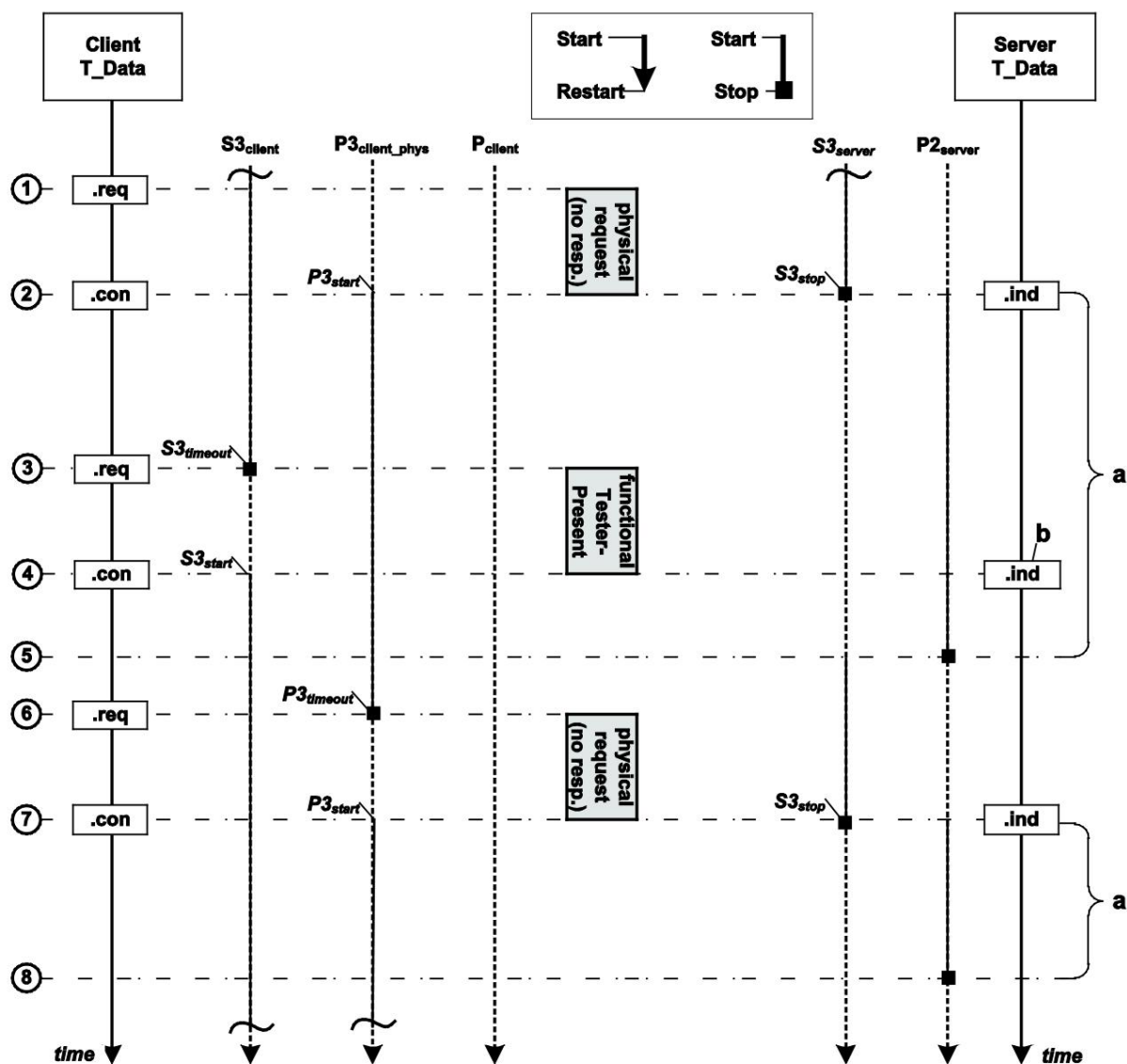
- a: скорость интерпретации диагностических сообщений с сервера №2
- b: функциональный тест на задержку
- 1: Клиент T_Data.req: диагностическое приложение отправляет сообщение запроса на транспортный/сервисный уровень.
- 2: Клиент T_Data.con: транспортный/сервисный уровень выдает диагностическому приложению подтверждение завершения сообщения запроса. Клиент запускает свой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max (если поддерживается T_DataSOM.ind) или P6Client = P6Client_max и, кроме того, свой таймер P3Client_Func.
- 3: Сервер №1 T_Data.req: диагностическое приложение подготовило ответное сообщение и отправляет T_Data.req на транспортный/сервисный уровень в P2Server. Сервер №1 останавливает таймер P2Server. Для сообщения запроса предполагается, что только сервер №1 поддерживает запрошенную информацию, что означает отсутствие ответа от сервера №2. Сервер №1 является занятым сервером и может сразу

ИСО 14229-2:2013(E)

- обрабатывает полученное сообщение запроса и передает с вой ответ в P2Server.
- 4 Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Клиент ожидал только ответного сообщения от сервера №1, поэтому он останавливает с вой таймер PClient.
- Сервер №1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Теперь сервер №1 запускает с вой таймер S3Server.
- 5 Сервер №2: Сервер №2 является медленным сервером и интерпретирует полученные запросы на периодический основе (скорость интерпретации данных диагностической службы). В худшем случае последняя проверка входящего сообщения-запроса выполняется до приема транспортного/ сетевого уровня сообщения-запроса с функциональной адресацией. Это будет означать, что запрос будет сориентирован в буфере и обработан как можно раньше, когда планировщик в следующий раз проверит наличие входящего запроса. Когда сервер №2 обрабатывает запрос, он определяет, что ему не нужно отвечать, поскольку он не поддерживает запрошенную информацию (например, сервер не отправляет сообщение об отсутствии ответа, поскольку отрицательный код ответа requestOutOfRange все равно подавляется в худшем случае функционально адресованного запроса). Сообщение). Теперь сервер №2 останавливает таймер P2Server и запускает с вой таймер S3Server.
- 6 Клиент: даже если клиент получил все ожидаемые ответные сообщения на сообщение запроса с функциональной адресацией, он должен дожидаться истечения времени P3Client_Func, прежде чем ему будет разрешено передать следующий сообщение запроса. В этот момент времени P3Client_Func истекает.
- 7 Клиент T_Data.req: диагностическое приложение отправляет сообщение запроса на транспортный/сетевой уровень. Клиент отправит следующий запрос сразу после завершения всех ожидаемых ответных сообщений.
- 8 Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения сообщения запроса. Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max (если поддерживается T_DataSOM.ind) или P6Client = P6Client_max) и, кроме того, с вой таймер P3Client_Func.
- All Server T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение сообщения запроса. Все серверы запускают таймер P2Server, используя значение по умолчанию P2Server = P2Server_max. Все серверы останавливают таймер S3Server.
- 9 Сервер №1 T_Data.req: диагностическое приложение подготовило ответное сообщение и выдает T_Data.req на транспортный/сетевой уровень в P2Server_max. Сервер №1 останавливает таймер P2Server. Для сообщения запроса предполагается, что только сервер №1 поддерживает запрошенную информацию, что означает отсутствие ответа от сервера №2. Сервер №1 является быстрым сервером и может немедленно обработать полученное сообщение запроса и передать с вой ответ в P2Server.
- 10 Клиент T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Клиент ожидал только ответного сообщения от сервера №1, поэтому он останавливает с вой таймер PClient.
- Сервер №1 T_Data.con: транспортный/сетевой уровень передает диагностическому приложению завершение ответного сообщения. Теперь сервер №1 запускает с вой таймер S3Server.
- 11 Клиент: Таймер S3Client клиента истекает, что вынуждает клиента передать с сообщение запроса TesterPresent (0x3E) с функциональной адресацией, не требуя ответного сообщения от адресованных серверов. В зависимости от ситуации, в которой таймер P3Client_Func все еще активен в этот момент времени, передача TesterPresent (0x3E) должна быть отложена до истечения времени таймера P3Client_Func.
- 12 Сервер №2: Сервер №2 является медленным сервером и интерпретирует полученные запросы на периодический основе (скорость интерпретации данных диагностической службы). В худшем случае последняя проверка входящего сообщения-запроса выполняется до приема транспортного/ сетевого уровня сообщения-запроса с функциональной адресацией. Это будет означать, что запрос будет сориентирован в буфере и обработан как можно раньше, когда планировщик в следующий раз проверит наличие входящего запроса. Когда сервер №2 обрабатывает запрос, он определяет, что ему не нужно отвечать, поскольку он не поддерживает запрошенную информацию (например, сервер не отправляет сообщение об отсутствии ответа, поскольку отрицательный код ответа requestOutOfRange все равно подавляется в худшем случае функционально адресованного запроса). Сообщение). Теперь сервер №2 останавливает таймер P2Server и запускает с вой таймер S3Server.
- 13 Client T_Data.req: Когда время таймера P3Client_Func истекает, функционально адресуемый запрос TesterPresent (0x3E) может быть передан клиентом через T_Data.req.
- 14 Клиент T_Data.con: после индикации завершения передачи с сообщения запроса TesterPresent (0x3E) через T_Data.con с его транспортно-сетевым уровнем клиент еще раз запускает с вой таймер S3Client. Это означает, что с сообщение запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает. Клиент запускает с вой таймер PClient, используя значение перезагрузки по умолчанию P2Client = P2Client_max (если поддерживается T_DataSOM.ind) или P6Client = P6Client_max) и, кроме того, с вой таймер P3Client_Func.
- All Server T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение сообщения запроса. Любое сообщение запроса TesterPresent (0x3E), полученное во время активированного таймера S3Server, перезагружает таймер S3Server.

Рисунок 18 — Минимальное время между сообщениями запроса с функциональной адресацией (P3Client_Func)

Рисунок 19 графически изображает обработку синхронизации P3Client_Phys для клиента. На рисунке показана обработка запроса с физической адресацией, не требующего ответа, и сообщения запроса TesterPresent (0x3E) с функциональной адресацией в клиенте по истечении времени ожидания S3Client.



Ключ

a скорость интерпретации данных диагностических устройств

b любой testerPresent, полученный в время отключения от таймера S3server, может быть проигнорирован сервером

- 1 Клиент T_Data.req: диагностическое приложение выдает сообщение-запрос на транспортный/сетевой уровень, не требующее ответа.
- 2 Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения сообщения-запроса. Клиент запускает свой таймер P3Client_Phys, используя значение перезагрузки по умолчанию P3Client_Phys = P3Client_Phys_max. Ответ не требуется для передачи, поэтому клиенту не нужно запускать свой таймер PClient.
- Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение сообщения-запроса. Сервер запускает таймер P2Server, используя значение по умолчанию P2Server = P2Server_max, и в любом случае, отличное от значения по умолчанию, таймер S3Server теперь останавливается.
- 3 Client T_Data.req: таймер S3Client клиента истекает, что заставляет клиента передать сообщение-запроса TesterPresent (0x3E) с функциональной адресацией, не требуя ответа от адресованных серверов. Предполагается, что таймер P3Client_Func в этот момент времени не активен, это означает, что запрос передается немедленно.
- 4 Клиент T_Data.con: после индикации завершения передачи сообщения-запроса TesterPresent (0x3E) через T_Data.con с его транспортно-сетевого уровня клиент еще раз запускает свой таймер S3Client. Это означает, что сообщение-запроса TesterPresent (0x3E) с функциональной адресацией отправляется периодически каждый раз, когда S3Client истекает.

ИСО 14229-2:2013(E)

Сервер T_Data.ind: любое сообщение запроса TesterPresent (0x3E), полученное во время обработки другого сообщения запроса, может быть проигнорировано сервером, поскольку он уже установил свой таймер S3Server и перезапустит его после полной обработки выполняемой службы.

- 5 Сервер: Сервер периодически интерпретирует полученные запросы (скорость интерпретации данных диагностической службы). Запрос обрабатывается при следующей проверке планировщиком входящих запросов. Завершенное выполнение службы перезапустит таймер S3Server во время логического сеанса, отличного от сеанса по умолчанию, и установит таймер P2Server.
- 6 Клиент T_Data.req: когда время таймера P3Client_Phys на клиенте истекает, клиент может передать следующему сообщению запроса с физической адресацией, отправив T_Data.req на свой транспортный/сетевой уровень.
- 7 Клиент T_Data.con: транспортный/сетевой уровень выдает диагностическому приложению подтверждение завершения сообщения запроса. Теперь клиент снова запускает свой таймер P3Client_Phys. Ответ не требуется для передачи, поэтому клиенту не нужно запускать свой таймер PClient.

Сервер T_Data.ind: транспортный/сетевой уровень передает диагностическому приложению завершение сообщения запроса. Сервер запускает таймер P2Server, используя значение по умолчанию P2Server = P2Server_max, и в любом случае, отличном от значения по умолчанию, таймер S3Server теперь устанавливается.

- 8 Сервер: Сервер периодически интерпретирует полученные запросы (скорость интерпретации данных диагностической службы). Запрос обрабатывается при следующей проверке планировщиком входящих запросов. Завершенное выполнение службы перезапустит таймер S3Server во время логического сеанса, отличного от сеанса по умолчанию, и установит таймер P2Server.

Рисунок 19 — Минимальное время между сообщениями запроса с физической адресацией (P3Client_Phys)

Приложение
А (обязательное)

Интерфейс T_PDU

На рисунке А.1 показан T_PDU (виртуальный PDU) как интерфейс между PDU унифицированных диагностических служб и любым протоколом связи.

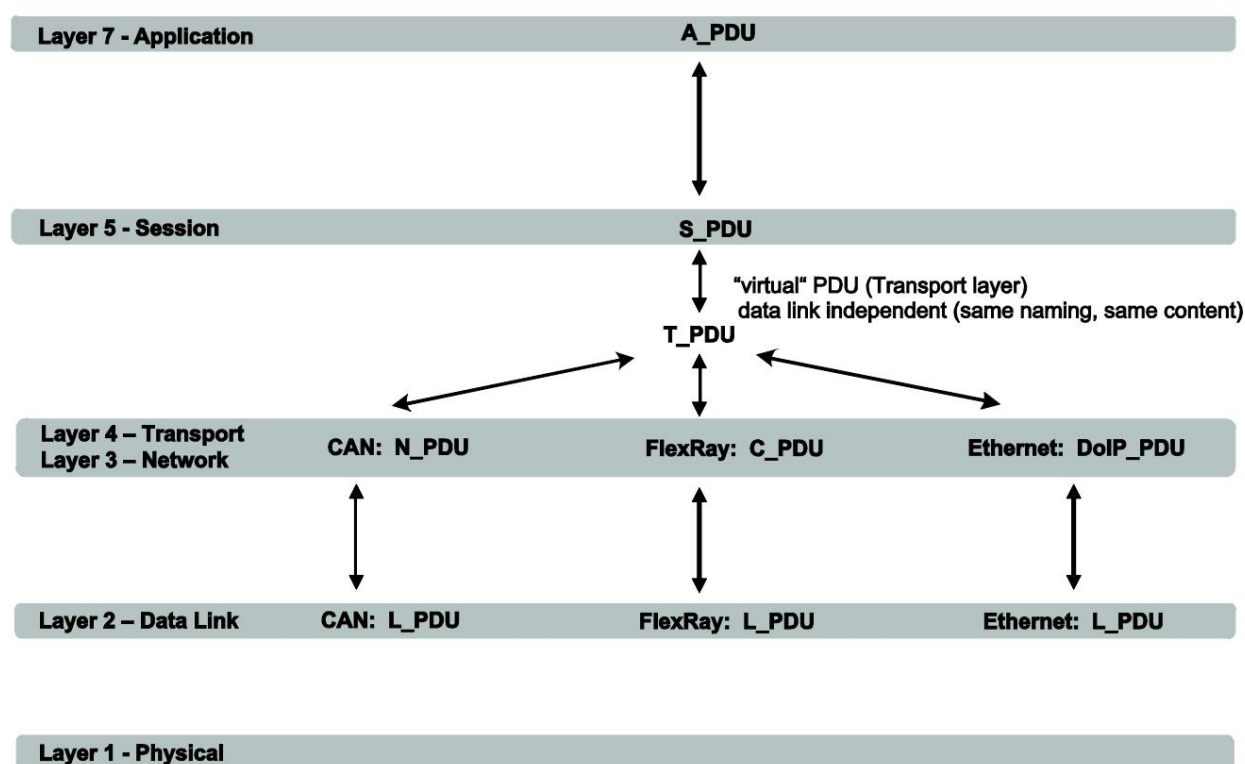


Рисунок А.1 — Интерфейс виртуального PDU T_PDU для любого протокола коммуникации

Приложение
Б (информативное)

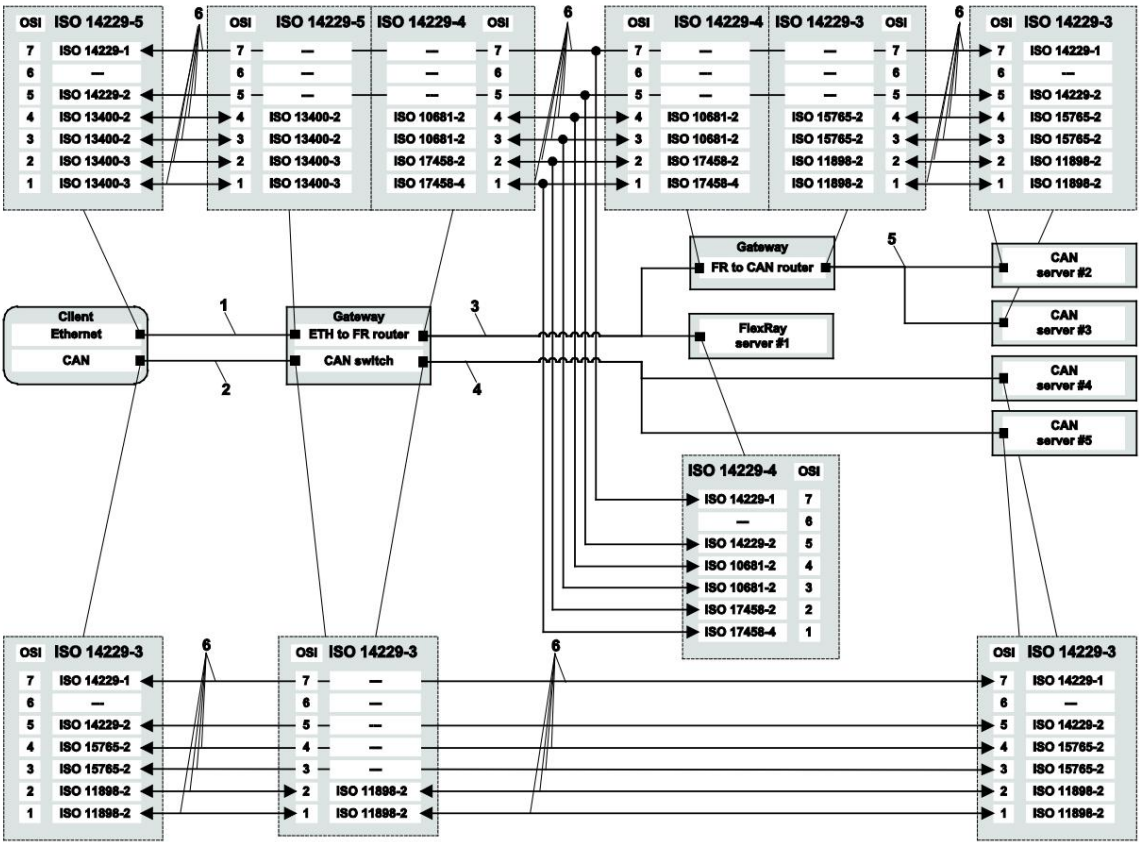
Примеры архитектуры уровня OSI диагностики автомобиля

В.1 Пример шлюза уровня OSI диагностики автомобиля

На рисунке В.1 показан пример архитектуры сети диагностики автомобиля с различными сетевыми технологиями и двумя экземплярами шлюза. «Маршрутизатор Ethernet-FlexRay», реализованный в устройстве шлюза, представляет собой сетевое устройство, которое передает PDU на уровнях 3 и 4 OSI. «Коммутатор CAN», реализованный в устройстве шлюза, представляет собой сетевое устройство, которое передает PDU на уровне 2 OSI.

Обрамленные прямоугольники (пунктирные линии) представляют модель уровня ISO/OSI для различных протоколов, которые должны поддерживаться соответствующим интерфейсом.

Ключи каналов передачи данных 1 и 2 относятся к диагностическим данным автомобиля. Ключи каналов передачи данных с 3 по 5 являются примерами автомобильных коммуникационных сетей, а ключ 5 расположен за вторым шлюзом (маршрутизатор FlexRay-CAN).



- Ключ
- 1 Диагностический Ethernet (ETH)
 - 2 Диагностическая CAN
 - 3 Система связи FlexRay (FR)
 - 4 МОЖЕТ
 - 5 Логическое соединение CAN
 - 6 неприменимо

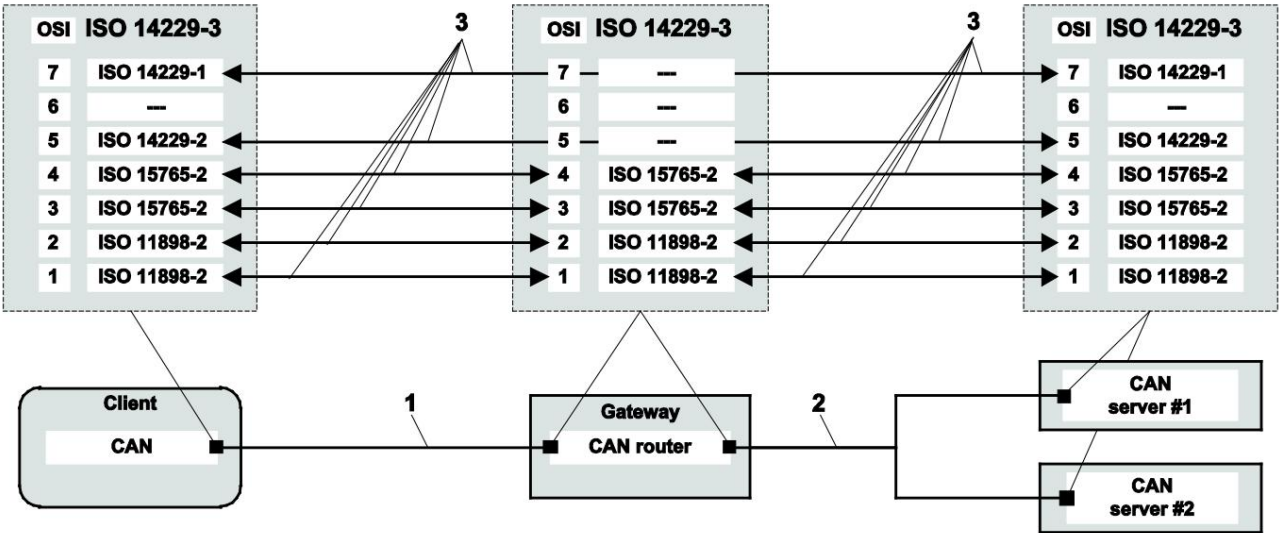
Рисунок В.1 — Пример шлюза уровня OSI для диагностики автомобиля

В.2 Пример маршрутизатора CAN уровня OSI для диагностики автомобиля

На рисунке В.2 показан пример архитектуры диагностической сети транспортного средства «маршрутизатор CAN», реализованный в шлюзе. «CAN-маршрутизатор», реализованный в устройстве шлюза, представляет собой сетевое устройство, которое передает PDU на уровнях 3 и 4 OSI.

Обрамленные прямоугольники (пунктирные линии) представляют модель уровня ISO/OSI для различных протоколов, которые должны поддерживаться с соответствующими интерфейсами.

Ключ канала передачи данных 1 доступен для доступа к диагностическим данным автомобиля. Ключ канала передачи данных 2 является примером автомобильной коммуникационной сети CAN.



Ключ
1 Диагностическая CAN
2 CAN 3
логическое соединение --- не применимо

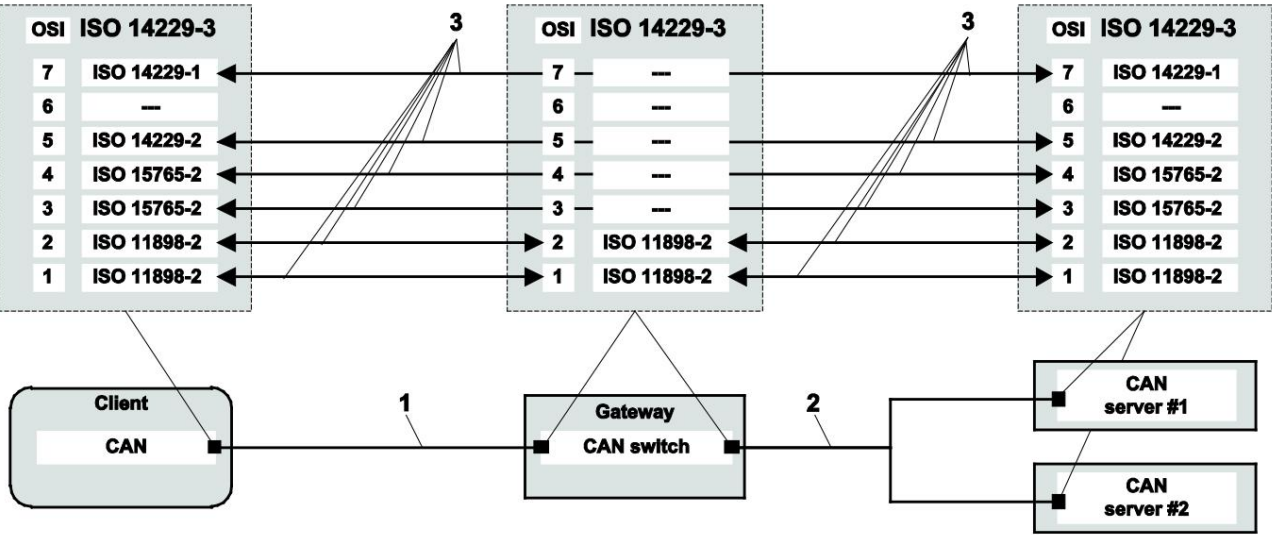
Рисунок В.2 — Пример маршрутизатора CAN уровня OSI для диагностики автомобиля

В.3 Пример переключения CAN уровня OSI диагностик и автомобиля

На рисунке В.3 показан пример архитектуры диагностической сети транспортного средства «переключателем CAN», реализованным в шлюзе. «Коммутатор CAN», реализованный в устройстве шлюза, представляет собой сетевое устройство, которое передает PDU на уровне 2 OSI.

Обрамленные прямоугольники (пунктирные линии) представляют модель уровня ISO/OSI для различных протоколов, которые должны поддерживаться с соответствующим интерфейсом.

Ключ канала передачи данных 1 доступен для двусторонней диагностической связи с автомобилем. Ключ канала передачи данных 2 является примером автомобильной коммуникационной сети CAN.



- Ключ
- 1 Диагностическая CAN
 - 2
 - 3 Логическое
 - соединение CAN неприменимо

Рисунок В.3 — Пример переключения CAN уровня OSI для диагностик и автомобиля

Библиография

- [1] ISO/IEC 7498-1, Информационные технологии. Взаимосвязь открытых систем. Базовая эталонная модель: Базовая модель
- [2] ISO 10681-2, Дорожные транспортные средства. Связь на FlexRay. Часть 2. Услуги и уровни связи
- [3] ISO/IEC 10731, Информационные технологии. Взаимосвязь открытых систем. Базовый справочник. Модель — Соплашения для определения услуг OSI
- [4] ISO 11898-1, Транспортный. Сеть контроллеров (CAN). Часть 1. Канальный уровень и физическая реализация
- [5] ISO 11898-2 Транспортный. Сеть контроллеров (CAN). Часть 2. Устройство для высокоскоростной передачи
- [6] ISO 13400-2, Транспортный. Диагностическая связь по Интернет-протоколу (DoIP). Часть 2: Транспортный протокол и услуги сетевого уровня
- [7] ISO 13400-3, Транспортный. Диагностическая связь по Интернет-протоколу (DoIP). Часть 3. Проводной интерфейс транспортного средства на основе IEEE 802.3
- [8] ISO 14229-3, Транспортный. Унифицированные диагностические службы (UDS). Часть 3. Унифицированные диагностические службы для реализации CAN (UDSonCAN)
- [9] ISO 14229-4, Дорожные транспортные средства. Унифицированные диагностические службы (UDS). Часть 4. Унифицированные диагностические службы при реализации FlexRay (UDSonFR)
- [10] ISO 14229-5 Транспортный. Унифицированные диагностические услуги (UDS). Часть 5. Унифицированная диагностика услуг и внедрению интернет-протокола (UDSonIP)1)
- [11] ISO 14229-6 Транспортный. Унифицированные диагностические услуги (UDS). Часть 6: UDS на линии K реализации (UDSonK-Line)1)
- [12] ISO 14230-1 Транспортный. Диагностическая связь по K-линии (DOK-Line). Часть 1. Физические характеристики. Слой
- [13] ISO 14230-2 Транспортный. Диагностическая связь по K-линии (DOK-Line). Часть 2. Канал передачи данных. Слой
- [14] ISO 15765-2, Дорожные транспортные средства. Диагностическая связь по локальной сети контроллеров (DoCAN) — Часть 2: Транспортный протокол и службы сетевого уровня
- [15] ISO 17458-2 Транспортный. Система связи FlexRay. Часть 2. Канальный уровень. Спецификация1)
- [16] ISO 17458-4 Транспортный. Система связи FlexRay. Часть 4. Электрический физический уровень. Спецификация1)
- [17] ISO 27145-2, Транспортный. Внедрение всемирно согласованной бортовой диагностики. Требования к связи (WWH-OBD). Часть 2. Общий словарь данных (CDD)

1) Для публикации.

ИСО 14229-2:2013(Е)

[18] ISO 27145-4, Транспорт дорожный. Внедрение всемирно согласованной бортовой диагностики.

Требования к связи (V2X-OBD). Часть 4. Соединение между транспортным средством и испытательным оборудованием

[19] IEEE 802.3, Стандарт IEEE для информационных технологий. Телекоммуникации и обмен информацией между системами.

Локальные и городские сети. Обычные требования. Часть 3. Метод доступа с контролем несущей и обнаружением коллизий (CSMA/CD) и спецификации физического уровня²⁾

2) Эквивалентно ИСО/МЭК 8802-3.

ИСО 14229-2:2013(Е)

ИКС 43.180

Ц енаук аана за 48 с траниц .