ГОСТ Р МЭК 870—5—5—96

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

УСТРОЙСТВА И СИСТЕМЫ ТЕЛЕМЕХАНИКИ

Часть 5. ПРОТОКОЛЫ ПЕРЕДАЧИ

Раздел 5. ОСНОВНЫЕ ПРИКЛАДНЫЕ ФУНКЦИИ

Издание официальное

БЗ 11—95/489

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН АО «Научно-исследовательский институт электроэнергетики (ВНИИЭ)»

ВНЕСЕН Министерством топлива и энергетики Российской Федерации и Российским акционерным обществом энергетики и электрификации «ЕЭС РОССИИ»

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 24 апреля 1996 г. № 294

Настоящий стандарт содержит полный аутентичный текст международного стандарта МЭК 870—5—5—95 «Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 5. Основные прикладные функции»

3 ВВЕДЕН ВПЕРВЫЕ

© ИПК Издательство стандартов, 1996

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Госстандарта России

Содержание

Введение 1
1 Область применения и объект 1
2 Нормативные ссылки 2
3 Определения 2
4 Сервис пользователя 3
4.1 Сервисные примитивы пользователя 3
5 Общая концепция прикладных функций 5
6 Основные прикладные функции 6
6.1 Инициализация работы станций 9
6.2 Сбор данных при помощи опроса 21
6.3 Циклическая передача данных 25
6.4 Сбор данных о событиях 26
6.5 Сбор данных о событиях процедурой быстрой проверки (quick-check) 27
6.6 Общий опрос.Опрос КП 27
6.7 Синхронизация по времени 29
6.8 Передача команд 32
6.9 Передача интегральных сумм (телесчет) 36
6.10 Загрузка параметров 37
6.11 Тестовая процедура 40
6.12 Пересылка файлов 41
6.13 Получение (определение) запаздывания передачи 49

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

УСТРОЙСТВА И СИСТЕМЫ ТЕЛЕМЕХАНИКИ

Часть 5. Протоколы передачи
Раздел 5. Основные прикладные функции

Telecontrol equipment and systems.
Part 5. Transmission protocols.
Section 5. Basic application functions

Дата введения 1997—07—01

1 ВВЕДЕНИЕ

Настоящий стандарт устанавливает общие функциональные требования к набору основных прикладных функций в системах телемеханики.

1 ОБЛАСТЬ ПРИМЕНЕНИЯ И ОБЪЕКТ

Настоящий стандарт распространяется на устройства и системы телемеханики с передачей данных последовательными двоичными кодами для контроля и управления территориально-распределенными процессами.

Стандарт определяет основные прикладные функции, которые выполняют стандартные процедуры систем телемеханики. Основные прикладные функции являются пользовательскими процедурами, которые находятся вне уровня 7 (пользовательский уровень) модели МОС (ISO)* для связи открытых систем. Определяемые прикладные функции используют стандартный сервис на уровне пользователя. Определения настоящего стандарта служат базовыми для различных сопутствующих (вспомогательных) стандартов, которые будут детально разработаны для отдельных телемеханических задач. Каждый сопутствующий стандарт может использовать специфический набор оп-

* МОС — Международная организация по стандартизации.
ISO — International Organization for Standardization.

Издание официальное

ределяемых функций. Основные прикладные функции, которых нет в настоящем стандарте, но которые необходимы для формирования сопутствующих стандартов по телемеханике, должны быть определены в этих сопутствующих стандартах. Только определенность сопутствующих стандартов дает возможность совместной работы различной аппаратуры телемеханики.

Общая структура ASDU*, используемых в процедурах, описанных в настоящем стандарте, определена в ГОСТ Р МЭК 870—5—3.

Настоящий стандарт должен применяться совместно с ГОСТ Р МЭК 870—5—1, ГОСТ Р МЭК 870—5—2, ГОСТ Р МЭК 870—5—3 и ГОСТ Р МЭК 870—5—4.

2 НОРМАТИВНЫЕ ССЫЛКИ

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ Р МЭК 870—5—1—95 Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 1. Форматы передаваемых кадров

ГОСТ Р МЭК 870—5—2—95 Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 2. Процедуры в каналах передачи

ГОСТ Р МЭК 870—5—3—95 Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 3. Общая структура данных пользователя

ГОСТ Р МЭК 870—5—4—96 Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 4. Определение и кодирование элементов пользовательской информации

3 ОПРЕДЕЛЕНИЯ

В настоящем стандарте использованы следующие определения:

3.1 **Основная прикладная функция** (в телемеханике) — процедура передачи, выполняющая функции контроля и управления,обычно используемые в системах телемеханики.

Примеры: передача команд, передача о возникновении событий, циклическая передача и т. д.

---

* ASDU — Application Service Data Unit — Пользовательский сервис данных (ГОСТ Р МЭК 870—5—3).

3.2 **Сопутствующий стандарт** (вспомогательный) — сопутствующий стандарт добавляет семантику в основные стандартные определения или в функциональный профиль. Это может быть выражено определением особых целей для объектов информации и определением дополнительных объектов информации, сервисных процедур и параметров основного стандарта.

Примечание — Сопутствующие стандарты не вносят изменения в стандарты, к которым они относятся, но делают более ясными взаимосвязи при их совместном применении для определенной области деятельности.

3.3 **Укрупненная структура (EPA)** — укрупненная модель сравнима с семиуровневой структурой основной модели, однако имеет трехуровневую структуру для получения меньшего времени реакции на важную информацию, но сервис при этом ограничен.

3.4 **Составное поле данных** (CP) — последовательность полей данных с последовательным распределением битов, которые образуют элемент информации.

3.5 **Направление команды** — направление передачи от пункта управления (ПУ) к контролируемому пункту (КП).

3.6 **Направление контроля** — направление передачи от КП к ПУ.

4 СЕРВИС ПОЛЬЗОВАТЕЛЯ

Каждый процесс пользователя может иметь «первичную функцию пользователя» и «вторичную функцию пользователя». «Первичная функция пользователя» — это часть процесса пользователя, которая инициирует запросы пользователя к удаленному объекту пользователя при помощи «вторичной функции пользователя», принадлежащей последнему. Запрашиваемые задачи исполняются при помощи сервиса связи, который включает в себя передачу PDU*. Последовательность процедур сервиса связи описана при помощи последовательности сервисных примитивов.

4.1 Сервисные примитивы пользователя

Первичный пользователь инициирует функцию сервисным примитивом «запрос». Сквитированный сервис пользователя требует ответов от вторичного пользователя. Вторичный пользователь возвращает соответствующие ответы сервисными примитивами ответа, которые доставляются к первичному пользователю сервисным примитивом подтверждения (рисунок 1).

---

* PDU — Protocol data unit — протокол блока данных.

Неподтвержденный сервис пользователя



Подтвержденный сервис пользователя



Рисунок 1 — Основные сервисы пользователя

Сервис. запрос (. req) — первичный пользователь посылает запрос при помощи этого сервисного примитива удаленной вторичной функции пользователя через сервис связи.

Сервис. индикация (. ind) — сервис связи использует этот сервисный примитив для доставки сервисного запроса индикации к вторичной функции пользователя.

Сервис. ответ (. res) — вторичная функция пользователя использует этот сервисный примитив для ответа на запрос от сервиса связи.

Сервис. подтверждение (. con) — сервис связи использует этот сервисный примитив для доставки ответа от вторичной к первичной функции пользователя.

5 ОБЩАЯ КОНЦЕПЦИЯ ПРИКЛАДНЫХ ФУНКЦИЙ

Процессы пользователя, которые применяют связь точка-точка для выполнения согласованных процедур между удаленными пунктами, используют средства, которые имеются на уровнях 7, 2 и 1 модели EPA (рисунок 2).



Рисунок 2 — Расположение сервисов связи и прикладных функций в модели EPA (укрупненная модель)

Одновременно могут происходить более одной процедуры (на разных станциях). Однако процедуры прикладных функций описываются каждая отдельно. Процедуры определяются в едином иерархическом изложении. Дополнительные определения при использовании многоиерархических телемеханических сетей (например, сети с концентраторами) указаны в сопутствующих стандартах.

Отдельные прикладные функции используют сервисные примитивы и элементы процедур передачи на уровнях 7, 2 и 1, как это определено в настоящем стандарте и в ГОСТ Р МЭК 870—5—1, ГОСТ Р МЭК 870—5—2, ГОСТ Р МЭК 870—5—3, ГОСТ Р МЭК 870—5—4.

Прикладные функции являются частью процессов пользователя и выполняют процедуры связи между процессами пользователя.

Нижеследующие пункты настоящего стандарта определяют набор основных прикладных функций. Каждая функция состоит из пере-

дачи определенных сервисных процедур между удаленными процессами пользователя. Содержание информации, форматы кадров различных PDU и списки параметров сервисных примитивов определяются выбранными сопутствующими стандартами.

6 ОСНОВНЫЕ ПРИКЛАДНЫЕ ФУНКЦИИ

Настоящий раздел определяет набор основных прикладных функций, которые используют стандартный сервис связи. Функции описаны с помощью диаграмм, показывающих последовательность блоков данных, которыми обмениваются КП и ПУ, и путем описания задач блоков данных, выполняющих эти функции. Первые две описанные основные прикладные функции, а именно инициализация пунктов и сбор данных при помощи опроса, являются базовыми для других основных прикладных функций. Эти две функции выполняются путем взаимодействия специального сервиса на пользовательском и канальном уровнях и описаны в деталях. Другие основные прикладные функции, которые могут включать использование процедур опроса, описываются без повторения деталей.

Последовательность процедур передачи показана стрелками. Каждая стрелка представляет протокол блока данных PDU. Иерархическая структура символов будет использоваться для обозначения APDU или ASDU; она может быть дополнена различными сопутствующими стандартами. В протоколах ГОСТ Р МЭК 870—5—1, ГОСТ Р МЭК 870—5—2, ГОСТ Р МЭК 870—5—3, ГОСТ Р МЭК 870—5—4 и настоящего стандарта определения APDU* и ASDU одинаковы, т. к. нет явно выделенного APCI*.

В таблице, приведенной ниже, указаны метки ASDU, расположенные в иерархическом порядке, что предполагает возможность использования общих меток в настоящем стандарте и особых меток — в различных сопутствующих стандартах.

К высшему уровню принадлежат:

Вид информации Уровень 1	Метка
Контрольная информация	M
Управляющая (командная) информация	C
Параметр	P
Передача файла	F

* APDU — пользовательский протокол блока данных
APCI — пользовательский протокол управляющей информации
} (см. ГОСТ Р МЭК 870—5—3)

Второй уровень определяет:

Вид информации Уровень 2	Метка
Контрольная информация	M
Одноэлементная информация	M_SP
Двухэлементная информация	M_DP
Измерения	M_ME
События (работа) защиты	M_EP
Интегральные суммы	M_IT
Информация о ступенчатых перемещениях	M_ST
Строки битов и байтов	M_BO
Конец инициализации	M_EI
Доступность пользовательского уровня	M_AA
Управляющая информация	C
Однопозиционная команда	C_SC
Двухпозиционная команда	C_DC
Команда уставки	C_SE
Команда пошагового регулирования	C_RC
Команда опроса	C_IC
Команда синхронизации по времени	C_CS
Определение запаздывания	C_CD
Команда опроса показаний счетчика	C_CI
Тестовая команда	C_TS
Команда установки процесса в исходное состояние	C_RP
Команда считывания	C_RD
Конец инициализации	C_EI
Параметр	P
Измеряемые параметры	P_ME
Активация	P_AC
Передача файла	F
Каталог (структура данных)	F_DR
Выбор или вызов файла или секции	F_SC
Последняя секция или сегмент	F_LS
Подтверждение приема (АСК) файла или секции	F_AF
Готовность файла	F_FR
Готовность секции	F_SR
Сегмент	F_SG

Третий уровень используется разными сопутствующими стандартами и определяет тип ASDU, использование метки времени и т. п. Первая буква в третьем уровне определяет наличие метки времени (N — нет метки времени, T — метка времени), вторая буква определяет тип. Каждый сопутствующий стандарт может устанавливать свои собственные типы в алфавитном порядке, начиная с буквы «A». Например:

Измерения, нормализованные величины без метки времени (тип A)	M_ME_NA
или	
Измерения, масштабированные величины с отметкой времени (тип B),	M_ME_TB
или	
Одиночная команда, без метки времени (тип A)	C_SC_NA

Кроме того, добавляется последняя цифра, показывающая какой сопутствующий стандарт определяет метку ASDU. Например:

Сопутствующий стандарт 101	M_ME_NA_1 или C_SC_NA_1
Сопутствующий стандарт 102	M_ME_NA_2 или C_SC_NA_2

Такая система меток открыта и может при необходимости дополняться на всех иерархических уровнях и в различных сопутствующих стандартах.

ASDU, используемая в направлении КП, может иметь зеркальное отражение в направлении ПУ. Такое зеркальное отражение ASDU используется для положительного/отрицательного подтверждения (квитанции). Необходимо, чтобы была возможность их отличия в обоих направлениях. Поэтому, кроме меток, эти ASDU маркируются следующей аббревиатурой в направлении КП и ПУ:

Направление КП: Активация	ACT
Направление ПУ: Подтверждение активации	ACTCON
Направление КП: Дезактивация	DEACT
Направление ПУ: Подтверждение дезактивации	DEACTCON
Направление ПУ: Прекращение активации	ACTTERM

Кроме того, используются следующие аббревиатуры:

Направление ПУ: Циклическая передача	CYCLIC
Направление ПУ: Спорадическая передача	SPONT

В случае небалансной процедуры передачи команда ACT может быть передана при помощи сервиса канала SEND/NO REPLY (посылка/без ответа) как общее сообщение (например, для опроса станции или синхронизации часов). Затем обратно передается сигнал подтверждения ACTCON, что сообщение ACT получено, индивидуально на каждый КП.

6.1 Инициализация работы станций

Процедура инициализации работы станции требуется для установки станции в правильное рабочее состояние до того, как начнутся телемеханические операции. Необходимо различать холодную и горячую процедуры запуска. Холодный запуск — это процедура первоначальной загрузки станции, чтобы привести базу данных в текущее состояние. Предполагается, что информация о переменных процесса сброшена в исходное состояние до загрузки. Горячий запуск — это процедура перезагрузки станции, которая устанавливается в исходное состояние или повторно активируется. Эта процедура означает, что информация о переменных процесса, полученная до повторной активации, не будет сброшена. Кроме того, различается инициализация КП и ПУ. Определения, приведенные ниже, рассматривают в основном процедуры инициализации, связанные с передачей данных между станциями.

ПУ часто оснащены избыточным оборудованием для управления и хранения базы данных, что гарантирует переключения без потери информации в случае повреждения активного оборудования управления. В этом случае нет необходимости в инициализации общего опроса базы данных ПУ. Однако после перерыва напряжения питания для обновления или общего сброса всего ПУ процедура общего опроса (см. 6.6) и в некоторых системах процедура временной синхронизации (см. 6.7) обязательны.

КП может устанавливаться в исходное состояние по местной команде или по запросу с ПУ.

6.1.1 *Описание основной процедуры инициализации (рисунок 3)*

На рисунке 3 показана инициализация ПУ и КП в общем виде. Более детальное определение, включая используемый сервис связи, приведено на последующих рисунках.

Инициализация контролирующей станции (ПУ).

После внутренней инициализации ПУ уровень канала устанавливает соединение с КП (см. 6.1.2, 6.1.5 и ГОСТ Р МЭК 870—5—2).

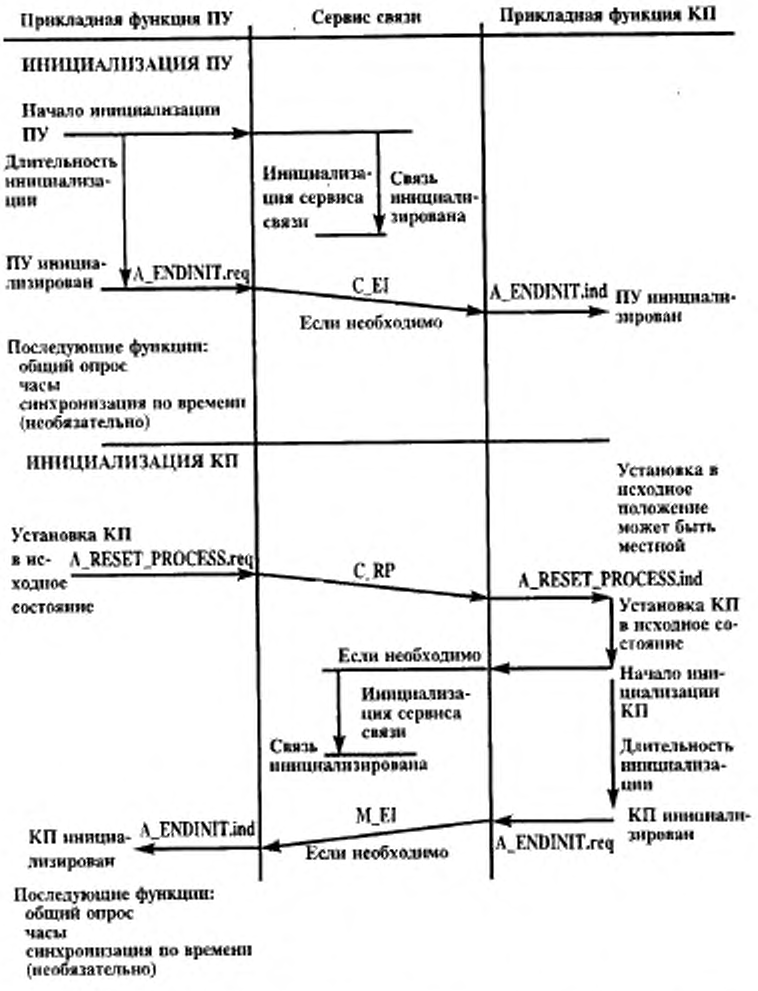


Рисунок 3 — Последовательная процедура — основная процедура инициализации

Когда ПУ готов к передаче информации на КП, он посылает (необязательно) сообщение C_EI (конец инициализации) к подсоединенному КП. После приема PDU C_EI КП может послать информацию о процессе к ПУ. ПУ затем посылает общий запрос (см. 6.6) с синхронизацией по времени (см. 6.7) (необязательно).

Инициализация КП.

При необходимости после внутренней инициализации КП уровень канала устанавливает соединение с ПУ (см. 6.1.3, 6.1.6 и ГОСТ Р МЭК 870—5—2). Если КП готов обрабатывать информацию, поступающую с ПУ, он может послать PDU M_EI к ПУ (необязательно). После получения этого PDU ПУ продолжает посылку общего запроса (см. 6.6), в некоторых системах — с синхронизацией по времени (см. 6.7).

6.1.2 *Инициализация ПУ в небалансных системах передачи (описание последовательности процедур; рисунок 4)*

Если «Начало местной инициализации» появляется сразу после данных, запрошенных с КП (например, как показано пунктиром на рисунке 4), то канал связи ПУ не может получить запрошенные данные, т. к. они уже недоступны. После начала инициализации ПУ уровень канала обычно устанавливается в исходное состояние и становится доступным раньше, чем другие внутренние функции ПУ во время его инициализации. Канал ПУ затем устанавливает соединение с каналом КП посылкой сообщения «Запрос состояния канала» на что получает ответ «Состояние канала». Для установления синхронизации канала ПУ передает команду «Установка удаленного канала в исходное состояние» и получает ответ «АСК». Это «АСК» подтверждает начальные условия уровня канала на ПУ, ожидая следующий бит счета кадров (FCB = 1, см. 5.1.2 ГОСТ Р МЭК 870—5—2). Состояние удаленного уровня канала может быть дополнительно опрошено при помощи команды «Запрос состояния канала». После завершения инициализации прикладных функций на ПУ соединение между прикладными функциями устанавливается передачей PDU C_EI на КП. В системах, в которых установление соединения канала происходит после завершения инициализации прикладных функций на ПУ, передача PDU C_EI не требуется. После инициализации ПУ обновляет информацию общим запросом (см. 6.6) и в некоторых системах синхронизируется по времени (см. 6.7). После этого могут начинаться обычные телемеханические операции.

6.1.3 *Местная инициализация КП в небалансных системах передачи (описание последовательности процедур; рисунок 5)*

После начала местной инициализации КП во время работы с ПУ ПУ определяет, что его канал отсоединен от КП ввиду отсутствия



Рисунок 4 — Последовательная процедура — инициализация ПУ в небалансных системах передачи

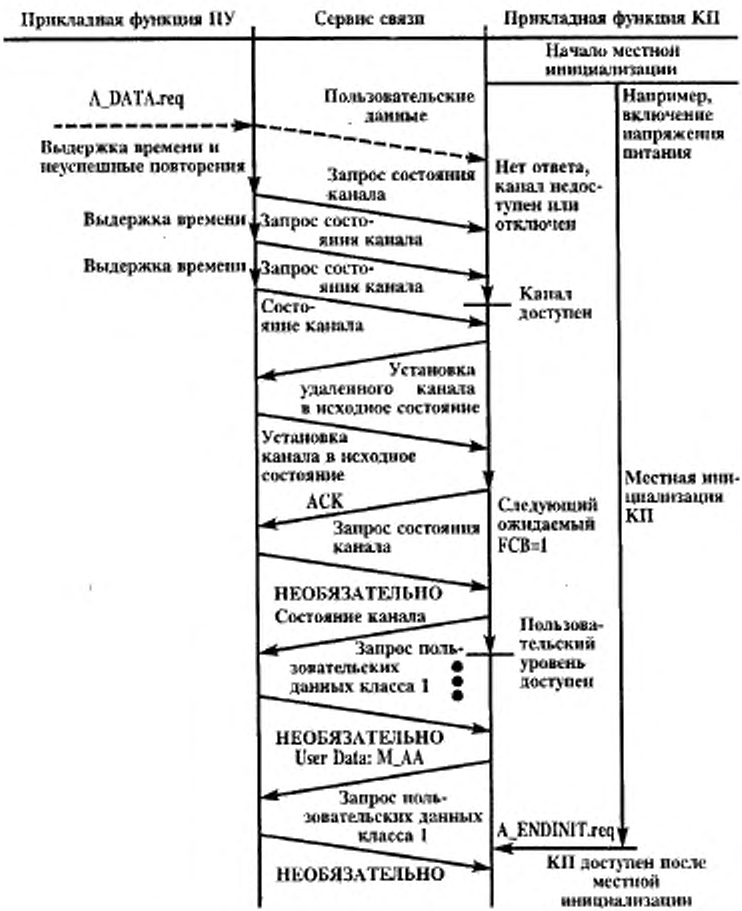


Рисунок 5 — Последовательная процедура — местная инициализация КП в небалансных системах передачи, лист 1

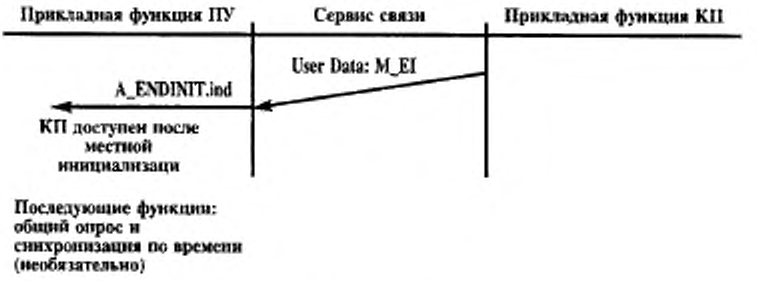


Рисунок 5, лист 2

подтверждения. После определенного числа безуспешных повторений (приложение А ГОСТ Р МЭК 870—5—2) ПУ пытается установить соединение канала посылкой повторных команд «Запрос состояния канала» с определенной выдержкой времени. Если канал на КП доступен, то приходит ответ «Состояние канала». Тогда ПУ передает «Установка удаленного канала в исходное состояние». КП подтверждает условия установки сигналом «АСК» к ПУ (ожидаемый бит счета кадров FCB = 1, см. 5.1.2 ГОСТ Р МЭК 870—5—2). Теперь ПУ может запрашивать КП повторением посылки «Запрос состояния канала». Если ответ будет «Состояние канала», что означает, что данные класса 1 доступны, то данные запрашиваются при помощи посылки «Запрос данных пользователя класса 1» и могут быть подтверждены сообщениями M_AA (прикладной уровень доступен) или M_EI (конец инициализации). Окончание инициализации прикладных функций на КП может быть показано ПУ посредством посылки PDU M_EI. Затем ПУ обновляет свою информацию передачей общего запроса (см. 6.6) и продолжает работать в некоторых системах с синхронизацией времени (см. 6.7). После этого можно начинать обычные телемеханические операции.

Примечание — M_AA применяется, когда ПУ информирован о готовности всей системы связи вдобавок к готовности уровня канала (что показывается сервисом канала «Состояние канала»).

6.1.4 *Дистанционная инициализация КП в небалансных системах передачи (описание последовательности процедур; рисунок 6)*

После получения дистанционной команды RESET_PROCESS C_RP ACT КП может ответить подтверждением RESET_PROCESS

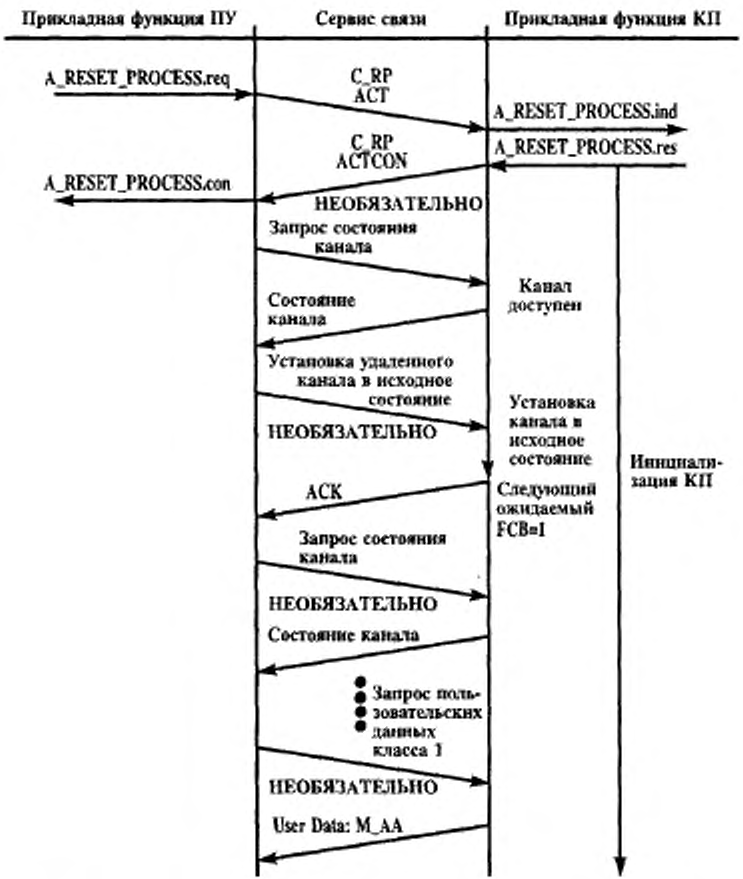


Рисунок 6 — Последовательная процедура — дистанционная инициализация КП в небалансных системах передачи, лист 1



Рисунок 6, лист 2

C_RP ACTCON. После опознания или необязательного подтверждения команды RESET_PROCESS все процессы пользователя выше уровня 7, как показано на рисунке 2, устанавливаются в исходное состояние и инициализируются. Все сообщения, ожидающие передачи, сбрасываются.ПУ опрашивает канал передачей посылки «Запрос состояния канала». Если канал КП доступен, он отвечает «Состояние канала». ПУ может передать сообщение «Установка удаленного канала в исходное состояние» вместе с командой RESET_PROCESS C_RP ACT (необязательно). КП подтверждает условия начала сигналом «ACK» (ожидаемый бит счета кадров FCB = 1, см. 5.1.2 ГОСТ Р МЭК 870—5—2). После этого ПУ может опросить КП повторной посылкой «Запрос состояния канала».

Примечание — Если используется команда «Установка удаленного канала в исходное состояние», то будет выполнена дистанционная инициализация всего КП

Если на «Запрос состояние канала» ответ будет «Состояние канала», это показывает, что данные класса 1 доступны. Данные запрашиваются посылкой «Запрос пользовательских данных класса 1». Прием данных может быть подтвержден или сигналом M_AA (доступен уровень пользователя) или сигналом M_EI (конец инициализации). Оба эти сервиса на КП необязательны, т. к. на нем уровень канала доступен только после конца завершения инициализации.

Примечание — Описываемая процедура дистанционной инициализации повторно запускает функции процесса, когда прикладные функции на КП доступны.

Если они недоступны, то все прикладные пользовательские процессы (прикладной уровень, прикладные функции и прикладные процессы) могут быть повторно запущены при помощи посылки функции обслуживания канала «Установка процессов пользователя в исходное состояние».

6.1.5 *Инициализация ПУ в балансных системах передачи (описание последовательности процедур; рисунок 7)*

После начала инициализации ПУ КП определяет, что канал отсоединен от ПУ ввиду отсутствия подтверждения. КП пытается установить соединение канала, передавая сигнал «Запрос состояния канала» с определенной выдержкой времени. Если уровень канала ПУ доступен, он подтверждает это, посылая к КП сигнал «Состояние канала», КП затем посылает сигнал «Установка удаленного канала в исходное состояние», на что получает ответ «АСК», который подтверждает условия установки на уровне канала ПУ (ожидаемый бит счета кадров FCB = 1, см. 5.1.2 ГОСТ Р МЭК 870—5—2). После этого ПУ синхронизирует соединение канала с КП, передавая посылки «Запрос состояния канала» и «Установка удаленного канала в исходное состояние». После получения «АСК» соединение канала устанавливается в обоих направлениях. Состояние канала может быть опрошено на обоих станциях при помощи посылки «Запрос состояния канала» (на рисунке 7 показан опрос только со стороны ПУ). После окончания инициализации ПУ он может передавать сигнал C_EI (конец инициализации) на КП. Передача PDU C_EI необязательна в системах, которые устанавливают соединение канала после окончания инициализации пользовательских функций на ПУ. После инициализации ПУ обновляет информацию посылкой общего запроса (см. 6.6) и продолжает передачи в некоторых системах посредством синхронизации по времени (см. 6.7). Затем могут начаться обычные телемеханические передачи.

6.1.6 *Местная инициализация КП в балансных системах передачи (описание последовательности процедур; рисунок 8)*

После начала местной инициализации на КП во время работы с ПУ ПУ определяет, что канал отключен от КП ввиду отсутствия подтверждения. Через определенное число безуспешных повторений (см. приложение А ГОСТ Р МЭК 870—5—2) ПУ пытается установить соединение канала посылкой сигнала «Запрос состояния канала» с заданной выдержкой времени. Если канал КП доступен, то приходит ответ «Состояние канала». Тогда ПУ передает сигнал «Установка удаленного канала в исходное состояние». КП подтверждает условия

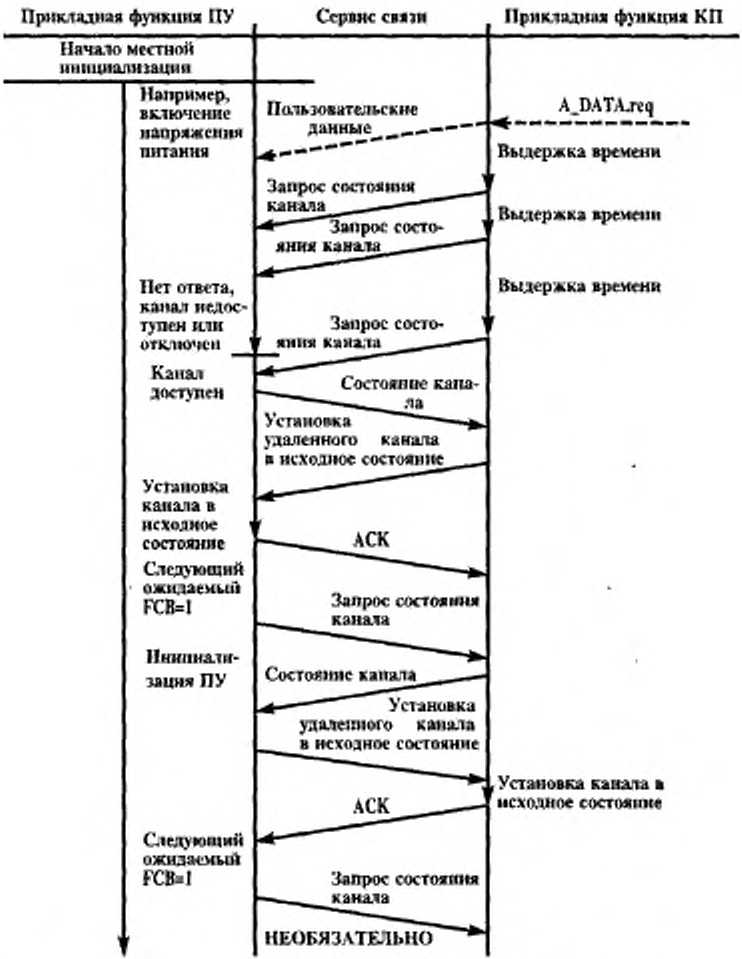


Рисунок 7 — Последовательная процедура — инициализация ПУ в балансных системах передачи, лист 1



Рисунок 7, лист 2

установки при помощи «АСК», передаваемого на ПУ (ожидаемый бит счета кадров FCB = 1, см. 5.1.2 ГОСТ Р МЭК 870—5—2). Затем КП тоже синхронизирует свой канал с ПУ, передавая ей посылки «Запрос состояния канала» и «Установка удаленного канала в исходное состояние». После приема сигнала «АСК» соединение канала устанавливается в обоих направлениях.

КП может показать доступность пользовательского уровня и/или окончание инициализации передачей на ПУ сигналов M_AA (пользовательский уровень доступен) и M_EI (окончание инициализации). Обе эти процедуры необязательны на КП, которые разрешают доступ к каналу после окончания инициализации.

6.1.7 *Дистанционная инициализация КП в балансной системе передачи (описание последовательности процедур; рисунок 9)*

После получения дистанционной команды RESET_PROCESS C_RP ACT КП отвечает подтверждением RESET_PROCESS C_RP ACTCON и начинает инициализацию процессов. ПУ опрашивает канал, передавая посылку «Запрос состояния канала». Если канал на КП доступен, приходит ответ «Состояние канала». Тогда ПУ может передать сигнал «Установка удаленного канала в исходное состояние» в добавление к команде RESET_PROCESS. КП подтверждает начальное состояние сигналом «АСК» на ПУ (ожидаемый бит счета кадров FCB = 1, см. 5.1.2 ГОСТ Р МЭК 870—5—2). ПУ синхронизирует канал с КП передачей посылок «Запрос состояния канала» и «Установка удаленного канала в исходное состояние». Затем КП может (необязательно) передать на ПУ сигналы M_AA (пользователь-

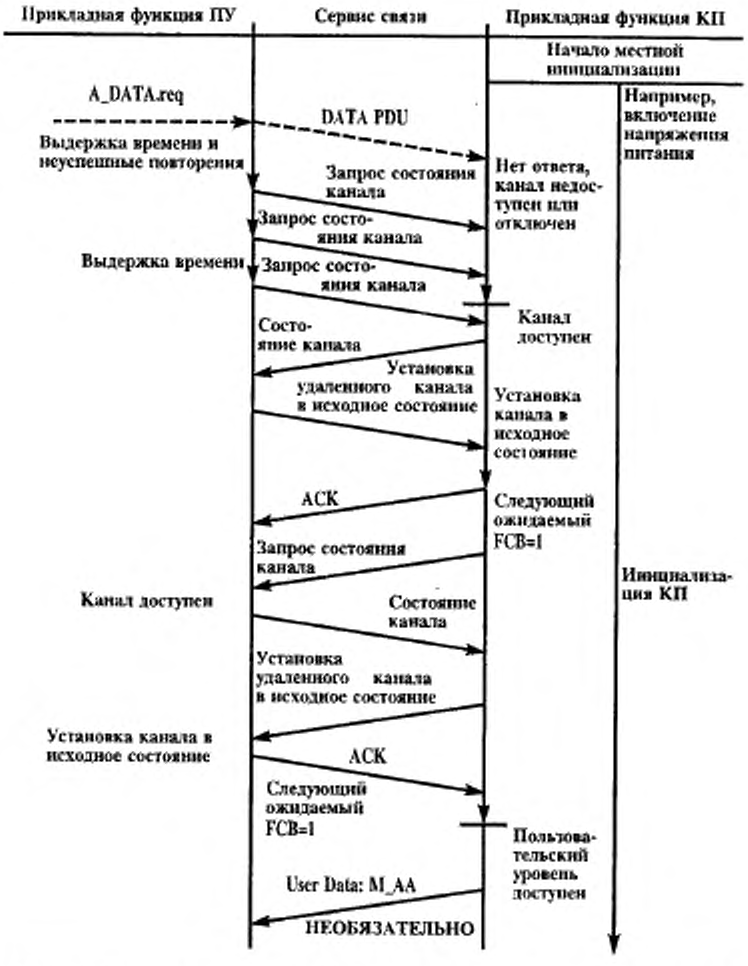


Рисунок 8 — Последовательная процедура — местная инициализация КП в балансных системах передачи, лист 1



Рисунок 8, лист 2

ский уровень доступен) и M_EI (окончание инициализации). Обе эти процедуры на КП необязательны, т. к. после окончания инициализации доступ к каналу восстанавливается.

Примечание — Описываемая процедура дистанционной инициализации повторно запускает функцию процесса, если прикладные функции на КП доступны. Если прикладные функции недоступны, то процесс пользователя может быть повторно запущен при помощи посылки сервисной функции канала «Установка процесса пользователя в исходное состояние».

6.2 Сбор данных при помощи опроса

Сбор данных при помощи опроса используется в системах SCADA, работающих с небалансными процедурами передачи данных, чтобы получить на ПУ действительное состояние переменных процесса на КП. ПУ производит опрос КП. КП может передавать данные только, когда его опрашивают.

Последовательности опроса зависят от системы. Ждущие системы телемеханики используют последовательный опрос только для событий, в то время как чисто циклические системы применяют только последовательный опрос при циклической передаче данных. Системы допускают оба типа опроса. Определенные последовательности опроса должны допускать возможность динамических изменений, вызванных процессом пользователя. Обычный метод – это процедура последовательного опроса на КП циклических данных с низким приоритетом, которая может быть прервана запросом на связь, вызванным таким событием, как передача команд, запросы данных, зависящим от пользователя и т. п. Различные методы могут быть использованы для получения сообщений о событиях, возникающих

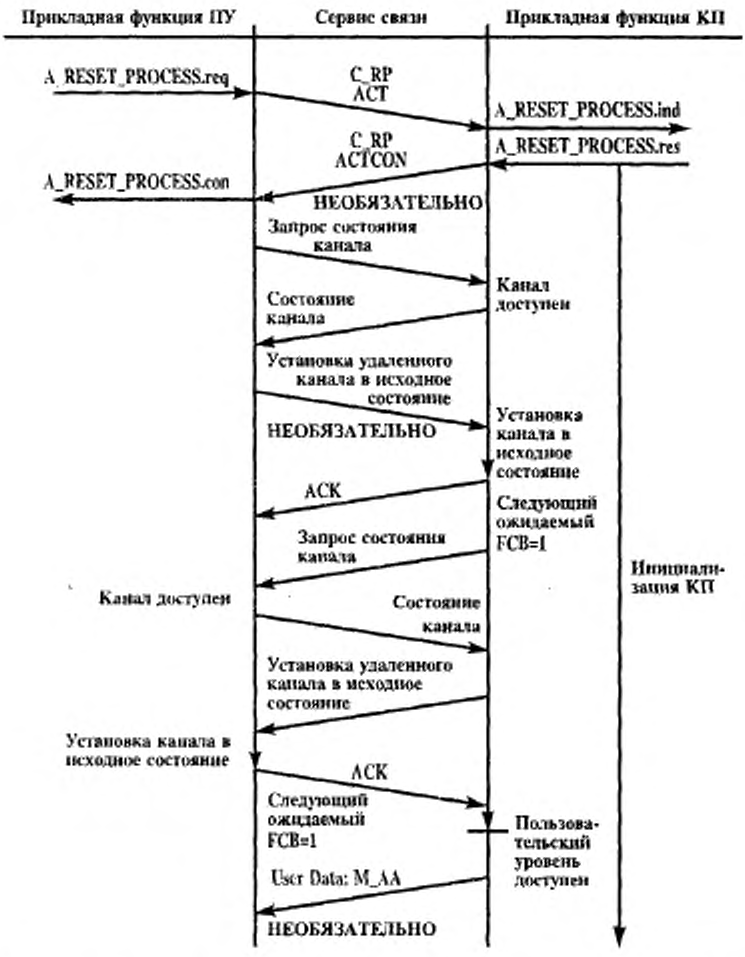


Рисунок 9 — Последовательная процедура — дистанционная инициализация КП в балансных системах передачи, лист 1



Рисунок 9, лист 2

на КП. Некоторые системы используют поочередный или перемежающийся последовательный опрос событий и циклических данных. Другие системы используют циклический последовательный опрос с целью оповещения о наличии событий в обратном цикле ответных данных.

Выбранная процедура опроса должна быть понятна процессу пользователя. Поэтому функция опроса осуществляется сервисом связи. В балансных системах функции опроса нет.

6.2.1 *Описание последовательности процедур (рисунок 10)*

На рисунке 10 показаны различные процедуры ОПРОСА, которые могут возникать при циклических и нециклических последовательностях ОПРОСА.

Первая процедура — это посылка «Запрос пользовательских данных класса 1», которая генерируется сервисом связи на ПУ. На нее приходит ответ «ACK». Это происходит в том случае, если нет событий, ожидающих передачи.

Следующая процедура — это посылка «Запрос пользовательских данных класса 2» к пункту, который возвращает данные. Ответ доставляется прикладной функцией на ПУ посылкой A_USER_DATA_CLASS 2.ind. Бит ACD, равный 1 (см. 5.1.2 ГОСТ Р МЭК 870—5—2), показывает ПУ, что данные класса 1, которые запрошены командой «Запрос пользовательских данных класса 1», доступны КП.

При третьей процедуре прикладная функция на ПУ выдает запрос A_RD_DATA, передаваемый с помощью PDU C_RD (сервис канала посылка/подтверждение) на КП. Затем запрошенные данные

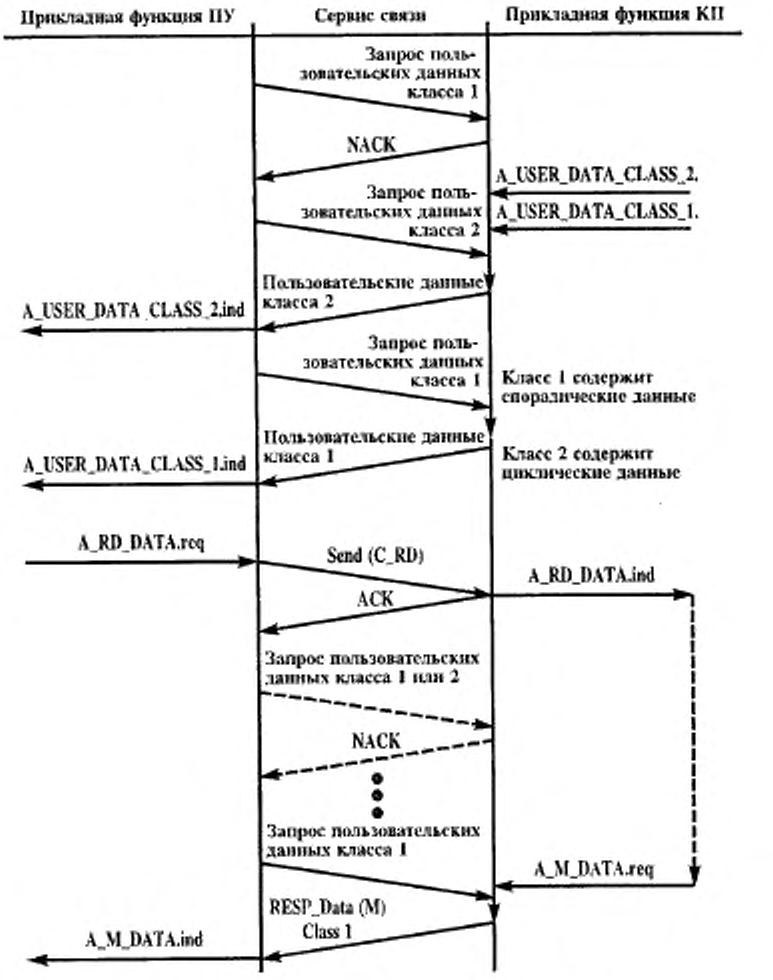


Рисунок 10 — Последовательная процедура — процедура опроса

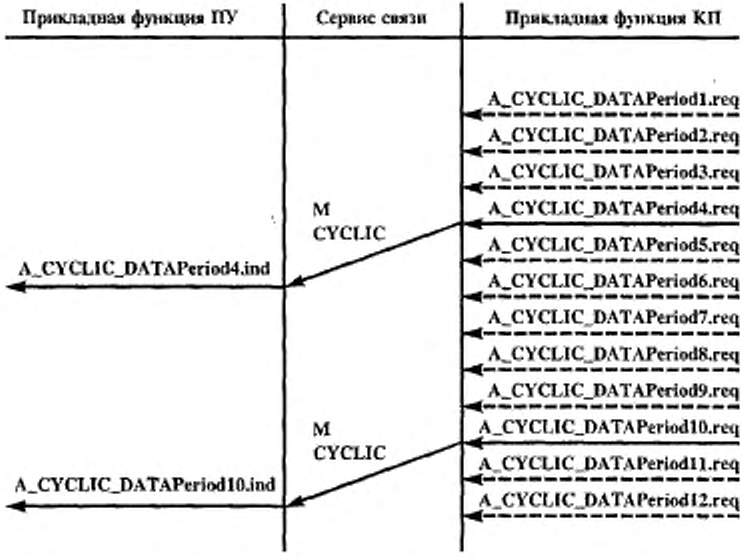
опрашиваются командой «Запрос пользовательских данных класса 1», передаваемой как M_PDU, и принимаются на ПУ как A_M_DATA.ind.

6.3 Циклическая передача данных

Циклическая передача данных используется для выполнения непрерывного опроса текущих значений переменных величин процесса в системах телемеханики, работающих с балансными и небалансными процедурами передач. Эта процедура выполняется с низким приоритетом, что означает, что она может быть прервана запросом на связь при появлении событий.

6.3.1 *Описание последовательности процедур (рисунок 11)*

Пользовательский процесс на КП циклически записывает действительные значения переменных величин процесса в буферную



Примечание — CYCLIC_DATA может быть набором периодически опрашиваемых данных, передаваемых в независимом цикле.

Рисунок 11 — Последовательная процедура — циклическая передача данных

память. Действительные значения из буферной памяти передаются на ПУ с циклическими интервалами, см. рисунок 11. Поступление данных на ПУ указывается сигналом A_CYCLIC_DATA.ind.

6.4 Сбор данных о событиях

На уровне пользователя события возникают спонтанно. Процедуры, выполняющие задачи оповещения удаленных станций о событиях, зависят от работы системы связи. Во всяком случае, местный процесс требует наличия буфера, чтобы сохранить события, которые могут возникать быстрее, чем происходит их пересылка на удаленную станцию.

В балансных системах непосредственная процедура передачи с КП событий с заданным приоритетом прерывает циклическую процедуру передачи с более низким приоритетом.

В небалансных системах процесс передачи на КП должен ждать запроса с ПУ.

6.4.1 *Описание последовательности процедур (рисунок 12)*

Инициализация сбора данных о событиях на ПУ в небалансных системах передачи происходит или периодически, или после запросов с КП, которые оповещают ПУ о появлении событий, когда идет опрос. В балансных системах передача информации о событиях производится процедурами посылка/подтверждение.

Если одно или несколько событий запоминаются на КП, то на ПУ эта информация передается как PDU M_SPONT и принимается пользователем как сигнал A_EVENT.ind (см. рисунок 12).

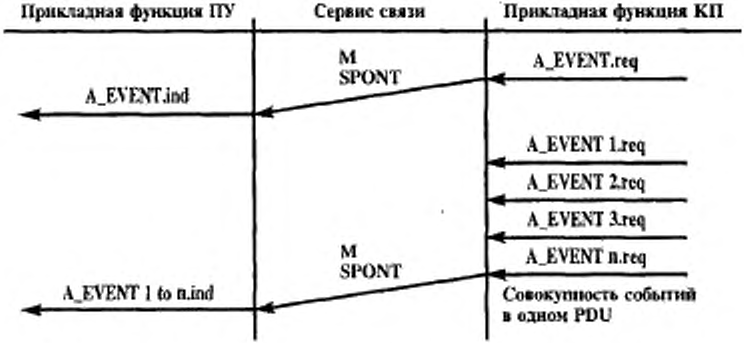


Рисунок 12 — Последовательная процедура – сбор данных о событиях

6.5 Сбор данных о событиях процедурой быстрой проверки (quick-check)

Этот метод иногда используется, чтобы ускорить сбор данных о событиях в небалансных системах.

6.5.1 *Описание последовательности процедур (рисунок 13)*

ПУ посылает через периодические промежутки общие запросы, состоящие из PDU требований доступа, направленные ко всем КП. После передачи таких PDU возможны три результата, см. рисунок 13.

Случай 1. С момента последней передачи событий новые события не появились. В этом случае нет ответа на общий запрос и процедура завершается с выдержкой времени.

Случай 2. Событие появилось на одном из адресуемых КП. Этот КП инициирует сообщение A_EVENT.req через примитив сервиса связи. После приема запроса на требование доступа КП посылает на ПУ ответный PDU — требование доступа. Затем ПУ посылает на КП посылку PDU «Запрос пользовательских данных класса 1». В ответ КП посылает на ПУ информацию о событии в форме PDU «Пользовательские данные класса 1». ПУ передает к прикладной функции примитива A_EVENT.ind.

Случай 3. События появились более чем на одной из адресуемых станций. В этом случае все станции, которые ждут передачи сообщений о событиях, посылают одновременно PDU — ответ на запрос требования доступа к ПУ, и кадры вступают в противоречие. Эта ситуация обнаруживается на ПУ. После передачи всех кадров ПУ начинает процедуру опроса события, как описано в 6.2.1.

6.6 Общий опрос. Опрос КП

Функция опроса КП используется для обновления данных ПУ после процедуры внутренней инициализации станции или, если ПУ обнаруживает потерю информации. Функция общего опроса с ПУ требует передачи с КП действительных значений величин всех переменных процесса.

После приема команды опроса КП передает запрошенную информацию. Количество требуемой информации по прикладным функциям обычно известно на ПУ и КП. Общее количество передаваемой запрошенной информации может быть выбрано ПУ. В этом случае окончание опроса КП определяется после получения всей запрошенной информации. Если требуемое количество запрашиваемой

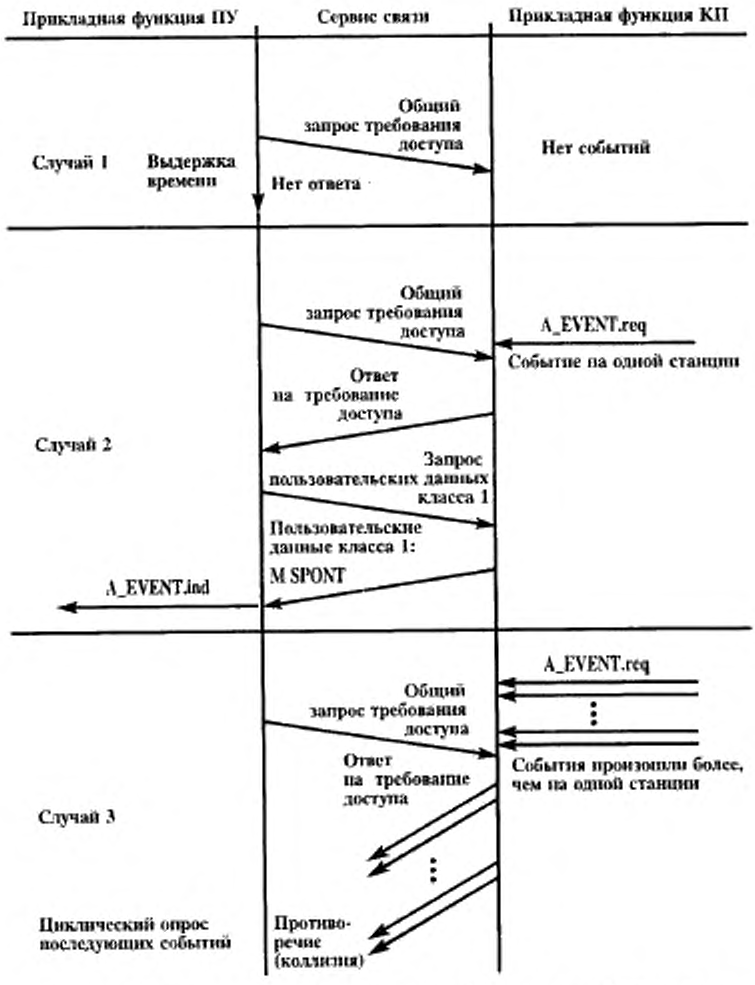


Рисунок 13 — Последовательная процедура — сбор данных о событиях при помощи процедуры быстрого контроля

информации не определяется на ПУ, то КП определяет окончание процедуры запроса после окончания сервиса запроса (необязательно).

Процедура опроса КП может быть прервана событиями, случайно возникающими на КП. Однако необходимо позаботиться, чтобы избежать любой путаницы, которая может произойти с принятой запрошенной информацией, которая из-за появления событий станет устаревшей.

6.6.1 *Описание последовательности процедур (рисунок 14)*

Процесс пользователя на ПУ посылает команду запроса в виде примитива A_GENINCOM.req на сервис связи; последний передает команду PDU C_IC ACT (команда запроса PDU). Эту команду получает процесс пользователя на КП в виде примитива A_GENINCOM.ind.

После начала процедуры запроса процесса пользователя на КП передается подтверждение запроса сигналом PDU C_IC ACTCON, который инициируется примитивом A_GENINACK.req. Это PDU поступает на прикладную функцию на ПУ в виде примитива A_GENINACK.ind.

Использование этого сервиса необязательно.

Прикладная функция на КП передает запрошенную информацию как PDU M (контрольная информация). Передача инициируется примитивом A_INTINF.req. Запрошенная информация поступает к прикладной функции как A_INTINF.ind.

После передачи последней запрошенной информации конец процедуры запроса может быть показан прикладной функции на КП. Окончание передачи запрошенной информации сообщается процессу пользователя на ПУ как примитив A_ENDINT.req при помощи посылок PDU C_IC ACTTERM и A_ENDINT.ind. Применение этого сервиса необязательно.

6.7 Синхронизация по времени

Время на КП должно быть синхронизировано с временем на ПУ, чтобы иметь правильную хронологическую последовательность событий или сообщений, которые передаются на ПУ или регистрируются на месте. В начале время синхронизируется ПУ после инициализации (запуска) системы, а затем синхронизация поддерживается периодически передачей команды PDU C_CS ACT (команда синхронизации по времени).

Команда PDU C_CS ACT содержит полное текущее время, т. е. дату и информацию о времени с требуемым разрешением по времени на момент передачи первого бита PDU C_CS ACT. Информация о времени должна быть скорректирована на КП, когда будет получен PDU, или на ПУ до посылки PDU. Величина корректировки време-

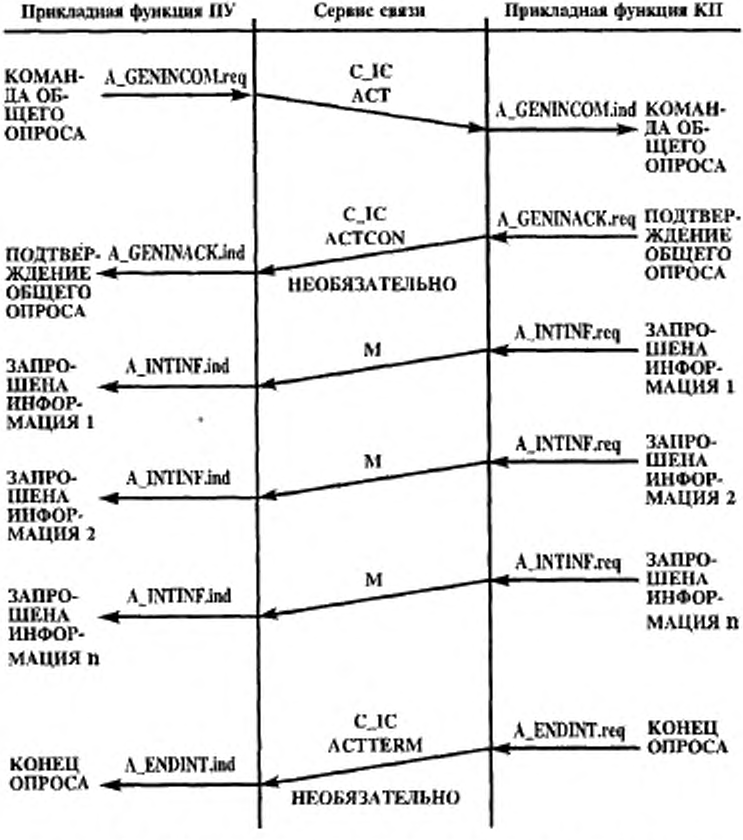


Рисунок 14 — Последовательная процедура — процедура опроса подстанций

ни определяется как сумма задержки (запаздывания) передачи и произведения длины кадра синхронизации на скорость передачи. Выполнение операции синхронизации на КП зависит от специфических требований процесса и не является объектом стандартизации. После выполнения синхронизации ПУ генерирует сигнал PDU C_CI ACTCON, содержащий информацию о местном времени до синхронизации минус величину коррекции времени. Это сообщение передается после любого запомненного PDU с отметкой времени, который ожидает передачу. События, возникшие после синхронизации по времени, передаются после посылки PDU C_CS ACTCON.

КП ожидает получение команды синхронизации в течение заданного заранее промежутка времени. Если команда на синхронизацию не появится за промежуток времени, который зависит от точности часов и допустимых отклонений времени, то в этом случае КП снабжает все объекты информации меткой, показывающей, что точность информации о времени сомнительна. Метка о возможной неточности информации о времени также ставится на объектах информации после повторного включения аппаратуры или инициализации КП до получения правильной команды PDU C_CS ACT (команда синхронизации). События с отметкой времени, возникшие после получения команды PDU C_CS ACT, передаются без метки.

Команда C_CS ACT (команда синхронизации) может быть послана как сервис ПОСЫЛКА/БЕЗ ОТВЕТА (возможно циркулярно к более, чем одному КП) или как сервис ПОСЫЛКА/ПОДТВЕРЖДЕНИЕ на уровне канала.

6.7.1 *Описание последовательности процедур (рисунок 15)*

Пользовательский процесс на ПУ посылает команды синхронизации по времени в виде примитива CLOCKSYN.req к сервису связи, сервис связи передает сигнал PDU C_CS ACT, содержащий значение времени; он выдается пользовательскому процессу на КП как примитив A_CLOCKSYN.ind.

После окончания операции синхронизации пользовательский процесс на ПУ вырабатывает сообщение о времени, которое передается как сигнал PDU C_CS ACTCON, инициируемый примитивом A_TIMEMESS.req. Этот PDU содержит достоверную информацию о времени в момент перед синхронизацией, за вычетом значения величины коррекции по времени, которая сообщена прикладной функции на ПУ как примитив A_TIMEMESS.ind.

Примечание — Динамическая процедура измерения значения величины запаздывания (задержки) передачи описана в 6.1.3.

6.8 Передача команд

Команды используются в системах телемеханики для изменения состояния оперативного оборудования. Таким образом, команды используются для управления контролируемым процессом в нужном направлении.

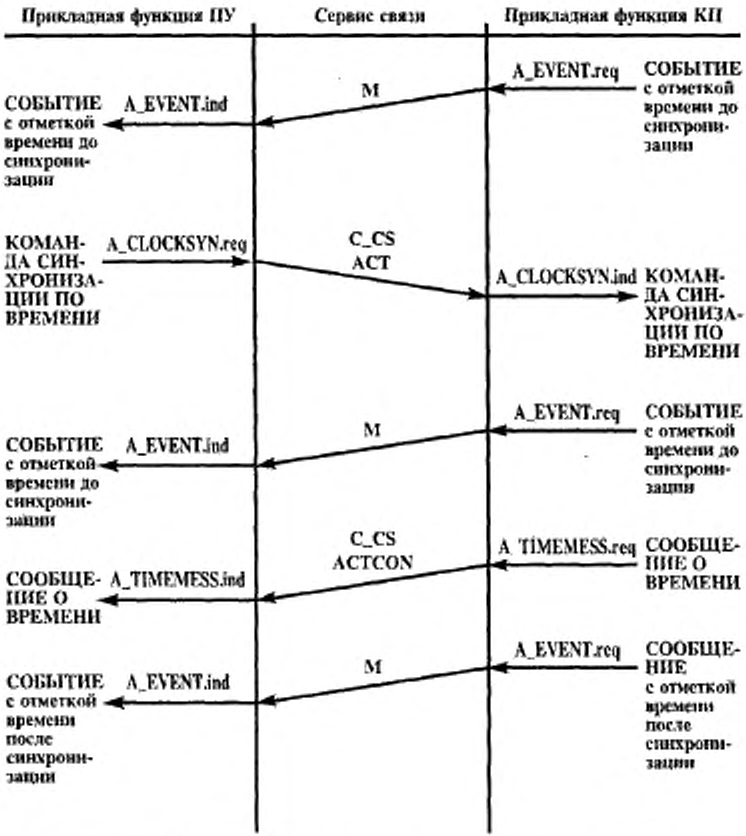


Рисунок 15 — Последовательная процедура — процедура синхронизации по времени, включенная в процедуру передачи событий с отметкой времени

Команды могут инициироваться оператором или автоматическими управляющими процедурами на ПУ. Меры предосторожности против неразрешенного доступа или нежелательных действий зависят от системы или процесса. Типовые части оперативного оборудования или задачи процесса включают в себя:

— электрические контакторы, разъединители;

— выключатели;

— пуск-останов местного процесса управления с пункта управления процессом;

— выполнение шага в местной управляющей последовательности;

— установление точки, пределы для сигналов аварий, особые параметры и т. п.

Для передачи команд существуют две стандартные процедуры, а именно:

1) прямая (непосредственная) команда;

2) команды выбора и исполнения.

Непосредственная команда используется на КП для прямых операций управления процессами на удаленном КП. Прикладная функция на КП проверяется по условиям безопасности — допустимость и обоснованность принятого командного сообщения — и, если проверка положительна, операция выполняется.

Команды выбора и исполнения используются на ПУ для подготовки заданной управляющей операции на удаленном КП, проверки, что управляющая операция подготовлена правильно и затем для выполнения команды.

Проверка выполняется оператором или пользовательской процедурой. КП не начинает операцию управления, пока не получит правильный сигнал исполнения.

6.8.1 *Описание последовательности процедур (рисунок 16)*

Последовательность процедур команд выбора и исполнения и прямой (непосредственный) команды показана на рисунке 16 и описана ниже.

В случае процедур выбора и исполнения пользовательский процесс ПУ посылает примитив A_SELECT.req к сервису связи; сервис связи передает команду PDU, содержащую С ACT (команда выбора), которая сообщается процессу пользователя на КП в виде примитива A_SELECT.ind. Если пользовательский процесс на КП готов к приему команды, что сообщается «Select command», то образуется

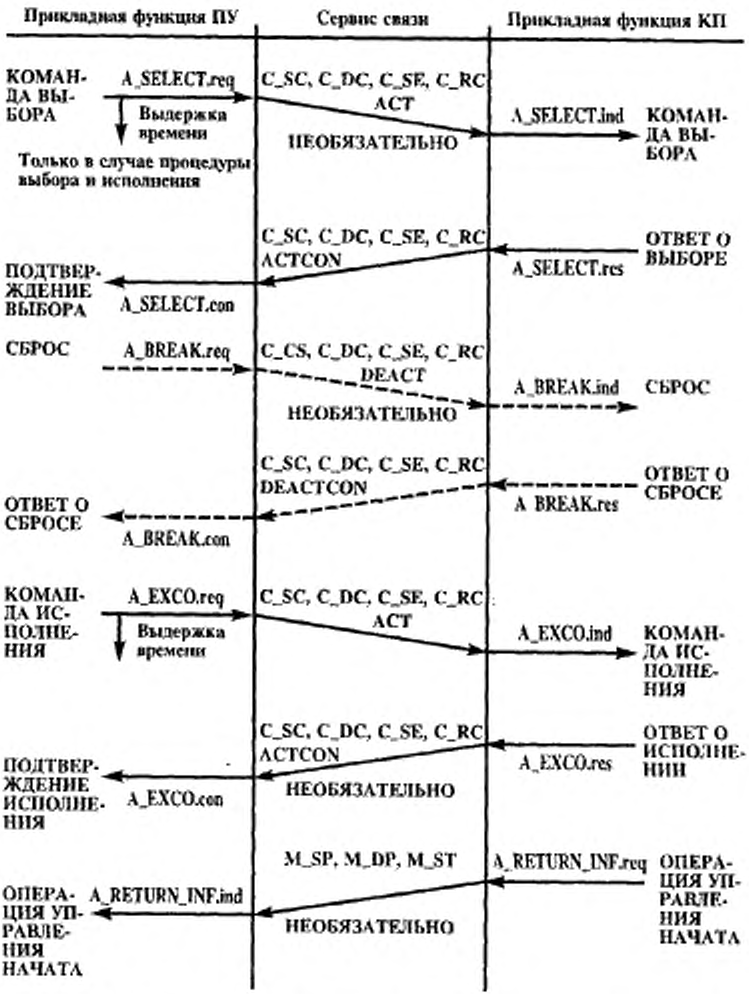


Рисунок 16 — Последовательная процедура — процедура передачи команд, лист 1



Рисунок 16, лист 2

«Select response», который возвращается к сервису связи как примитив A_SELECT.res. Эта команда ответа передается как сигнал PDU C_ACTCON и вырабатывает подтверждение «Select confirmation» в виде примитива A_SELECT.con. Такая процедура используется только в случае команд выбора и исполнения, которые не прерываются и контролируются посредством задания выдержки времени.

Процедура выбора прекращается при помощи команды «Break off command», передаваемой на КП сигналом C DEACT и завершаемой ответом C DEACTCON.

При необходимости команда «Execute command» выдается к сервису связи примитивом A_EXCO.req. Эта команда передается как PDU C ACT и поступает к пользовательскому процессу на КП как примитив A_EXCO.ind. Ответ «Execution response» может быть возвращен на ПУ как PDU C ACTCON; там вырабатывается положительное или отрицательное подтверждение о том, что готовится определенное управляющее действие. Эта процедура не прерываема и контролируется посредством задания выдержки времени.

При процедуре непосредственной (прямой) команды пользовательский процесс на КП проверяет, не заблокирован ли выход адресуемой команды, т. е. готов ли выход к выполнению команды. Если проверка показала, что все в порядке, пользовательский процесс на

КП выдает команду на исполнительное устройство, а затем может возвратить положительное подтверждение PDU C ACTCON. В противном случае ответом будет отрицательный сигнал PDU C ACTCON.

Когда команда поступит к процессу, адресуемое оперативное оборудование изменит свое состояние. Исполнение этого изменения контролируется и показывается на ПУ при помощи возврата информации. В случае специальной команды, такой как двухпозиционная команда, управляющая медленно действующими разъединителями, начало изменения положения (состояния), когда ранее бывшее состояние «включено» или «выключено» уже нарушено, это может быть (необязательно) показано на ПУ соответствующими PDU M (Return information: «Control operation commenced»). Когда исполнение команды завершено достижением нового определенного положения, процесс пользователя на КП покажет это соответствующей посылкой на ПУ PDU M (Return information: «Control operation complete»), см. рисунок 16.

Это может заканчиваться посылкой PDU C ACTTERM, которая показывает, что операция управления окончена (необязательно).

6.9 Передача интегральных сумм (телесчет)

Телесчет определяется как «передача значений измеряемых величин, проинтегрированных по заданному параметру», например, по времени, и использующих технику связи. Интегрирование может осуществляться до или после передачи. Если интегрирование осуществляется до передачи, то используется термин «передача интегральных значений».

Интегральное значение — это величина, проинтегрированная за определенный период времени. Текущее время и периодический временной интервал последовательного получения интегральных значений являются параметрами системы. Некоторые системы используют периодические команды для начала опроса интегральных значений на ПУ. В других системах периодическая активация вызывается местным источником времени (часами) на КП. Синхронизация по времени поддерживается или системой телемеханики (см. 6.7) или внешними синхронизирующими процедурами, например, приемом национального (государственного) или международного радиосигнала точного времени.

Для получения информации от счетчиков применяются два различных метода:

1) получение интегральных значений

КП периодически в определенное время запоминает интегральное значение в буферной памяти и передает запомненные значения на ПУ. Счетчики продолжают работать, не сбрасываясь в исходное состояние. В этом случае значение приращения за период рассчитывается на ПУ. Приращение — это разность между двумя последовательно переданными значениями;

2) получение информации с приращениями

КП периодически в определенное время запоминает интегральное значение в буферной памяти и сбрасывает интегральные значения в ноль, затем запомненные значения передаются на ПУ.

6.9.1 *Описание последовательности процедур (рисунок 17)*

ПУ передает (необязательно) периодически в определенное время команды PDU C_CI ACT (или Memorise counter command или Memorise increment command) на КП. Обе команды вызывают передачу интегральных значений в буферную память. В случае команды Memorise increment command интегральное значение переставляется дополнительно в ноль. В другом случае активация этой процедуры вызывается местным источником времени на КП.

После исполнения этих процедур запомненные значения могут быть запрошены или необязательной посылкой C_CI ACT (Reguest integral totals), получая в ответ сигнал C_CI ACTCON, или запомненные значения могут передаваться на ПУ как события на объекте. В этом случае запомненные значения (PDU M_IT) могут быт получены ПУ как события (см. 6.4).

Передача интегральных значений может быть остановлена сигналом примитива A_IBREAK.req, который передается на КП при помощи C_CI DEACT с ответом C_CI DEACTCON.

В случае ЗАПРОСА интегральных значений (integral totals) за ним следует заключительная посылка PDU C_CI ACTTERM, которая показывает, что операция управления окончена (также необязательно).

6.10 Загрузка параметров

Загрузка параметров используется в системах при измерении на КП заранее определенных (установленных) параметров, например значений порогов или пределов измеряемых величин. Обычно загрузка параметра выполняется в два процедурных шага:

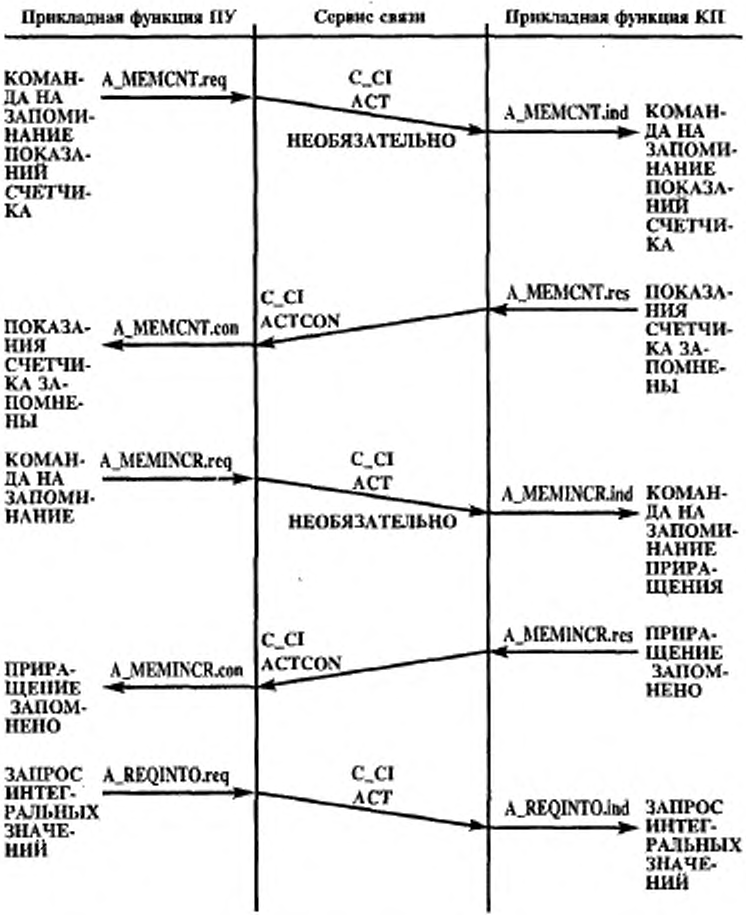


Рисунок 17 — Последовательная процедура — сбор данных об интегральных значениях, лист 1

1) один или более чем один параметр загружается в КП командой параметра. На КП эти параметры запоминаются и еще не активны;

2) на втором шаге предварительно загруженные параметры активизируются при помощи команды работы параметра.

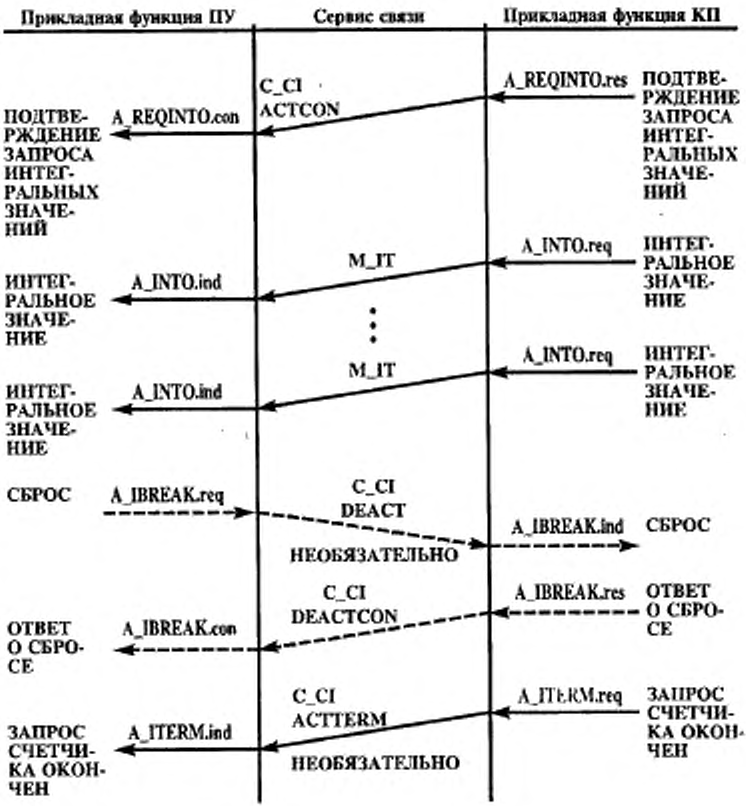


Рисунок 17, лист 2

Эти два шага (этапа) обязательны, если определенное число параметров необходимо активировать точно в одно время. При загрузке одного параметра активацию можно объединить с загрузкой, т. е. эти процедуры можно выполнить в один шаг (этап).

6.10.1 *Описание последовательности процедур (рисунок 18)*

Прикладная функция на ПУ посылает сигнал примитива A_PARAM.req на сервис связи; последний передает сигнал PDU, содержащий команду P_ME ACT (Parameter command), которая поступает к прикладной функции на КП как примитив A_PARAM.ind. Прикладная функция на КП вырабатывает команду Parameter command ACK, которая возвращается к сервису связи как примитив A_PARAM.res. Эта команда запроса передается как PDU P_ME ACTCON и вырабатывает подтверждение примитив A_PARAM.con на ПУ.

Если ранее загруженные параметры активируются по отдельности, то передается команда активации параметра Parameter activation command посредством примитива A_PACTIV.req на сервис связи. Последняя передается как PDU P_AC ACT и поступает на КП к функции пользовательского процесса как примитив A_PACTIV.ind. Сигнал подтверждения может возвратиться на КП как P_AC ACTCON, чтобы подтвердить, что ранее загруженные параметры уже работают.

В случае местного изменения параметра КП может передавать на ПУ сопутствующий PDU P_ME SPONT.

6.11 Тестовая процедура

Тестовая процедура используется для проверки полной петли от ПУ до КП и затем обратно к ПУ, включая соответствующие функции пользователя.

6.11.1 *Описание последовательности процедур (рисунок 19)*

Прикладная функция ПУ посылает сигнал примитива A_TEST.req к сервису связи, который передает команду PDU C_TS ACT (Test command), поступающую к прикладной функции КП как примитив A_TEST.ind. Прикладная функция на КП вырабатывает команду Test command ACK, которая возвращается на сервис связи как примитив A_TEST.req. Этот ответ передается как PDU C_TS ACTCON, вырабатывая на ПУ подтверждение в виде примитива A_TEST.con.

ПУ проверяет отраженный сигнал PDU C_TS. В случае получения такого же PDU за ограниченное время тест считается положительным.

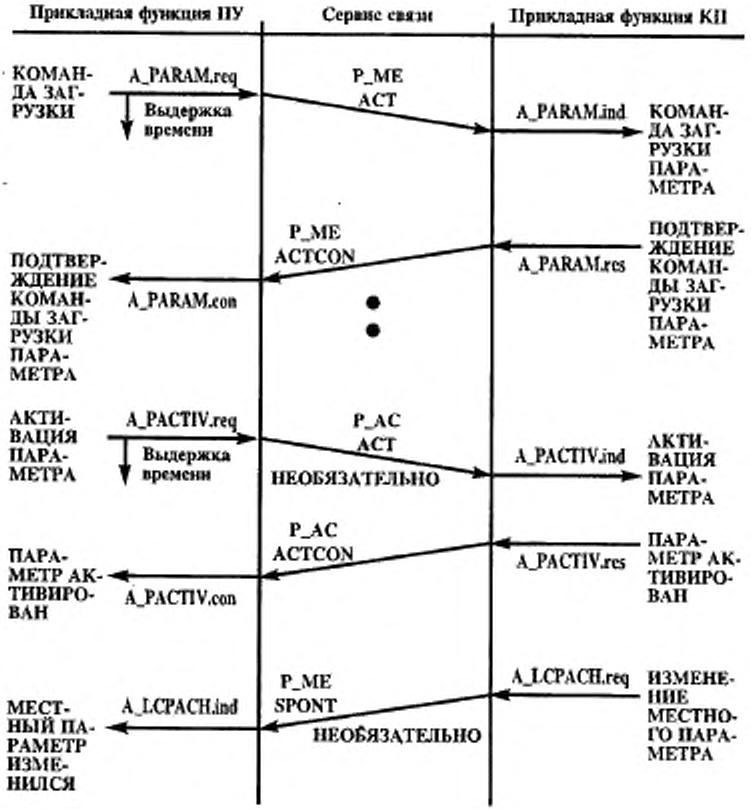


Рисунок 18 — Последовательная процедура — загрузка параметра

6.12 Пересылка файлов

Если размер индивидуального объекта информации превышает заданную максимальную длину ASDU в данной системе телемеханики, то требуется функция пересылки файлов. В этом случае объект информации передается к месту назначения в форме сегментов.

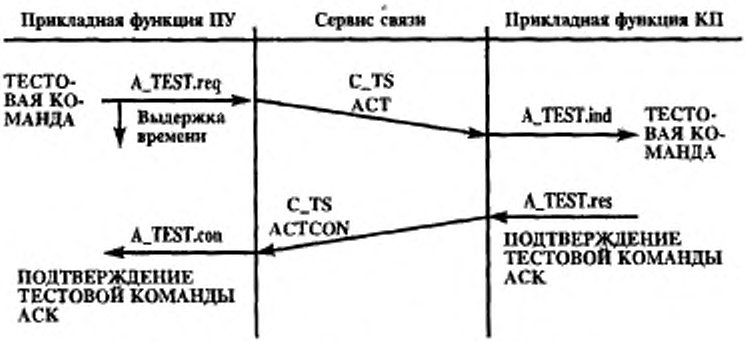


Рисунок 19 — Последовательная процедура — тестовая процедура

В системах телемеханики файлы пересылаются от КП к ПУ и наоборот. События, которые вызывают обширную регистрацию данных на КП (например, регистрация данных об ошибках), передаются на КП последовательно. Типы и количество таких файлов, регистрируемых на КП, должны быть сообщены на ПУ при помощи PDU Directory.

Процедуры загрузки для списков параметров или программ от ПУ к КП управляются с ПУ и поэтому не требуют передачи директив.

Структуры файлов одинаковы в обоих направлениях (рисунок 20). Файлы разделяются на секции, секции разделяются на сегменты, которые передаются в последовательных PDU. Индивидуальное ошибочное событие, фиксируемое регистром ошибок, может быть представлено как секция файла, а несколько величин могут быть собраны в отдельные сегменты файла.

Несмотря на одинаковые структуры файлов в обоих направлениях передачи, процедуры передачи различны и поэтому будут описаны отдельно.

6.12.1 Пересылка файла в направлении ПУ

Пересылка файлов от КП в основном используется для сообщения ПУ о возникших событиях, которые повлекли за собой регистрацию большого количества данных. Количество файлов, время регистрации и типы событий (например, мгновенные значения в момент срабатывания устройств защиты) должны быть переданы ПУ

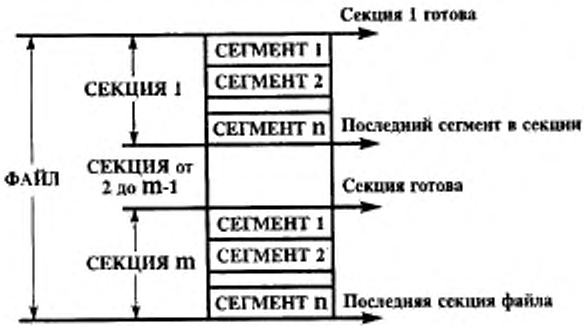


Рисунок 20 — Общее построение файла

при помощи сообщения PDU Directory. ПУ решает, должен ли быть переслан файл и какой именно. Запись успешно переданных файлов стирается на КП, чтобы освободить буфер для создания новых файлов.

6.12.1.1 Описание последовательности процедур (рисунок 21)

После создания нового файла на КП (при возникновении событий) на ПУ может быть самопроизвольно послано сообщение PDU Directory. PDU Directory содержит количество, тип и значения файлов, зарегистрированных, но еще не пересланных на ПУ.

Кроме того, ПУ может в любое время получать при помощи запроса A_CALL_DIRECTORY количество и типы файлов, зарегистрированных на КП.

Если ПУ готов получить файл, он посылает на КП команду PDU SELECT_FILE. КП предлагает передать выбранный файл и показывает ПУ при помощи PDU FILE_READY это состояние.

ПУ запрашивает выбранный файл при помощи посылки PDU CALL_FILE. КП в свою очередь показывает при помощи посылки PDU SECTION_READY, что первая секция файла готова к передаче.

ПУ запрашивает первую секцию при помощи посылки PDU CALL_SECTION (положительного) или отвергает ее при помощи посылки PDU CALL_SECTION (отрицательного). В случае отрицательного ответа КП предлагает для передачи вторую (следующую) секцию при помощи посылки PDU SECTION_READY. В положительном случае КП передает последовательно сегменты от 1 до n при

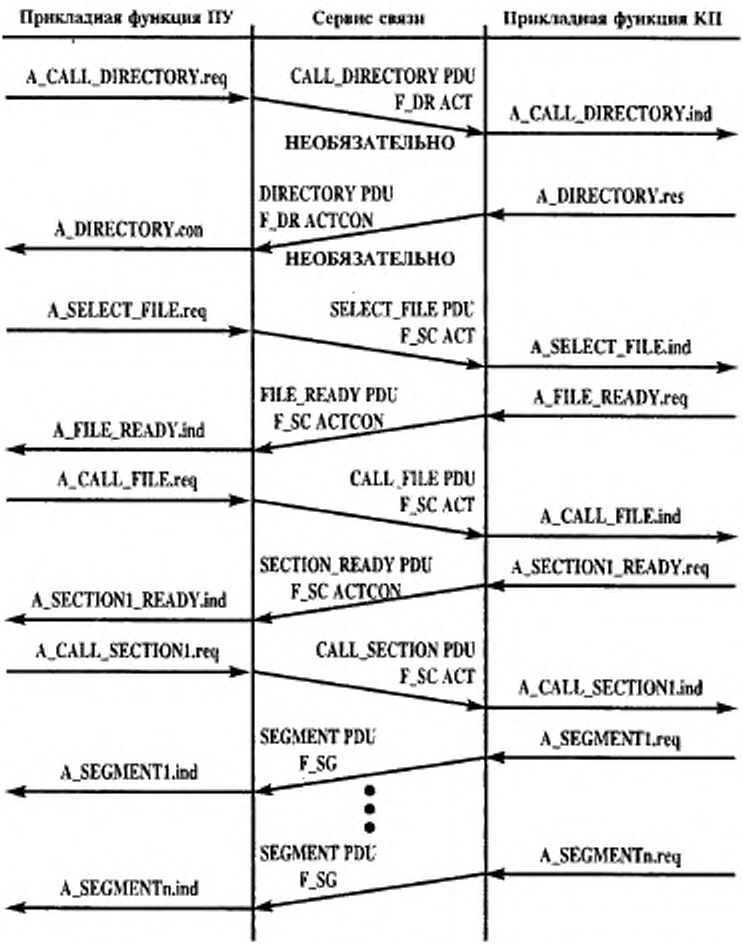


Рисунок 21 — Последовательная процедура — передача файла (в направлении ПУ), лист 1

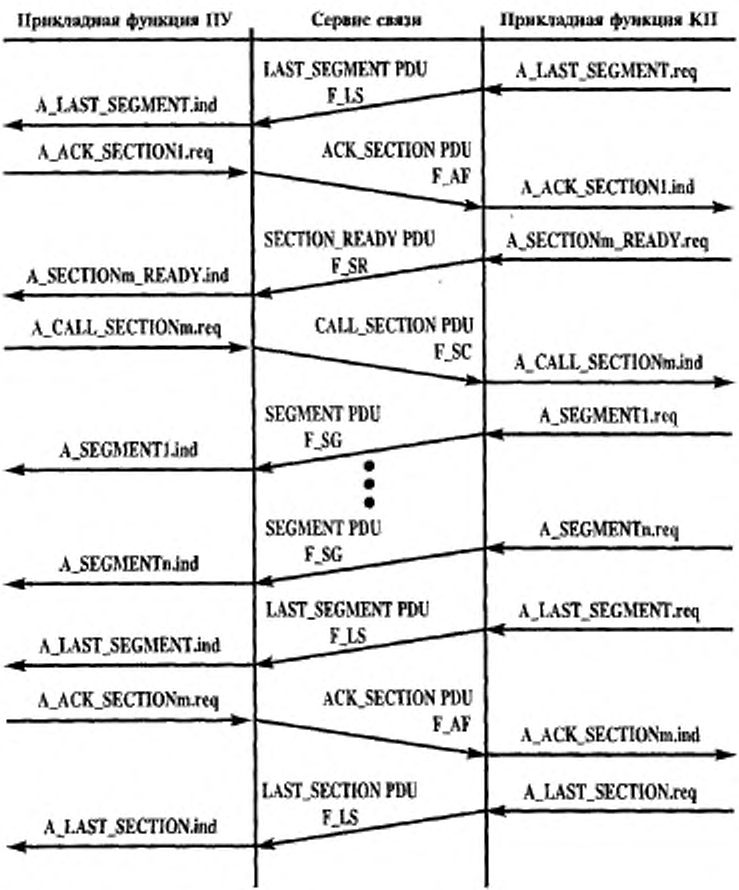


Рисунок 21, лист 2



Рисунок 21, лист 3

помощи сигнала PDU SEGMENT. После передачи последнего сегмента КП посылает на ПУ PDU LAST_SEGMENT. ПУ подтверждает получение соответствующей секции при помощи сигнала PDU ACK_SECTION положительного или отрицательного. В случае отрицательного подтверждения (квитанция) эта же секция предлагается снова при помощи посылки PDU SECTION_READY; при положительном подтверждении (квитанции) предлагается следующая секция при помощи посылки PDU SECTION_READY. Эта процедура повторяется при передаче последующих секций файла.

После передачи последней секции сигнал PDU LAST_SECTION показывает окончание передачи файла. ПУ подтверждает при помощи посылки PDU ACK_FILE правильное получение всего файла. КП может теперь стереть файл на буферной памяти и из директивы. Затем на ПУ может быть послано действительное состояние директивы при помощи сообщения PDU Directory.

6.12.2 *Пересылка файлов в направлении КП*

Пересылка файлов в направлении КП используется в основном для загрузки списков параметров и программ. ПУ ответственен за расположение типов, количества и размеров пересылаемых файлов данных. Поэтому пересылка директив не требуется.

6.12.2.1 Описание последовательности процедур (рисунок 22)

ПУ при помощи сообщения PDU FILE_READY извещает о намерении переслать файл на КП. Как только КП готов принять файл, он передает на ПУ сигнал PDU CALL_FILE. ПУ тогда извещает при помощи посылки PDU SECTION_READY, что секция подготовле-

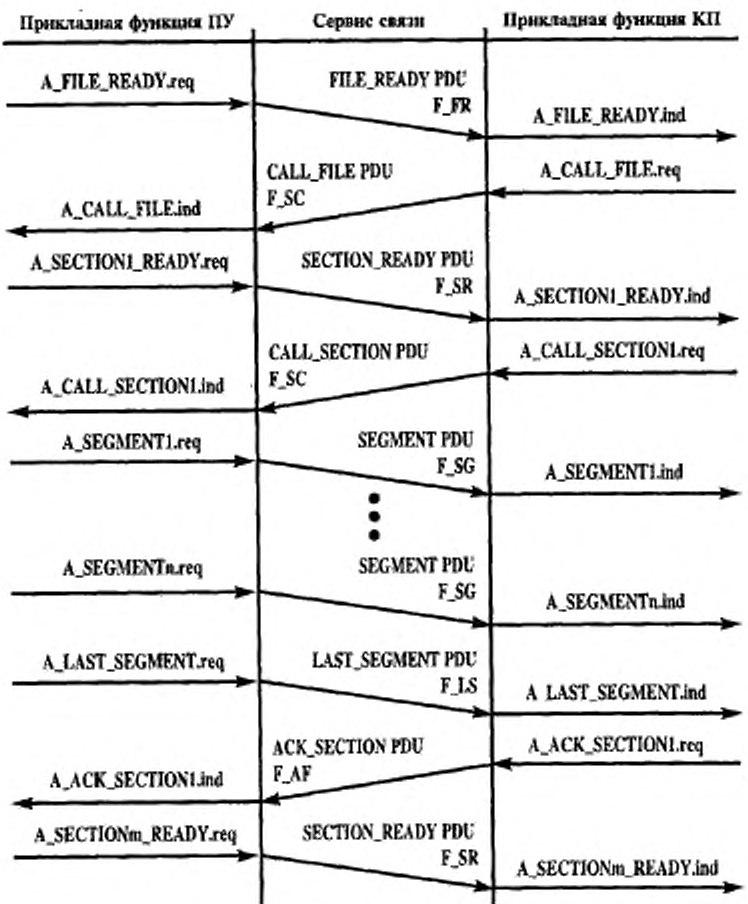


Рисунок 22 — Последовательная процедура — передача файла (в направлении КП), лист 1

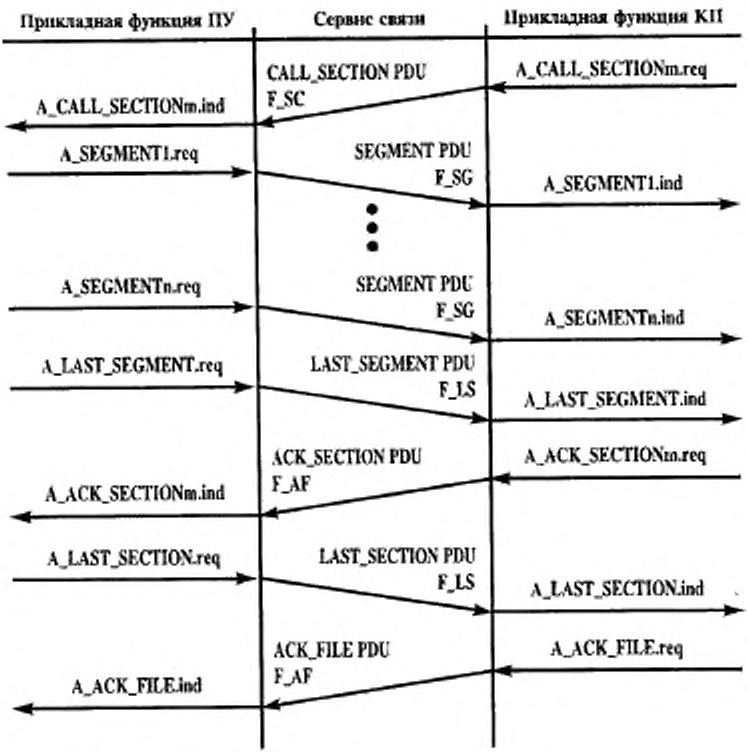


Рисунок 22, лист 2

на. Если КП готов получить ее, то он передает сигнал PDU CALL_SECTION.

ПУ при помощи посылки PDU SEGMENTS передает затем сегменты секции. Последний сегмент обозначается как PDU LAST_SEGMENT. После правильного приема сегментов КП передает сигнал PDU ACK_SECTION.

ПУ последовательно передает следующие секции, как описано выше. После передачи последней секции ПУ передает сигнал PDU

LAST_SECTION, что обозначает конец файла. Если КП получает весь файл правильно, это подтверждается посредством посылки PDU ACK_FILE.

6.13 Получение (определение) запаздывания передачи

Синхронизация часов на КП, включающая корректировку времени, описана в 6.7. Значение коррекции времени определяется суммой задержки передачи и внутренней задержки в аппаратуре. Последняя зависит от особых требований процесса и не является объектом стандартизации. Задержка передачи — это величина, которая может быть независимо определена параметрически или динамически по инициативе ПУ. Нижеследующая процедура показывает динамическое определение задержки передачи.

6.13.1 *Описание последовательности процедур (рисунок 23)*

PDU C_CD ACT (PDU посылки определения запаздывания) содержит значение времени в момент, когда передается первый бит PDU C_CD ACT (время SDT). КП синхронизирует свои внутренние часы [или дополнительные (вспомогательные) часы] после получения этого значения времени. Таким образом, время на КП синхронизировано с временем SDT. КП возвращает обратно сообщение PDU C_CD ACTCON, которое содержит значение времени часов КП (SDT + tR) в момент, когда передается первый бит кадров PDU C_CD ACTCON. ПУ получает ответное PDU в момент RDT. Таким образом, задержка передачи tD может быть вычислена на ПУ по формуле

$$tD = \frac{RDT - (SDT + tR)}{2}.$$

Значение определенной задержки передачи передается на КП (необязательно) как C_CD SPONTANEOUS и может быть использовано для корректировки синхронизации по времени, как описано в 6.7.

Примечание — Описанный метод предусматривает одинаковые скорости передачи в направлении команды и контроля.

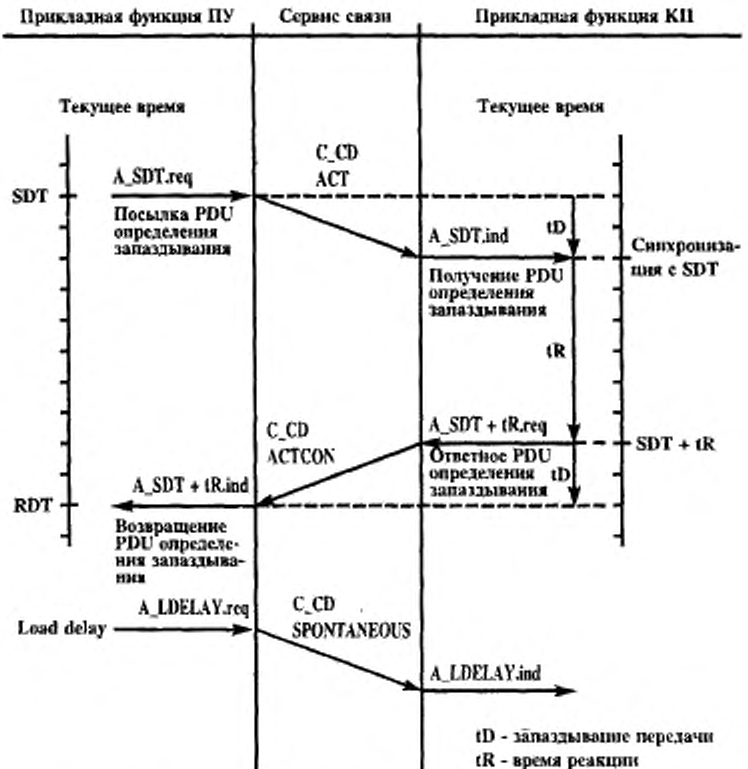


Рисунок 23 — Последовательная процедура — определение запаздывания передачи

УДК 621.398:006.354 ОКС 33.200 П77 ОКП 42 3200

Ключевые слова: устройства телемеханики, системы телемеханики, требования функциональные, набор функций, функции основные прикладные, передача данных, коды последовательные, коды двоичные, контроль процессов, управление процессами

Редактор *Т. С. Шеко*
Технический редактор *Л. А. Кузнецова*
Корректор *Т. А. Васильева*
Компьютерная верстка *А. Г. Хоменко*

Изд. лиц. № 021007 от 10.08.95. Сдано в набор 27.05.96. Подписано в печать 16.08.96. Усл.печ.л. 3,26. Уч.-изд.л. 3,38. Тираж 262 экз. С 3727. Зак. 946.

gost.e-ecolog.ru

ьство стандартов, 107076, Москва, Колодезный пер., 14.
Набрано в Калужской типографии стандартов на ПЭВМ.
Калужская типография стандартов, ул. Московская, 256.
ПЛР № 040138