ГОСТ Р 34.1952—92
(ИСО 8473—88)

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ. ПРОТОКОЛ ДЛЯ ОБЕСПЕЧЕНИЯ УСЛУГ СЕТЕВОГО УРОВНЯ В РЕЖИМЕ БЕЗ УСТАНОВЛЕНИЯ СОЕДИНЕНИЯ

Издание официальное

БЗ 7—92/737

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

УДК 681.224:621.391:006.354 Группа П85

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ. ПРОТОКОЛ ДЛЯ ОБЕСПЕЧЕНИЯ УСЛУГ СЕТЕВОГО УРОВНЯ В РЕЖИМЕ БЕЗ УСТАНОВЛЕНИЯ СОЕДИНЕНИЯ

Information technology.
Open systems interconnection.
Protokol for providing the connectionless mode network service

ГОСТ Р 34.1952—92 (ИСО 8473—88)

ОКСТУ 0034

Дата введения 01.01.94

Настоящий стандарт распространяется на сетевой уровень эталонной модели взаимосвязи открытых систем (ВОС) и определяет протокол, обеспечивающий услуги сетевого уровня в режиме без-установления-соединения.

Настоящий стандарт подготовлен методом прямого применения стандарта Международной организации по стандартизации ИСО 8473, за исключением:

а) ссылки на стандарты ИСО заменены ссылками на соответствующие государственные стандарты;

б) упорядочено использование аббревиатур.

0. ВВЕДЕНИЕ

Настоящий стандарт входит в комплекс взаимосвязанных стандартов, разработанных для обеспечения ВОС.

Этот комплекс стандартов охватывает услуги и протоколы, необходимые для достижения совместимости систем в сетях.

Настоящий стандарт следует применять вместе с ГОСТ 28906.

В частности, он определяет протокол сетевого уровня, который может использоваться между логическими объектами сетевого уровня в оконечных системах или в трансляционных систе-

Издание официальное

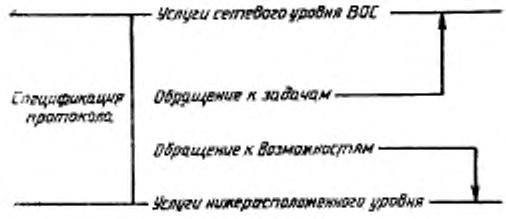
★

© Издательство стандартов, 1993

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

мая сетевого уровня (либо в тех и других). Протокол обеспечивает услуги сетевого уровня в режиме без-установления-соединения, определенные в ГОСТ 34.951

Взаимосвязь между сетевым и нижерасположенным уровнями модели ВОС показана на черт. 1.



Черт. 1

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий стандарт устанавливает протокол, используемый для обеспечения услуг сетевого уровня в режиме без-установления-соединения, определенных в ГОСТ 34.951. Данный протокол использует возможности нижерасположенных услуг режима без-установления-соединения, обеспечиваемых реальными подсетями и/или звеньями данных. Нижерасположенные услуги режима без-установления-соединения, на которые рассчитывает данный протокол, могут быть получены либо непосредственно из реальной подсети, работающей в режиме без-установления-соединения, либо косвенно посредством действий соответствующей функции сходимости, зависимой от подсети (ФСЗП), или протокола ПСЗП через реальную подсеть, работающую в режиме с-установлением-соединения в соответствии с ИСО 8648. Настоящий стандарт определяет способ, посредством которого нижерасположенные услуги подсети режима без-установления-соединения получаются из реальных подсетей, соответствующих ГОСТ 28907 или ГОСТ 34.950. Нижерасположенные услуги подсети режима без-установления-соединения могут быть также получены из таких реальных подсетей, которые не соответствуют ГОСТ 28907 или ГОСТ 34.950, способ их получения не определяется настоящим стандартом.

Примечание. Функция ФСЗП, определяемая для работы настоящего протокола в конфигурациях с подсетями, использующими ППУ X.25, может быть обобщена на операции настоящего протокола в конфигурациях с под-

сетями, использующими ориентированные на соединение протоколы, отличные от ППУ X.25.

Настоящий стандарт устанавливает:

а) процедуры передачи данных и управляющей информации в режиме без-установления-соединения от одного логического объекта к другому равноправному логическому-объекту-сетевого-уровня;

б) кодирование протокольных блоков данных (ПБД), используемых для передачи данных и управляющей информации, включая форматирование протокольных заголовков переменной длины;

в) процедуры корректной интерпретации управляющей протокольной информации;

г) функциональные требования к реализациям, заявляющим о соответствии настоящему стандарту.

Процедуры определяются с точки зрения взаимодействий между:

а) равноправными логическими-объектами-сетевого-уровня путем обмена протокольными блоками данных;

б) логическим-объектом-сетевого-уровня и пользователем услуг сетевого уровня путем обмена примитивами услуг сетевого уровня;

в) логическим-объектом-сетевого-уровня и поставщиком услуг нижерасположенного уровня путем обмена сервисными примитивами.

2. ССЫЛКИ

ГОСТ 28906 (ИСО 7498) «Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель».

ГОСТ 34.950 (ИСО 8208) «Информационная технология. Передача данных. Протокол пакетного уровня X.25 для оконечного оборудования данных».

ГОСТ 34.951 (ИСО 8348) «Информационная технология. Взаимосвязь открытых систем. Услуги сетевого уровня».

ИСО 8348/Доп. 2* «Информационная технология. Взаимосвязь открытых систем. Услуги сетевого уровня. Дополнение 2. Адресация на сетевом уровне».

ИСО/ТО 8509* «Системы обработки информации. Взаимосвязь открытых систем. Соглашения по услугам».

ИСО 8648* «Системы обработки информации. Взаимосвязь открытых систем. Внутренняя организация сетевого уровня».

---

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат ТК 22 «Информационная технология».

ГОСТ 28907 (ИСО 8802/2) «Системы обработки информации. Локальные вычислительные сети. Протокол и услуги уровня управления логическим звеном данных».

ГОСТ 34.913.3 (ИСО 8802/3) «Информационная технология. Локальные вычислительные сети. Метод случайного доступа к шине и спецификация физического уровня».

ГОСТ 34.961 (ИСО 8073—88) «Информационная технология. Соединение открытых систем. Спецификация транспортного протокола, ориентированного на соединение».

ИСО 9074* «Системы обработки информации. Взаимосвязь открытых систем. ESTELLE (метод формализованного описания, основанный на расширенной модели переходов состояний)».

Часть 1. ОБЩИЕ ПОЛОЖЕНИЯ

3. ОПРЕДЕЛЕНИЯ

3.1. В настоящем стандарте использованы следующие понятия, определенные в ГОСТ 28906:

а) оконечная-система;
б) логический-объект-сетевого-уровня;
в) сетевой уровень;
г) протокол сетевого уровня;
д) протокольный блок данных сетевого уровня;
е) трансляция на сетевом уровне;
ж) услуги-сетевого-уровня;
з) пункт-доступа-к-услугам-сетевого-уровня;
и) адрес-пункта-доступа-к-услугам-сетевого-уровня;
к) маршрутизация;
л) услуга;
м) сервисный блок данных;
н) сервисный примитив.

3.2. В настоящем стандарте использованы следующие понятия технического отчета ИСО/ТО 8509:

а) поставщик услуг;
б) пользователь услуг.

3.3. В настоящем стандарте использованы следующие понятия, определенные в стандарте ИСО 8648:

а) промежуточная система;
б) трансляционная система;

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат ТК 22 «Информационная технология».

в) подсеть;
г) протокол сходимости, не зависимый от подсети;
д) функция сходимости, не зависимая от подсети;
е) протокол доступа к подсети.

3.4. В настоящем стандарте использованы следующие понятия из стандарта ИСО 8348/Доп. 2:

а) регион сетевой адресации;
б) адресная информация протокола сетевого уровня;
в) пункт подключения к подсети.

3.5. В настоящем стандарте использованы следующие понятия из ГОСТ 28907:

а) локальная вычислительная сеть;
б) управление логическим звеном;
в) управление доступом к среде.

3.6 В настоящем стандарте использованы следующие понятия из ГОСТ 34.950:

а) аппаратура окончания канала данных;
б) оконечное оборудование данных;
в) логический канал;
г) постоянный виртуальный канал;
д) виртуальный канал.

3.7. Дополнительные определения

3.7.1. Производный ПБД — протокольный блок данных, поля которого идентичны полям исходного ПБД, за исключением того, что он переносит только сегмент данных пользователя из примитива СУ-БЛОК-ДАННЫХ. запрос.

3.7.2. Исходный ПБД — протокольный блок данных, переносящий все данные пользователя из примитива СУ-БЛОК-ДАННЫХ. запрос.

3.7.3. Частный вопрос — решение, принимаемое системой относительно своего поведения на сетевом уровне, т. е. не предписываемое и не ограничиваемое настоящим стандартом.

3.7.4. Наименование логического-объекта-сетевого-уровня — идентификатор логического-объекта-сетевого-уровня, который имеет тот же абстрактный синтаксис, что и адрес ПДУСУ, и который может быть использован для однозначной идентификации логического-объекта-сетевого-уровня в оконечной или промежуточной системе.

3.7.5. Сборка — действия по восстановлению исходного ПБД из двух или более производных ПБД.

3.7.6. Сегмент — отдельная единица данных, содержащая часть данных пользователя или все данные пользователя, обеспечиваемые в примитиве СУ-БЛОК-ДАННЫХ. запрос и доставленные в примитиве СУ-БЛОК-ДАННЫХ. индикация.

3.7.7. Сегментирование — действия по выработке двух или более производных ПБД из одного исходного или производного ПБД. Эти производные ПБД вместе переносят все данные пользователя исходного или производного ПБД, из которого они были образованы.

4. СОКРАЩЕНИЯ

4.1. Блоки данных

ПБД — протокольный блок данных;
СБД — сервисный блок данных;
СБДП — сервисный блок данных подсети;
СБДС — сервисный блок данных сетевого уровня.

4.2. Протокольные блоки данных

ПБД ДН — протокольный блок данных «данные»;
ПБД ОШ — протокольный блок данных «информирование об ошибке»

4.3. Поля протокольного блока данных

АО — адрес отправителя;
АП — адрес получателя;
В/Р — версия/расширение идентификатора протокола;
ВС — время существования;
ДАО — длина адреса отправителя;
ДАП — длина адреса получателя;
ДС — длина сегмента;
ИБД — идентификатор блока данных;
ИПСУ — идентификатор протокола сетевого уровня;
КС — контрольная сумма;
ОД — общая длина;
СС — смещение сегмента;
ТП — тип;
УД — указатель длины;
ФИО — флаг «информирование об ошибке»;
ФДС — флаг «дополнительные сегменты»;
ФСР — флаг «сегментирование разрешено».

4.4. Параметры

АО — адрес отправителя;
АП — адрес получателя;
КУ — качество услуг.

4.5. Разное

АИПСУ — адресная информация протокола сетевого уровня;
АКД — аппаратура окончания канала данных;
НИ — ненумерованная информация;

ООД — оконечное оборудование данных;
ПВК — постоянный виртуальный канал;
ПДП — протокол доступа к подсети;
ПДУСУ — пункт доступа к услугам сетевого уровня;
ППП — пункт подключения к подсети;
ПСЗП — протокол сходимости, зависимый от подсети;
ПСНЗП — протокол сходимости, независимый от подсети;
ПСТ — подсеть;
ПСУ-БУС — протокол сетевого уровня в режиме без-установления-соединения (т. е. протокол, определяемый настоящим стандартом);
УДС — управление доступом к среде;
УЛЗ — управление логическим звеном;
УПП — указатель подключения к подсети;
УСУ — услуги сетевого уровня;
ФСЗП — функция сходимости, зависимая от подсети.

5. КРАТКОЕ ОПИСАНИЕ ПРОТОКОЛА

5.1. Внутренняя организация сетевого уровня

Архитектурная организация сетевого уровня описана в ИСО 8648, который определяет и классифицирует способы выполнения внутренних функций сетевого уровня протоколами сетевого уровня, обеспечивая тем самым унифицированную основу для описания способов использования этих протоколов, действующих по отдельности или во взаимодействии с целью обеспечения услуг сетевого уровня ВОС. Настоящий протокол сетевого уровня ориентирован на использование в контексте метода межсетевого протокола для обеспечения услуг-сетевого-уровня в режиме без-установления-соединения, определенных ИСО 8648.

Настоящий протокол предназначен выполнять роль протокола сходимости, не зависимого от подсети (ПСНЗП). Протокол, выполняющий роль ПСНЗП, функционирует с целью создания услуг-сетевого-уровня ВОС с использованием определенного набора нижерасположенных услуг, выполняя функции, необходимые для обеспечения унифицированного вида услуг сетевого уровня ВОС в режиме без-установления-соединения при работе через однородные или неоднородные взаимосвязанные подсети. Настоящий протокол должен обеспечить необходимую гибкость в тех случаях, когда протоколы сходимости, зависимые от подсети, и/или протоколы доступа к подсети не обеспечивают всех функций, необходимых для обеспечения услуг-сетевого-уровня в ре-

жиме без-установления-соединения по всему маршруту между двумя ПДУСУ, или в какой-то части этого маршрута.

Как указано в ИСО 8648, протокол на сетевом уровне может выполнять различные роли в разных конфигурациях. Несмотря на то, что данный протокол ориентирован конкретно на выполнение роли ПСНЗП в контексте протокола межсетевого взаимодействия для обеспечения услуг-сетевого-уровня в режиме без-установления-соединения, он может выполнять и другие роли, а следовательно, может использоваться в контексте других методов организации взаимосвязи подсетей.

Операции протокола сетевого уровня определены относительно услуг нижерасположенных уровней, доступных через действия других протоколов сетевого уровня или посредством услуг уровня звена данных. Услуги нижерасположенных уровней, на которые рассчитывает данный протокол, описаны в п. 5.5.

5.2. Подмножества протокола

Определены два подмножества полного протокола, которые используют известные характеристики подсетей конкретных конфигураций и, следовательно, не являются независимыми от подсети.

Подмножество *инертный протокол сетевого уровня* является подмножеством с нулевой функциональностью, которое может быть использовано, когда известно, что оконечные-системы отправителя и получателя соединены одной подсетью, и когда ни одна из функций, выполняемых полным протоколом, не требуется для обеспечения услуг-сетевого-уровня в режиме-без-установления-соединения между любой парой оконечных-систем.

Несегментируемое подмножество протокола допускает упрощение заголовка в тех случаях, когда известно, что оконечные-системы отправителя и получателя соединены такими подсетями, у которых длины отдельных сервисных блоков данных больше или равны известному пределу, достаточно большому, чтобы сегментирование не требовалось. Это подмножество выбирается при установлении флага *сегментирование разрешено* в значение ноль (см. п. 6.7).

5.3. Адреса и наименования

5.3.1. Адреса

Параметры *адрес отправителя и адрес получателя*, рассматриваемые в п. 7.3 настоящего стандарта, являются адресами-пунктов-доступа-к-услугам-сетевого-уровня (ПДУСУ) ВОС. Синтаксис и семантика адресов ПДУСУ ВОС описаны в ИСО 8348/Доп.2.

Кодирование передаваемых адресов ПДУСУ, используемое настоящим протоколом, осуществляется в предпочтительном дво-

ичном коде, определенном в ИСО 8348/Доп.2. Адрес ПДУСУ, закодированный в виде последовательности двоичных октетов согласно ИСО 8348/Доп.2, передается полностью в адресных полях, описанных в п. 7.3.

5.3.2. Наименования логических-объектов-сетевого-уровня

Наименование логического-объекта-сетевого-уровня представляет собой идентификатор этого объекта в оконечной или промежуточной системе. Наименования логических-объектов-сетевого-уровня назначаются из того же пространства имен, из которого назначаются адреса ПДУСУ, а определить, в каком случае данное имя является адресом ПДУСУ, а в каком наименованием логического-объекта-сетевого-уровня можно по контексту, в котором интерпретируется данное имя. Значения параметров *маршрутизация со стороны отправителя* и *регистрации маршрута*, определенных в пп. 7.5.4 и 7.5.5 соответственно, являются наименованиями логических-объектов-сетевого-уровня. Значения параметров *адрес отправителя* и *адрес получателя* в ПБД «информирование об ошибке», определенном в п. 7.9, также являются наименованиями логических-объектов-сетевого-уровня.

Кодирование, используемое настоящим протоколом для передачи наименований логических-объектов-сетевого-уровня, осуществляется в предпочтительном двоичном коде, определенном ИСО 8348/Доп.2. Наименование логического-объекта-сетевого-уровня, закодированное в виде последовательности двоичных октетов, согласно ИСО 8348/Доп.2 передается целиком в полях, рассмотренных в пп. 7.5.4, 7.5.5 и 7.9.1.2.

5.4. Услуги, обеспечиваемые протоколом

Протокол сетевого уровня обеспечивает УСУ в режиме без-установления-соединения, описанные в ГОСТ 34.951. Обеспечиваемые примитивы УСУ приведены в табл.

Таблица 1

Примитивы услуг-сетевого-уровня в режиме без-установления-соединения

Примитив	Параметр
СУ-БЛОК-ДАННЫХ ,запрос ,индикация	УСУ-адрес-отправителя УСУ-адрес-получателя УСУ-качество-услуг УСУ-данные-пользователя

Примечание. ГОСТ 34.951 установлено, что максимальная длина СБДС в режиме без-установления-соединения равна 64512 октетов.

5.5. Нижерасположенные услуги, на которые рассчитывает протокол

Задача состоит в том, чтобы протокол сетевого уровня мог функционировать, основываясь на услугах режима без установления-соединения, получаемых из широкого набора реальных подсетей и звеньев данных. Поэтому, чтобы упростить спецификацию протокола, его операции определяются (в разд. 6) относительно абстрактных «нижерасположенных услуг», а не конкретных услуг реальной подсети. Эти нижерасположенные услуги содержат один примитив ПСТ-БЛОК-ДАННЫХ, который переносит адреса пункта подключения подсети — отправителя и получателя, параметр «качество услуг» подсети и минимальное число октетов данных пользователя.

Примитив ПСТ-БЛОК-ДАННЫХ используется для описания абстрактного интерфейса, существующего между протокольным автоматом и нижерасположенной реальной подсетью или функцией сходимости, зависимой от подсети, действующей через реальную подсеть или реальное звено данных для обеспечения требуемых услуг нижерасположенного уровня.

Обеспечиваемые примитивы приведены в табл. 2.

Таблица 2

Сервисные примитивы нижерасположенного уровня

Примитив	Параметр
ПСТ-БЛОК-ДАННЫХ запрос индикация	ПСТ-адрес-отправителя, ПСТ-адрес-получателя, ПСТ-качество-услуг, ПСТ-данные-пользователя

Обеспечение нижерасположенных услуг описано в разд. 8.

5.6. Услуги, ожидаемые от локальных средств

Услуга тайм-аутов должна обеспечиваться для того, чтобы дать возможность логическому объекту протокола упорядочить во времени происходящие события.

Существуют три примитива, относящиеся к услуге СТ-ТАЙМ-АУТ:

а) СТ-ТАЙМ-АУТ. запрос;

б) СТ-ТАЙМ-АУТ. ответ;

в) СТ-ТАЙМ-АУТ. аннулирование.

Эти примитивы приведены в табл. 3.

Примитив СТ-ТАЙМ-АУТ. запрос сообщает локальным средствам, что они должны инициировать тайм-аут с заданными параметрами имени и индекса и отсчитывать его в течение периода, определенного параметром «время».

Примитив СТ-ТАЙМ-АУТ. ответ инициируется локальным средством для информирования о том, что время, запрошенное соответствующим примитивом СТ-ТАЙМ-АУТ. запрос, истекло.

Примитив СТ-ТАЙМ-АУТ. аннулирование сообщает локальным средствам, что определенный(е) тайм-аут(ы) должен(ны) быть аннулирован(ы). Если параметр «СТ-индекс» не определен, то аннулируются все тайм-ауты с заданным именем, в противном случае аннулируется тайм-аут для заданного имени и индекса. Если ни один из тайм-аутов не соответствует определенному параметру, то локальные средства не выполняют никаких действий.

Параметры примитивов услуги СТ-ТАЙМ-АУТ определены в табл. 3.

Таблица 3

Примитивы тайм-аута

Примитив	Параметр
СТ-ТАЙМ-АУТ .запрос	СТ-время СТ-имя СТ-индекс
.ответ	СТ-имя СТ-индекс

Параметр «СТ-время» указывает длительность конкретного тайм-аута. Идентифицирующая метка связана с тайм-аутом посредством параметра «ст-имя». Параметр «СТ-индекс» определяет показатель, различающий тайм-ауты с одинаковым именем. Имя и индекс вместе образуют однозначную ссылку на тайм-аут.

Тайм-ауты, используемые в логической связи с конкретной протокольной функцией, определяются с учетом этой функции.

Примечание. Настоящий стандарт не определяет конкретных значений рассматриваемых тайм-аутов. Любые их значения, указанные в настоящем стандарте, не являются обязательными. Значения тайм-аутов должны выбираться таким образом, чтобы можно было обеспечить запрошенное качество услуг при наличии известных характеристик нижерасположенных услуг.

Часть 2. СПЕЦИФИКАЦИЯ ПРОТОКОЛА

6. ФУНКЦИИ ПРОТОКОЛА

В настоящем разделе описаны функции, выполняемые как часть рассматриваемого протокола.

Не все функции должны выполняться в каждой реализации. В п. 6.19 определены те функции, которые могут быть опущены, и указаны правильные действия в тех случаях, когда запрошенные функции не предусмотрены.

6.1. Функция «Формирование ПБД»

Функция «Формирование ПБД» несет ответственность за формирование протокольного блока данных в соответствии с правилами кодирования ПБД, приведенными в разд. 7. Протокольная управляющая информация, необходимая для доставки блока данных его получателю, определяется исходя из текущего состояния и локальной информации, а также на основании параметров, обеспечиваемых примитивом СУ-БЛОК-ДАННЫХ. запрос.

Адресная информация протокола сетевого уровня (АИПСУ) для полей *адрес отправителя* и *адрес получателя* заголовка ПБД образуется из параметров «УСУ-адрес-отправителя» и «УСУ-адрес-получателя». Параметры «УСУ-адрес-получателя» и «УСУ-качество-услуг», вместе с текущим состоянием и локальной информацией используются для определения необходимого выбора факультативных функций. Данные пользователя, передаваемые от пользователя УСУ (УСУ-данные-пользователя), образуют часть «данные» ПБД.

При формировании ПБД назначается *идентификатор блока данных,* который должен отличать запрос на передачу «УСУ-данные-пользователя» конкретному адресуемому пользователю УСУ от других подобных запросов. Отправитель ПБД должен выбрать идентификатор блока данных таким образом, чтобы он оставался уникальным (для данной пары адресов отправителя и получателя) в течение максимального времени существования исходного ПБД в данной сети; это правило применяют для любого ПБД, образованного из исходного ПБД в результате применения функции «сегментирование» (см. п. 6.7). Производные ПБД рассматриваются как соответствующие тому же исходному ПБД и, следовательно, тому же примитиву СУ-БЛОК-ДАННЫХ. запрос, если эти ПБД имеют одинаковые адреса отправителя и получателя и одинаковый идентификатор блока данных.

Идентификатор блока данных используется также для вы-

полнения служебных функций, например для информирования об ошибках (см. п. 6.10).

Общая длина ПБД, выраженная в октетах, определяется его отправителем и помещается в поле *общая длина* заголовка ПБД. Это поле не изменяется ни в одном производном ПБД в течение всего времени существования данного ПБД.

При использовании несегментируемого подмножества протокола отсутствует как поле «общая длина», так и поле «идентификатор блока данных». Правила, определяющие функцию «формирование ПБД», модифицируются в данном случае следующим образом. В процессе формирования ПБД значение общей его длины, выраженное в октетах, определяется отправителем и помещается в поле *длина сегмента* заголовка ПБД. Это поле не изменяется в течение всего времени существования ПБД. Идентификация блока данных не обеспечивается.

6.2. Функция «разборка ПБД»

Эта функция несет ответственность за извлечение из ПБД протокольной управляющей информации. Во время этого процесса информация, относящаяся к генерации примитива СУ-БЛОК-ДАННЫХ. индикация определяется следующим образом. Параметры «УСУ-адрес-отправителя» и «УСУ-адрес-получателя» примитива СУ-БЛОК-ДАННЫХ. индикация извлекаются из АИПСУ и помещаются в поля *адрес отправителя* и *адрес получателя* заголовка ПБД. Часть «данные» принятого ПБД сохраняется до тех пор, пока не будут приняты все сегменты исходного СБД; все это в совокупности формирует параметр «УСУ-данные-пользователя» примитива СУ-БЛОК-ДАННЫХ. индикация. Информация, относящаяся к качеству услуг, обеспечиваемых при передаче ПБД, определяется качеством услуг и другой информацией, содержащейся в факультативной части заголовка ПБД. Эта информация образует параметр «УСУ-качество-услуг» примитива СУ-БЛОК-ДАННЫХ. индикация.

6.3. Функция «анализ формата заголовка»

Эта функция определяет, используется ли весь протокол, описанный в настоящем стандарте, или только одно из соответствующих определенных его подмножеств. Если ПБД содержит идентификатор протокола сетевого уровня, который показывает, что это стандартная версия протокола, то данная функция определяет, доставлен ли принятый ПБД своему получателю, адрес которого указан в заголовке ПБД. Если адрес получателя, содержащийся в ПБД, идентифицирует ПДУСУ, обслуживаемый данным логическим объектом сетевого уровня, то этот ПБД бу-

дет доставлен своему адресату, если нет — он должен быть передан дальше.

Если ПБД содержит идентификатор протокола сетевого уровня, который показывает, что используется подмножество «инертный протокол сетевого уровня», то никакого дальнейшего анализа заголовка ПБД не требуется. В этом случае логический-объект-сетевого-уровня определяет одно из двух: либо адрес пункта подключения подсети, закодированный в виде АИПСУ в поддерживающем протоколе подсети, соответствует адресу ПДУСУ, обслуживаемого этим логическим-объектом-сетевого-уровня, либо имеет место ошибка.

6.4. Функция «управление временем существования ПБД»

Эта функция используется для обеспечения максимального времени существования ПБД. Она тесно связана с функцией «анализ формата заголовка». Эта функция определяет, может ли принятый ПБД передаваться дальше, либо заданное для него время существования кончилось и он должен быть аннулирован.

Действия функции «управление временем существования ПБД» зависят от поля *время существования* заголовка ПБД. Это поле содержит в любой момент времени оставшуюся часть времени существования ПБД (представленное в единицах 500 мс). Время существования исходного ПБД определяется логическим-объектом-сетевого-уровня отправителя и помещается в поле «время существования» ПБД. Если функция «сегментирование» применима к ПБД, то значение поля «время существования» исходного ПБД копируется во все производные ПБД.

Время существования ПБД уменьшается каждым логическим-объектом-сетевого-уровня, обрабатывающим ПБД. Когда логический-объект-сетевого-уровня обрабатывает ПБД, он уменьшает время существования ПБД, по меньшей мере, на единицу. Значение поля «время существования» ПБД должно уменьшаться больше чем на единицу, если сумма двух значений:

а) транзитной задержки в нижерасположенных услугах, из которых был принят ПБД;

б) задержки внутри системы, обрабатывающей ПБД, превышают или оценочно превышает 500 мс. В этом случае поле «время существования» должно уменьшаться на единицу за каждые дополнительные 500 мс задержки. Определение задержки не обязательно должно быть точным, но в тех случаях, когда точное значение не может быть установлено, используемое значение следует завышать, но не занижать.

Если поле «время существования» достигнет нулевого значения

до того, как ПБД будет доставлен получателю, то этот ПБД должен быть аннулирован. Функция «информирование об ошибке» должна привлекаться в соответствии с изложенным в п. 6.10. Она может привести к генерации ПБД «информирование об ошибке».

Вопрос о том, выполняет логический-объект-сетевого-уровня функцию «управление временем существования» или нет, является частным вопросом.

6.5. Функция «маршрутизация ПБД»

Данная функция определяет тот логический-объект-сетевого-уровня, которому должен быть направлен ПБД, и те услуги нижерасположенного уровня, которые должны использоваться для достижения этого логического-объекта-сетевого-уровня с использованием полей этого ПБД «адрес получателя» и «общая длина». При необходимости сегментирования функция «маршрутизация ПБД», кроме того, определяет, с помощью какой услуги нижерасположенного уровня должны передаваться производные ПБД/ /сегменты, чтобы достичь этого логического-объекта-сетевого-уровня. Результаты выполнения функции «маршрутизация ПБД» сообщаются функции «продвижение ПБД» (вместе с самим ПБД) для дальнейшей обработки.

Выбор услуги нижерасположенного уровня, которая должна использоваться для достижения «следующей» в маршруте системы, вначале зависит от параметра «УСУ-качество-услуг» примитива СУ-БЛОК-ДАННЫХ. запрос, который определяет КУ, запрошенное передающим пользователем УСУ. Вопрос о том, обеспечивается ли это КУ непосредственно протоколом ПСУ-БУС путем выбора параметра «СУ-качество-услуг» и других факультативных параметров или с помощью средств КУ, обеспечиваемых каждой из услуг нижерасположенного уровня, определяется до привлечения функции «продвижение ПБД». Выбор маршрута промежуточными системами может впоследствии зависеть от значений параметра «СУ-качество-услуг» (при его наличии) и других факультативных параметров (при их наличии).

6.6. Функция «продвижение ПБД»

Данная функция выдает примитив ПСТ-БЛОК-ДАННЫХ. запрос (см. п. 5.5), который снабжает подсеть или ФСЗП, идентифицируемую функцией «маршрутизация ПБД», протокольным блоком данных в виде подлежащих передаче данных пользователя, адресной информацией, необходимой для этой подсети или для ФСЗП с целью идентификации «следующей» системы в пределах специфичного для подсети региона адресации (это может быть промежуточная система или адресуемая оконечная-система),

а также предельными значениями качества услуг (при их наличии), которые должны рассматриваться в процессе обработки данных пользователя.

Если подлежащий передаче ПБД длиннее максимального СБД, обеспечиваемого услугами нижерасположенного уровня, то используется функция «сегментирование» (см. п. 6.7).

6.7. Функция «сегментирование»

Сегментирование выполняется в том случае, когда длина ПБД превышает максимальную длину СБД, обеспечиваемую услугами нижерасположенного уровня, которые должны использоваться для передачи данного ПБД.

Сегментирование — это образование двух или более новых ПБД (производных ПБД) из принятого ПБД. Принятым ПБД может быть исходный ПБД или производный ПБД. Вся информация заголовка ПБД, подлежащего сегментированию, за исключением полей фиксированной части ПБД «длина сегмента» и КПБ, а также смещения сегмента сегментируемой части, удваивается в каждом производном ПБД, включая всю адресную часть, идентификатор блока данных и общую длину сегментируемой части, а также факультативные части (при их наличии).

Примечание. Правила продвижения и сегментирования гарантируют, что длина заголовка будет одинакова для всех сегментов (производных ПБД) исходного ПБД и равна длине заголовка исходного ПБД. Длина заголовка ПБД не должна изменяться под влиянием любой функции протокола.

Данные пользователя, скомпонованные в принятом ПБД, делятся таким образом, чтобы производные ПБД удовлетворяли требованиям, предъявляемым к длине поля параметра «ПСТ-данные-пользователя» примитива ПСТ-БЛОК-ДАННЫХ. запрос, который используется для доступа к услугам нижерасположенных уровней. Каждый производный ПБД, кроме последнего, должен содержать число октетов, не равное нулю и кратное 8. Таким образом, значение поля «смещение сегмента» в любом ПБД либо равно нулю, либо не равно нулю, но кратно 8. Сегментирование не должно приводить к образованию таких производных ПБД, которые содержит менее восьми октетов данных пользователя.

Производные ПБД идентифицируются как образованные из одного и того же исходного ПБД посредством поля:

а) «адрес отправителя»;

б) «адрес получателя»;

в) «идентификатор блока данных».

Следующие поля заголовка ПБД используются в сочетании с функцией «сегментирование»:

а) *смещение сегмента* — идентифицирует октет, с которого начинается сегмент относительно начала передачи исходного ПБД;

б) *длина сегмента* — определяет число октетов в производном ПБД, включая заголовок и данные;

в) *флаг дополнительных сегментов* — устанавливается в единицу, если данный производный ПБД не содержит последнего октета исходного ПБД в качестве своего последнего октета данных пользователя;

г) *общая длина* — определяет всю длину исходного ПБД, включая заголовок и данные.

Производные ПБД могут сегментироваться далее без каких-либо ограничений на маршруты отдельных производных ПБД.

Флаг *сегментирование разрешено* устанавливается в единицу, показывая, что сегментирование разрешено. Если исходный ПБД не подлежит сегментированию в любой момент времени его существования в сети, то этот флаг устанавливается в ноль логическим-объектом-сетевого-уровня отправителя. Установленное значение флага «сегментирование разрешено» не может быть изменено никаким другим логическим-объектом-сетевого-уровня в течение времени существования исходного ПБД и любых производных ПБД.

6.8. Функция «сборка»

Функция «сборка» восстанавливает исходный ПБД из производных ПБД, образованных в результате операций функции «сегментирование» над исходными ПБД (и, рекурсивно, над последующими производными ПБД).

Предусмотрен временной предел, в течение которого сегменты (производные ПБД) исходного ПБД будут храниться в месте сборки до аннулирования; поэтому ресурсы сборки могут быть освобождены, когда больше не ожидается поступления в место сборки никаких неподтвержденных сегментов исходного ПБД. При приеме производного ПБД тайм-аут сборки начинает отсчет со значения, показывающего промежуток времени, который должен истечь до того момента, когда можно предполагать, что все неподтвержденные сегменты исходного ПБД потеряны. При истечении этого тайм-аута все имеющиеся в месте сборки сегменты (производные ПБД) исходного ПБД аннулируются; ресурсы, выделенные для этих сегментов, освобождаются и генерируется ПБД «Информирование об ошибке», если он используется (см. п. 6.10).

Хотя точное взаимоотношение между продолжительностью сборки и временем существования ПБД является частным вопросом, функция «сборка» должна сохранять смысл времени существования ПБД. Следовательно, функция сборки должна аннулировать те ПБД, время существования которых могло бы за-

кончиться, если бы они не находились под управлением функции сборки, т. е. продолжительность сборки для данного ПБД должна быть меньше, чем время существования каждого производного ПБД, находящегося в месте сборки.

Примечания:

1. Методы ограничения продолжительности сборки рассматриваются в приложении В.

2. Функции «сегментирование» и «сборка» ориентированы на такое использование, чтобы в каждом пункте сегментирования вырабатывалось как можно меньшее число сегментов, а сборка происходила в конечном пункте назначения ПБД. Однако возможны и другие варианты, при которых не запрещено:

а) взаимодействие с алгоритмом маршрутизации по предпочтительным маршрутам, где вырабатывается меньшее число сегментов;

б) выработка большего числа сегментов, чем необходимо для того, чтобы избежать дополнительного сегментирования в каком-то последующем пункте;

в) допускается частичная или полная сборка в некотором промежуточном пункте маршрута.

Информация, необходимая для использования одной из этих альтернативных стратегий, может быть получена в результате действий функции «управление сетевым уровнем» или другими способами.

3. Отправитель исходного ПБД определяет значение флага «сегментирование разрешено» в исходном ПБД и во всех производных ПБД (при их наличии). Частичная или полная сборка в промежуточной системе (см. примечание 2в) не может изменять этого значения в исходном ПБД или в любом производном от него ПБД и поэтому не может добавлять в заголовок или удалять из него сегментируемую часть.

6.9. Функция «аннулирование ПБД»

Данная функция выполняет все действия, необходимые для освобождения ресурсов, зарезервированных логическим-объектом-сетевого уровня, в следующих встречающихся ситуациях (следует заметить, что приводимый перечень не является исчерпывающим):

а) произошло нарушение протокольной процедуры;

б) получен ПБД, чья контрольная сумма не соответствует его содержимому;

в) получен ПБД, но вследствие локальной перегрузки он не может быть обработан;

г) получен ПБД, заголовок которого не может быть проанализирован;

д) получен ПБД, который невозможно сегментировать и продвигать дальше, поскольку его длина превышает максимальную длину СБД, обеспечиваемого любой услугой нижерасположенного уровня, способной передавать ПБД следующему логическому-объекту-сетевого-уровня в выбранном маршруте;

е) получен ПБД, получатель которого недоступен или неизвестен;

ж) со стороны отправителя был определен неправильный или недействительный маршрут. Сюда могут относиться синтакси-

ческая ошибка в поле маршрутизации отправителя, неизвестное, или недействительное наименование логического-объекта-сетевого-уровня в поле маршрутизации отправителя, либо маршрут, недоступный по другим причинам;

з) получен ПБД, время существования которого истекло или истекает во время сборки;

и) получен ПБД, содержащий необеспечиваемую факультативную возможность.

6.10. Функция «информирование об ошибке».

6.10.1. Краткое описание

Функция «информирование об ошибке» пытается передать ПБД «информирование об ошибке» логическому-объекту-сетевого-уровня — отправителю, когда протокольный блок данных аннулируется в соответствии с п. 6.9.

ПБД «информирование об ошибке» идентифицирует аннулированный ПБД, определяет тип обнаруженной ошибки и место в заголовке аннулированного ПБД, где обнаружена ошибка. Весь заголовок аннулированного ПБД и, по усмотрению отправителя ПБД «информирование об ошибке», часть «данные» помещается целиком, частично или вообще не помещается в часть «данные» этого ПБД.

Отправитель ПБД «данные» управляет генерацией ПБД «информирование об ошибке». Флаг *информирования об ошибке* в исходном ПБД устанавливается логическим-объектом-сетевого-уровня — отправителем для того, чтобы указать на необходимость передачи ПБД «информирование об ошибке», если исходный ПБД или любые производные от него ПБД аннулируются. Если флаг не установлен, то такие информирующие об ошибке ПБД не генерируются.

Примечания:

1. Подавление блоков ПБД «информирование об ошибке» происходит под управлением логического-объекта-сетевого-уровня — отправителя, но не пользователя УСУ. Отправитель должен внимательно следить за подавлением ПБД ОШ с тем, чтобы информирование об ошибке не подавлялось для каждого генерируемого ПБД.

2. Неполучение ПБД «информирование об ошибке» еще не означает правильной доставки ПБД, выданного логическим-объектом-сетевого-уровня-отправителем

6.10.2. Требования к функции «информирование об ошибке»

ПБД «информирование об ошибке» не должен генерироваться для информирования об аннулировании ПБД «информирование об ошибке».

ПБД «информирование об ошибке» не должен генерироваться для информирования об аннулировании ПБД «данные», если только для этого ПБД не был установлен флаг «информирование

об ошибке» в значение, разрешающее информирование об ошибке.

Если ПБД «данные» аннулирован и флаг «информирование об ошибке» установлен в значение, разрешающее информирование об ошибке, то должен быть сгенерирован ПБД «информирование об ошибке», если причиной аннулирования является одна из перечисленных в п. 6.9, которая вызвана условиями, указанными в п. 6.10.4.

Примечание. Если ПБД «данные», у которого флаг «информирование об ошибке» установлен в значение, разрешающее информирование об ошибке, аннулируется по любой другой причине, то ПБД ОШ может быть сгенерирован (как факультативная возможность реализации).

6.10.3. Обработка ПБД «информирование об ошибке»

ПБД «информирование об ошибке» состоит из информации, содержащейся в заголовке аннулированного ПБД «данные», к которому относится информирование об ошибке. Содержимое поля «адрес отправителя» аннулированного ПБД «данные» используется в качестве *адреса получателя ПБД* «информирование об ошибке». Этот параметр, который в контексте ПБД «данные» использовался как адрес ПДУСУ, в контексте ПБД «информирование об ошибке» используется как наименование того логического-объекта-сетевого-уровня, который отправил ПБД «данные». Наименование логического-объекта-сетевого-уровня отправителя ПБД «информирование об ошибке» передается в поле *адрес отправителя* заголовка ПБД «информирование об ошибке». Значение поля «время существования» определяется в соответствии с п. 6.4. Факультативные параметры выбираются в соответствии с п. 6.10.4.

Сегментирование ПБД «информирование об ошибке» не разрешается; следовательно, сегментируемые части здесь отсутствуют. Общая длина ПБД ОШ, выраженная в октетах, размещается в поле *длина сегмента* заголовка ПБД ОШ. Это поле не изменяется в течение всего времени существования ПБД ОШ. Если отправитель ПБД ОШ определяет, что длина ПБД ОШ превышает максимальную длину СБД услуги нижерасположенного уровня, то ПБД ОШ должен быть усечен до максимальной длины СБД (см. п. 8.3) и передан дальше без каких-либо других изменений. ПБД «информирование об ошибке» маршрутизируется и продвигается логическими-объектами-сетевого-уровня промежуточной системы таким же способом, как и ПБД «данные».

Примечание. Изложенное в п. 8.3 требование, чтобы услуги нижерасположенного уровня, на которые рассчитывает ПСУ-БУС, были способны обеспечивать СБД длиной 512 октетов, гарантирует, что, по меньшей мере, весь заголовок аннулированного ПБД «данные» может передаваться в части «данные» любого ПБД ОШ.

Если ПБД ОШ разбивается на части при достижении своего пункта назначения, то информация, которая может использоваться для интерпретации ПБД «информирование об ошибке» и действий над ним, получается следующим образом. Наименование логического-объекта-сетевого-уровня, получаемое из АИПСУ в поле «адрес отправителя» заголовка ПБД ОШ, используется для идентификации логического — объекта-сетевого-уровня, генерирующего ПБД «информирование об ошибке». Причина генерации этого ПБД извлекается из факультативной части заголовка ПБД. Для участия в оценке характера ошибки весь заголовок аннулированного ПБД «данные» (а также часть или все исходные данные пользователя) извлекаются из части «данные» ПБД ОШ.

6.10.4. Взаимоотношение факультативных возможностей ПБД «данные» с информированием об ошибках

Генерация ПБД «информирование об ошибке» зависит от факультативных возможностей, имеющихся в соответствующем ПБД «данные». Наличие факультативных возможностей в исходном ПБД «данные», которые не обеспечиваются системой, аннулировавшей данный ПБД, может вызвать подавление ПБД «информирование об ошибке», даже если исходный ПБД «данные» показывает, что ПБД «информирование об ошибке» должен быть сгенерирован в случае аннулирования.

Обработка ПБД «информирование об ошибке» зависит также от факультативных возможностей, имеющихся в соответствующем ПБД «данные». Факультативные возможности, выбранные для исходного ПБД «данные», влияют на факультативные возможности, включаемые в соответствующий ПБД «информирование об ошибке». Выбор факультативных возможностей для ПБД «информирование об ошибке» определяется следующими требованиями:

а) если факультативная возможность «приоритет» или «обеспечение КУ» выбрана в исходном ПБД «данные» и система, генерирующая ПБД «информирование об ошибке», обеспечивает эту факультативную возможность, то ПБД «информирование об ошибке» должен определить ту же факультативную возможность, используя ее значение, определенное в исходном ПБД «данные»;

б) если факультативная возможность «защита» выбрана в ПБД «данные» и система, генерирующая информирование об ошибке, обеспечивает эту факультативную возможность, то ПБД «информирование об ошибке» должен определить эту факультативную возможность, используя значение, определенное в исходном ПБД «данные».

Если система не обеспечивает факультативной возможности «защита», то информирование об ошибке не должно генерироваться для ПБД «данные», который выбрал эту факультативную возможность;

в) если факультативная возможность «полная маршрутизация со стороны отправителя» выбрана в исходном ПБД «данные» и система, генерирующая ПБД «информирование об ошибке», обеспечивает эту факультативную возможность, то информирование об ошибке должно определить факультативную возможность «полная маршрутизация со стороны отправителя». Значение параметра «маршрутизация со стороны отправителя» получается путем извлечения из исходного ПБД «данные» той части полного маршрута со стороны отправителя, который уже пройден, и изменения порядка наименований логических-объектов-сетевого-уровня, входящих в список.

Если система не обеспечивает факультативной возможности «полная маршрутизация со стороны отправителя», то информирование об ошибке не должно генерироваться для ПБД «данные», который выбирает эту факультативную возможность;

г) факультативные возможности «заполнение», «частичная маршрутизация со стороны отправителя» и «регистрация маршрута» при их обеспечении могут быть определены в ПБД «информирование об ошибке».

Примечание. Значения факультативных параметров в подпункте г) могут быть получены в виде частного решения или могут основываться на соответствующих значениях исходного ПБД «данные».

6.11. Обнаружение ошибки в заголовке ПБД

Функция «обнаружение ошибки в заголовке ПБД» защищает от ошибок логический-объект-сетевого уровня промежуточной или оконечной системы, обусловленной обработкой ошибочной информации в заголовке ПБД. Данная функция реализуется с помощью контрольной суммы, вычисляемой на основе всего заголовка ПБД. Контрольная сумма проверяется в каждом пункте, где обрабатывается заголовок ПБД. Если результат вычисления контрольной суммы отрицательный, то соответствующий ПБД должен быть аннулирован. Если поля заголовка ПБД смодифицированы (например вследствие действий функции «время существования»), то контрольная сумма изменяется таким образом, чтобы остаться правильной.

Использование функции «обнаружение ошибок в заголовке» является факультативным и оно определяется логическим-объектом-сетевого-уровня-отправителем. Если эта функция не исполь-

зуется, то поле «контрольная сумма» заголовка ПБД устанавливается в нуль.

Если использование данной функции выбрано логическим-объектом-сетевого-уровня-отправителем, то значение поля «контрольная сумма» обуславливает необходимость выполнения формул

$$\sum_{i=1}^{L} a_i \text{ (мод. 255)} = 0,$$

$$\sum_{i=1}^{L} (L - i + 1) a_i \text{ (мод. 255)} = 0,$$

где L — число октетов в заголовке ПБД;

a_i — значение октета в позиции i. Считается, что первый октет заголовка ПБД занимает позицию $i = 1$.

Если данная функция используется, то ни один из октетов поля «контрольная сумма» не может быть установлен в нуль.

Примечания:

1. Чтобы функция «ошибка в заголовке ПБД» сохраняла возможность обнаруживать случайную модификацию заголовка, происшедшую при обработке ПБД промежуточной системой (например из-за неисправности запоминающего устройства), логический-объект-сетевого-уровня промежуточной системы не должен повторно вычислять контрольную сумму для всего заголовка, даже если его поля изменились.

2. В приложении С приведено описание алгоритмов, которые могут использоваться для вычисления правильного значения поля «контрольная сумма» при формировании ПБД и для обновления значения поля «контрольная сумма» при модификации заголовка.

6.12. Функция «заполнение»

Функция «заполнение» предусмотрена для того, чтобы зарезервировать в заголовке ПБД пространство, которое не используется при обеспечении любой другой функции. Кратность октету должна сохраняться.

Примечание. Пример использования этой функции состоит в том, чтобы обеспечить начало части «данные» ПБД на удобной для логического-объекта-сетевого-уровня — отправителя границе, например на границе машинного слова.

6.13. Защита данных

Обеспечение услуг по защите данных (например аутентификация источника данных, секретность данных и целостность данных отдельного СБДС в режиме без-установления-соединения) выполняется функцией «защита».

Функция «защита» относится к параметру качества услуг *защита от несанкционированного доступа,* описанного в ГОСТ 34.951. Функция реализуется путем выбора параметра «защита» в факультативной части заголовка ПБД.

В данном стандарте не определен сам метод обеспечения услуг защиты; стандарт предусматривает только кодирование информации защиты в заголовке ПБД. Для упрощения взаимодействия между оконечными-системами и трансляционными системами сетевого уровня путем устранения различных интерпретаций одних и тех же кодов в п. 7.5.3 указывается, как отличить коды защиты, определяемые пользователем, от стандартизованных кодов защиты.

Примечание. В качестве решения, относящегося к конкретной реализации, идентификация отправителя данных может осуществляться путем использования криптографически генерируемой или зашифрованной контрольной суммы (в отличие от вырабатываемой механизмом обнаружения ошибок в заголовке ПБД); секретность и целостность данных может обеспечиваться механизмами управления маршрутизацией.

6.14. Функция «маршрутизация со стороны отправителя»

Функция «маршрутизация со стороны отправителя» позволяет самому отправителю определить маршрут передачи сформированного ПБД. Маршрутизация со стороны отправителя может осуществляться только источником данного ПБД и она выполняется путем использования списка наименований логических-объектов-сетевого-уровня, содержащихся в параметре факультативной части заголовка ПБД. Длина этого параметра определяется логическим-объектом-сетевого-уровня — отправителем и не изменяется в течение всего времени существования ПБД.

Параметр «маршрутизация со стороны отправителя» содержит информацию, используемую оконечной-системой — отправителем при определении исходного маршрута ПБД. В списке содержатся наименования логических-объектов-сетевого-уровня только промежуточной системы; наименование логического-объекта-сетевого-уровня — получателя ПБД отсутствует в списке.

С каждым списком наименований логических-объектов-сетевого уровня связан указатель, который идентифицирует следующий, подлежащий использованию элемент списка; этот указатель изменяется получателем ПБД, когда следующее наименование в списке совпадает с его собственным. Данный указатель модифицируется по мере продвижения ПБД так, чтобы можно было идентифицировать соответствующий элемент в каждой фазе трансляции.

Обеспечиваются два вида функции «маршрутизация со стороны отправителя». Первый вид, называемый «полная маршрутизация со стороны отправителя», требует выбора определенного маршрута, то есть ПБД по пути к пункту назначения может проходить только через идентифицированные в списке системы и все системы он должен проходить в заданной последовательности.

Если заданный маршрут не может быть использован, ПБД должен быть аннулирован. В п. 6.10 описаны ситуации, при которых должна осуществляться попытка информировать отправителя об аннулировании ПБД с использованием функции «информирование об ошибке».

Второй вид функции называется «частичная маршрутизация со стороны отправителя». Как и при полной маршрутизации со стороны отправителя ПБД по пути к пункту назначения должен проходить через каждую идентифицированную в списке систему в заданной последовательности. Однако при этом виде маршрутизации ПБД может выбирать любой путь к следующей промежуточной системе списка, в том числе проходить промежуточные системы, не указанные в списке. ПБД не будут аннулироваться (по причинам, связанным с источником маршрутизации), если только какая-либо из указанных систем не может быть достигнута по любому доступному маршруту.

6.15. Функция «регистрация маршрута»

Функция «регистрация маршрута» регистрирует маршруты(ы), по которому(ым) следует ПБД, проходя последовательность промежуточных систем. Зарегистрированный маршрут содержит список наименований логических-объектов-сетевого-уровня, содержащихся в параметре факультативной части заголовка ПБД. Длина этого параметра определяется логическим-объектом-сетевого-уровня-отправителем и не изменяется в течение всего времени существования ПБД.

Список составляется по мере продвижения ПБД по маршруту к его конечному пункту назначения. В регистрацию маршрута включаются только наименования логических-объектов-сетевого-уровня промежуточных систем. Наименование логического-объекта-сетевого-уровня-отправителя ПБД не регистрируется в списке.

Когда логический-объект-сетевого-уровня промежуточной системы обрабатывает ПБД, содержащий параметр «регистрация маршрута», эта система добавляет в конец списка регистрируемых наименований наименование своего логического-объекта-сетевого-уровня. Предусмотрен указатель для идентификации следующего доступного октета, который подлежит использованию для регистрации маршрута. При добавлении элементов в список этот указатель изменяется следующим образом. Длина элемента, подлежащего добавлению к списку, присоединяется к значению указателя следующего доступного октета и полученная сумма сравнивается с длиной параметра «регистрация маршрута». Если в результате добавления элемента в список длина параметра окажется превышенной, устанавливается указатель следующего

доступного октета, показывая, что регистрация маршрута закончена. При этом наименование логического-объекта-сетевого-уровня не добавляется в список. ПБД может далее продвигаться к своему конечному пункту назначения без последующих добавлений наименований логических-объектов-сетевого-уровня.

Если добавление элемента не приводит к превышению длины параметра «регистрация маршрута», то указатель следующего доступного октета изменяет свое значение и наименование логического-объекта-сетевого-уровня добавляется в конец списка.

Обеспечиваются два вида функции «регистрация маршрута». Первый вид называется «полная регистрация маршрута». Он требует, чтобы список наименований логических-объектов-сетевого-уровня был полным и точно регистрировал все промежуточные системы, проходимые ПБД (в том числе производными ПБД), исключая случай, когда недостаток места в поле факультативной регистрации маршрута приводит к окончанию регистрации маршрута, как описано выше. Если выбрана полная регистрация маршрута, то сборка ПБД в промежуточных системах выполняется только в том случае, если производные ПБД, которые должны собираться, проходят по одному и тому же маршруту; в противном случае ПБД аннулируется и генерируется ПБД «информирование об ошибке» (если он используется) (см. п. 6.10).

Второй вид функции называется «частичная регистрация маршрута». Он также требует регистрации всех промежуточных систем, которые проходит ПБД. Если выбирается частичная регистрация маршрута, то сборка ПБД в промежуточных системах всегда разрешается. При выполнении сборки в промежуточной системе маршрут, регистрируемый в любом из производных ПБД, может быть указан в ПБД, полученном в результате сборки.

Примечание. Функция «регистрация маршрута» предназначена для использования при решении проблем диагностики подсети и/или для обеспечения обратного маршрута, который может использоваться в качестве маршрута со стороны отправителя в последующем ПБД.

6.16. Функция «обеспечение качества услуг»

Функция «обеспечение качества услуг» обеспечивает логическим-объектам-сетевого-уровня в промежуточных системах информацию, которая может использоваться для принятия решений о маршрутизации в тех случаях, когда эти решения влияют на общее КУ, обеспечиваемое для пользователей УСУ. Эта информация передается логическим-объектам-сетевого-уровня промежуточной системы в параметре факультативной части заголовка ПБД.

В тех случаях, когда требуемое КУ не может быть обеспечено, логические-объекты-сетевого-уровня промежуточной системы дол-

жны попытаться доставить ПБД с КУ, отличным от запрашиваемого. Логические-объекты-сетевого-уровня промежуточной системы не обязательно должны информировать о том, что запрошенное качество услуг не обеспечено.

6.17. Функция «приоритет»

Функция «приоритет» обеспечивает предпочтительную обработку тех ПБД, которые имеют более высокое численное значение приоритета по сравнению с другими ПБД. Эта функция реализуется путем выбора параметра в факультативной части заголовка ПБД.

Наинизшее значение приоритета равно нулю. Функция «приоритет» обеспечивает средство, с помощью которого ресурсы логических-объектов-сетевого-уровня оконечной и промежуточной систем, такие как буферы и очереди, могут использоваться для предпочтительной обработки ПБД с более высоким приоритетом по отношению к обработке ПБД с более низким приоритетом. Конкретные действия, выполняемые отдельным логическим-объектом-сетевого-уровня для обеспечения функции «приоритет», являются частным вопросом.

6.18. Функция «информирование о перегрузке»

Для того, чтобы пользователь УСУ мог предпринять соответствующие действия при возникновении перегрузки у поставщика УСУ, промежуточная система может проинформировать логический-объект-сетевого-уровня получателя о перегрузке, используя флаг в параметре «обеспечение КУ» факультативной части заголовка ПБД. Вначале отправитель ПБД устанавливает значение этого флага в нуль (0), а любая промежуточная система, обрабатывающая ПБД, может установить его в единицу (1) для информирования о возникшей перегрузке. Вопрос о том, когда нужно выполнять это действие, является частным вопросом.

Примечание. Обычно перегрузка связана с недоступностью буферной области для хранения выходных очередей. Соответствующие правила информирования о перегрузке могут зависеть от размера очереди на выходе, выбранной для ПБД (в соответствии с адресом получателя или другой информацией о маршрутах). Если конкретная очередь на выходе превышает определенный размер этой очереди, то промежуточная система может начать «инулирование ПБД. Промежуточная система может установить флаг «наличие перегрузки» в следующем подлежащем продвижению ПБД и продолжать действовать так до тех пор, пока не произойдет разгрузка.

6.19. Классификация функций

Конкретные реализации не требуют обеспечения всех функций, перечисленных в пп. 6.1—6.18. Эти функции подразделяются на три типа:

Тип 1. Эти функции должны обеспечиваться.

Тип 2. Эти функции могут обеспечиваться или не обеспечиваться. Если реализация не обеспечивает функции типа 2, а такая функция выбрана в ПБД, то этот ПБД должен быть аннулирован, сгенерирован ПБД «информирование об ошибке» и направлен логическому-объекту-сетевого-уровня отправителя при условии, что флаг «информирование об ошибке» установлен и условия п. 6.10.4 соблюдаются.

Тип 3. Эти функции могут обеспечиваться или не обеспечиваться. Если реализация не обеспечивает функцию типа 3, а такая функция выбрана в ПБД, то она не выполняется и ПБД обрабатывается точно так же, как и в случае, когда эта функция не выбрана. По этой причине протокольный блок данных не должен аннулироваться.

В табл. 4 показано разделение функций на эти три типа.

Таблица 4

Классификация функций протокола

Функция	Полный протокол	Несегментируемое подмножество	Инертное подмножество
Формирование ПБД	1	1	1
Разборка ПБД	1	1	1
Анализ формата заголовка	1	1	1
Управление временем существования ПБД	1	1	Н/И
Маршрутизация ПБД	1	1	Н/И
Продвижение ПБД	1	1	Н/И
Сегментирование ПБД	1	Н/И	Н/И
Сборка ПБД	1	Н/И	Н/И
Аннулирование ПБД	1	1	Н/И
Информирование об ошибках	1	1	Н/И
Обнаружение ошибок в заголовке	1	1	Н/И
Защита	2	2	Н/И
Полная маршрутизация со стороны отправителя	2	2	Н/И
Полная регистрация маршрута	2	2	Н/И
Частичная маршрутизация со стороны отправителя	3	3	Н/И
Частичная регистрация маршрута	3	3	Н/И

Продолжение табл. 4

Функция	Полный протокол	Несегментируемое подмножество	Инертное подмножество
Приоритет	3	3	Н/И
Обеспечение КУ	3	3	Н/И
Информирование о перегрузке	3	3	Н/И
Заполнение	3	3	Н/И

Примечания:

1. Н/И — не используется.

2. Если функции «информирование об ошибке» и «обнаружение ошибок в заголовке» должны обеспечиваться, то они привлекаются только при их выборе логическим-объектом-сетевого-уровня-отправителем.

3. Причина введения функций типа 3 состоит в том, что иногда более важно продвинуть ПБД между промежуточными системами или доставить их оконечной системе, чем обеспечить некоторые функции. Функции типа 3 должны использоваться в тех случаях, когда они имеют консультативный характер; эти функции не могут быть причиной аннулирования ПБД, если они не обеспечиваются.

7. СТРУКТУРА И КОДИРОВАНИЕ ПБД

7.1. Структура

Все протокольные блоки данных должны содержать целое число октетов. Нумерация октетов ПБД начинается с единицы (1) и возрастает в порядке поступления октетов к услугам нижерасположенных уровней. Биты в октете нумеруются от единицы (1) до восьми (8), где бит единица (1) — бит младшего порядка (наименьшей значимости).

Если для представления двоичного номера используется последовательность октетов, то октет с меньшим номером имеет наибольшую значимость.

В спецификации любой реализации, обеспечивающей этот протокол, должен быть указан с использованием понятия «бит старшей значимости» и «бит младшей значимости» способ передачи октетов. ПБД этого протокола определяются с использованием понятий «бит старшей значимости» и «бит младшей значимости».

Примечание. Если кодирование ПБД осуществляется с использованием диаграмм этого раздела, то применимо следующее правило:

а) младший по номеру октет располагается слева с возрастанием вправо номеров октетов;

б) в пределах октета бит восемь (8) размещается слева, а бит единица (1) — справа.

ПБД должны содержать следующие элементы в перечисляемой последовательности:

а) фиксированную часть;

б) адресную часть;

в) сегментируемую часть (при ее наличии);

г) факультативную часть (при ее наличии);

д) часть «данные» (при ее наличии).

Эта структура ПБД показана в табл. 5.

Таблица 5

Составная часть ПБД	Пункт, в котором приведено описание ПБД
Фиксированная часть	7.2
Адресная часть	7.3
Сегментируемая часть	7.4
Факультативная часть	7.5
Часть «данные»	7.6

7.2. Фиксированная часть

7.2.1. Общие понятия

Фиксированная часть имеет формат, показанный в табл. 6.

Таблица 6

Фиксированная часть заголовка ПБД	Номер октета
Идентификатор протокола сетевого уровня	1
Указатель длины	2
Версия/расширение идентификатора протокола	3
Время существования	4
ФСР ФДС ФИО Тип	5
Длина сегмента	6,7
Контрольная сумма	8,9

7.2.2. Идентификатор протокола сетевого уровня

Значение этого поля устанавливается в двоичном коде 1000 0001 для идентификации данного протокола сетевого уровня

как протокола настоящего стандарта. Это поле устанавливается в двоичное значение 0000 0000 для идентификации инертного подмножества протокола сетевого уровня.

7.2.3. Указатель длины

Длина обозначается двоичным числом с максимальным значением 254 (1111 1110). Указываемая длина — это длина, выраженная в октетах заголовка, см. п. 7.1. Значение 255 (1111 1111) зарезервировано для возможного в будущем расширения.

Примечание. Правила продвижения и сегментирования гарантируют одинаковую длину заголовка для всех сегментов (производных ПБД) исходного ПБД, равную длине заголовка исходного ПБД. Длина заголовка ПБД не должна изменяться под действием любой функции протокола.

7.2.4. Версия / расширение идентификатора протокола

Значение этого поля выражается в двоичном коде 0000 0001 и идентифицирует стандартную версию 1 настоящего протокола.

7.2.5. Время существования ПБД

Поле «время существования ПБД» кодируется двоичным числом, которое показывает оставшееся время существования ПБД в единицах по 500 мс.

7.2.6. Флаги

7.2.6.1. Сегментирование разрешено

Флаг «сегментирование разрешено» показывает, что сегментирование разрешается. Его значение определяется отправителем ПБД и не может быть изменено никаким другим логическим-объектом-сетевого-уровня в течение времени существования исходного ПБД и любых других производных ПБД.

Значение единица (1) указывает, что сегментирование разрешено. Значение нуль (0) указывает, что сегментирование не разрешено. Если выбрано значение нуль, то сегментируемая часть заголовка ПБД отсутствует и значение поля «длина сегмента» определяет общую длину ПБД (см. пп. 7.2.8 и 7.4.3).

7.2.6.2. Дополнительные сегменты

Флаг «дополнительные сегменты» определяет, является ли часть «данные» этого ПБД последним октетом данных пользователя в СБДС (в виде своего последнего октета). Если флаг «дополнительные сегменты» установлен в единицу (1), то происходит сегментирование и последний октет СБДС не содержится в этом ПБД. Флаг «дополнительные сегменты» не может устанавливаться в единицу (1), если флаг «сегментирование разрешено» не установлен в единицу (1).

Если флаг «дополнительные сегменты» установлен в нуль (0),

то последний октет части «данные» ПВД является последним октетом СБДС.

7.2.6.3. Информирование об ошибке

Если флаг «информирование об ошибке» установлен в единицу (1), то правила п. 6.10 используются для определения необходимости генерации ПБД «информирование об ошибке», если нужно аннулировать этот ПБД «данные».

Если флаг «информирование об ошибке» установлен в ноль (0), то аннулирование ПБД «данные» не будет вызывать генерацию ПБД «информирование об ошибке».

7.2.7. Код типа

Поле «код типа» идентифицирует тип протокольного блока данных. Допустимые значения приведены в табл. 7.

Таблица 7

Тип ПБД	Код типа ПБД Биты: 5 4 3 2 1
ПБД ДН	1 1 1 0 0
ПБД ОШ	0 0 0 0 1

7.2.8. Длина сегмента ПБД

Поле «длина сегмента» обозначает полную длину ПБД в октетах, включая заголовок и данные (при их наличии). Если используется полный протокол и ПБД не сегментируется, то значение этого поля идентично значению поля «общая длина», расположенному в сегментируемой части данного заголовка.

При использовании несегментируемого подмножества протокола сегментируемая часть в заголовке отсутствует. В этом случае поле «длина сегмента» определяет полную длину исходного ПБД, включая одновременно заголовок и данные (при их наличии).

Поле «длина сегмента» не изменяется в течение всего времени существования ПБД.

7.2.9. Контрольная сумма ПБД

Контрольная сумма вычисляется на основе полного заголовка ПБД. В ПБД «данные» она учитывает сегментируемую и факультативную части (при их наличии). В ПБД «информирование об ошибке» учитывается также поле «причина аннулирования».

Значение контрольной суммы, равное 0, введено для того, чтобы показать, что контрольная сумма должна игнорироваться. Действие функции «обнаружение ошибок в заголовке ПБД» (см. п. 6.11) обеспечивает, что значение нуль не даст правильной конт-

рольной суммы. Ненулевое значение указывает, что контрольная сумма должна обрабатываться.

7.3. Адресная часть

7.3.1. Общие положения

Адресная часть располагается непосредственно за фиксированной частью заголовка ПБД. Адресная часть показана в табл. 8.

Таблица 8

Адресная часть заголовка ПБД	Номер октета
Указатель длины адреса получателя	10
Адрес получателя	11 . . . $m-1$
Указатель длины адреса отправителя	m
Адрес отправителя	$m+1$ . . . $n-1$

7.3.1.1. Адреса получателя и отправителя

Адреса получателя и отправителя, используемые данным протоколом, представляют собой адреса-пункта-доступа-к-услугам-сетевого-уровня, определенные в ИСО 8348/Доп. 2.

Адреса получателя и отправителя имеют переменную длину. Поля адресов получателя и отправителя кодируются в виде адресной информации протокола сетевого уровня, использующей предпочтительное/двоичное кодирование, определенное в ИСО 8348/Доп. 2.

Поле «указатель длины адреса получателя» определяет длину адреса получателя в октетах. Поле «адрес получателя» расположено за полем «указатель длины адреса получателя».

Поле «указатель длины адреса отправителя» определяет длину адреса отправителя в октетах и следует за полем «адрес получателя». Поле «адрес отправителя» следует за полем «указатель длины адреса отправителя».

Каждый параметр адреса кодируется, как показано в табл. 9.

Таблица 9

Параметр адреса	Номер октета
Указатель длины параметра адреса (например, m)	n
Значение параметра адреса	$n+1$ до $n+m$

7.4. Сегментируемая часть

Если флаг «сегментирование разрешено» в фиксированной части заголовка ПБД (см. п. 7.2.6.1) установлен в единицу, то сегментируемая часть заголовка, показанная в табл. 10, должна присутствовать.

Таблица 10

Сегментируемая часть заголовка ПБД	Номер октета
Идентификатор блока данных	n, $n+1$
Смещение сегмента	$n+2$, $n+3$
Общая длина	$n+4$, $n+5$

Если флаг ,сегментирование разрешено» установлен в нуль, то сегментируемая часть отсутствует (используется несегментируемое подмножество протокола).

7.4.1. Идентификатор блока данных

Идентификатор блока данных идентифицирует исходный ПБД (и, следовательно, его производные ПБД) так, чтобы сегментируемый блок данных мог быть собран правильно. Длина идентификатора блока данных равна двум октетам.

7.4.2. Смещение сегмента

Для каждого производного ПБД поле «смещение сегмента» определяет соответствующее положение сегмента, находящегося в части «данные» производного ПБД, относительно начала части «данные» исходного ПБД. Смещение выражается в октетах. Смещение первого сегмента (и, следовательно, исходного ПБД) равно нулю; несегментируемый (исходный) ПБД имеет значение смещения сегмента, равное нулю (0). Значение этого поля должно быть кратно восьми (8).

7.4.3. Общая длина ПБД

Поле «общая длина» определяет полную длину исходного ПБД в октетах, включая заголовок и данные. Это поле не изменяется в течение всего времени существования исходного ПБД (и, следовательно, его производных ПБД)

7.5. Факультативная часть

7.5.1. Общие положения

Факультативная часть заголовка ПБД показана в табл. 11.

Таблица 11

Факультативная часть заголовка ПБД	Номер октета
Факультативные возможности	$+$ p

При наличии факультативной части она может содержать от одного до нескольких параметров. Число параметров, которые могут содержаться в факультативной части, ограничивается длиной факультативной части, определяемой по следующей формуле:

Длина факультативной части равна длине заголовка ПБД (длина фиксированной части плюс длина адресной части плюс длина сегментируемой части), а также длиной отдельных факультативных параметров.

Параметры, определенные в факультативной части, могут быть представлены в любой последовательности. Не допускается дублирования факультативных возможностей. Прием протокольного блока данных с продублированной факультативной частью должен рассматриваться как протокольная ошибка. Правила, относящиеся к обработке протокольных ошибок, описываются в п. 6.10 «Функция «информирование об ошибке».

Кодирование параметров, содержащихся в факультативной части заголовка ПБД, показано в табл. 12.

Таблица 12

Параметр факультативной части заголовка ПБД	Номер октета
Код параметра	n
Длина параметра (например m)	$n+1$
Значение параметра	$n+2$ до $n+m+1$

Поле *код параметра* кодируется двоичным числом и обеспечивает максимум 255 различных параметров. Ни один код параметра не использует биты 8 и 7 в значении 00, поэтому фактическое максимальное число параметров меньше. Код параметра 255 (в двоичном коде 1111 1111) зарезервирован для возможных в будущем расширений.

Поле *длина параметра* показывает длину поля *значение параметра*, выраженную в октетах. Длина выражается положительным двоичным числом *m*, теоретическое максимальное значение которого 254. Практическое максимальное значение *m* меньше. Например, в случае одного параметра, содержащегося внутри факультативной части, необходимы два октета для кода параметра и указателей длины параметра. Таким образом, значение *m* ограничивается значением 252 — (длина фиксированной части плюс длина адресной части плюс длина сегментируемой части).

Соответственно для каждого последующего параметра максимальное значение *m* уменьшается.

Поле «значение параметра» содержит значение параметра, идентифицируемое в поле «код параметра».

В факультативной части допускаются следующие параметры.

7.5.2. Заполнение

Параметр «заполнение» используется для удлинения заголовка ПБД до нужного размера (см. п. 6.12).

Код параметра: 1100 1100.

Длина параметра: переменная.

Значение параметра: допускается любое значение.

7.5.3. Защита

Этот параметр предусматривает единственный и однозначный уровень защиты, назначаемый протокольному блоку данных (см. п. 6.13).

Код параметра: 1100 0101.

Длина параметра: переменная.

Значение параметра: два бита высшего порядка первого октета.

Код формата защиты определяют в соответствии с табл. 13.

Остальная часть первого октета зарезервирована и должна быть установлена в нуль. Оставшаяся часть поля «значение параметра» определяет уровень защиты, описываемый в пп. 7.5.3.1—7.5.3.3.

7.5.3.1. Специфичный адрес отправителя

Значение кода формата защиты в двоичном выражении 01 означает, что остальные октеты поля «значение параметра» определяют уровень защиты, единственный и однозначный в контексте

системы классификации защиты, применяемой уполномоченным за назначением адреса ПДУСУ — отправителя.

Таблица 13

Код формата защиты	Тип поля „ защита"
00	Зарезервировано
01	Специфичный адрес отправителя
10	Специфичный адрес получателя
11	Глобальная уникальная

7.5.3.2. Специфичный адрес получателя

Значение кода формата защиты в двоичном выражении 10 означает, что остальные октеты поля «значение параметра» определяют уровень защиты — единственный и однозначный в контексте системы классификации защиты, применяемой уполномоченным за назначение адреса ПДУСУ — отправителя.

7.5.3.3. Глобальная уникальная защита

Значение кода формата защиты в двоичном выражении 11 означает, что остальные октеты поля «значение параметра» определяют глобальный уникальный и однозначный уровень защиты. Эта система классификации защиты не определена в настоящем стандарте.

7.5.4. Маршрутизация со стороны отправителя

Параметр «маршрутизация со стороны отправителя» определяет либо полный, либо частичный маршрут, который должен быть пройден от адреса отправителя до адреса получателя сетевого уровня (см. п. 6.14).

Код параметра: 1100 1000.

Длина параметра: переменная.

Значение параметра: два октета управляющей информации, за которыми следует цепочка записей наименований логических-объектов-сетевого-уровня в последовательности от отправителя до получателя.

Первый октет поля «значение параметра» является кодом типа и имеет следующее значение:

0000 0000 — частичная маршрутизация со стороны отправителя;

0000 0001 — полная маршрутизация со стороны отправителя
<все остальные значения зарезервированы>.

Второй октет означает смещение октета следующего в списке подлежащего обработке наименования логического-объекта-сетевого-уровня. Он относится к началу параметра, так что значение три (3) означает, что следующая запись наименования логических-объектов-сетевого-уровня начинается сразу после этого управляющего октета. Последующие октеты обозначаются соответственно большими значениями этого указателя.

Третий октет начинает список наименований логических-объектов-сетевого-уровня. Этот список состоит из наименований логических-объектов-сетевого-уровня переменной длины. Первый октет каждой записи указывает длину наименования логического-объекта-сетевого-уровня, которое образует остальную часть записи.

7.5.5. Регистрация маршрута

Параметр «регистрация маршрута» идентифицирует промежуточные системы, пройденные ПБД (см. п. 6.15).

Код параметра: 1100 1011.

Длина параметра: переменная.

Значение параметра: два октета управляющей информации, за которыми следует цепочка записей наименования логического-объекта-сетевого-уровня в последовательности от отправителя к получателю.

Первый октет поля «значение параметра» содержит код типа и имеет следующее значение:

0000 0000 — действует частичная регистрация маршрута;
0000 0001 — действует полная регистрация маршрута
<все остальные значения зарезервированы>.

Второй октет идентифицирует первый октет, не используемый в данный момент для зарегистрированного наименования логического-объекта-сетевого-уровня и, следовательно, идентифицирует конец списка. Он кодируется относительно начала значения параметра так, что значение параметра три (3) означает, что ни одного наименования логического-объекта-сетевого-уровня еще не зарегистрировано. Значение, выраженное одними единицами, используется для указания завершения регистрации маршрута.

Третий октет начинает список наименований логических-объектов-сетевого-уровня. Этот список состоит из записей наименований логических-объектов-сетевого-уровня переменной длины. Первый октет каждой записи определяет длину наименования логического-объекта-сетевого-уровня, которое содержится в остальной части записи. Записи наименований логических-объектов-сетевого-уровня всегда добавляются в конец списка.

Примечание. Длина параметра «регистрация маршрута» определяется отправителем ПБД и не изменяется в течение всего времени существования ПБД, поэтому действие функции «регистрация маршрута» не влияет на длину заголовка.

7.5.6. Обеспечение качества услуг

Параметр «обеспечение качества услуг» передает информацию относительно качества услуг, запрошенных пользователем услуг сетевого уровня со стороны отправителя.

Логические объекты сетевого уровня в промежуточных системах могут (но не обязательно) использовать эту информацию как вспомогательное средство выбора маршрута при наличии нескольких маршрутов, удовлетворяющих другим критериям маршрутизации, и если известно, что имеющиеся маршруты различаются качеством услуг (см. п. 6.16).

Код параметра: 1100 0011.

Длина параметра: переменная.

Значение параметра: два старших бита первого октета определяют код формата КУ (см. табл. 14).

Таблица 14

Код формата КУ	Поле, тип КУ
00	Зарезервировано
01	Специфичный адрес отправителя
10	Специфичный адрес получателя
11	Глобальное уникальное

Остальные биты первого октета зарезервированы для использования глобальным уникальным форматом КУ, описанным в п. 7.5.6.3. Если выбран любой другой код формата КУ, то биты 6—1 первого октета должны быть равны нулю. Остальная часть поля «значение параметра» определяет КУ в соответствии с изложенным в последующих разделах.

7.5.6.1. Специфичный адрес отправителя

Значение кода формата КУ в двоичном выражении 01 означает, что остальные октеты поля «значение параметра» определяют КУ, которое является уникальным и недвусмысленным в контексте системы обеспечения КУ, применяемой уполномоченным за назначение адресов ПДУСУ — отправителя.

7.5.6.2. Специфичный адрес получателя

Значение кода формата КУ в двоичном выражении 10 означает, что остальные октеты поля «значение параметра» определя-

ют КУ, которое является уникальным и недвусмысленным в контексте системы обеспечения КУ, применяемой уполномоченным за назначение адресов ПДУСУ — получателя.

7.5.6.3. Глобальное уникальное КУ

Значение кода формата КУ в двоичном выражении 11, означает, что остальная часть поля «значение параметра» определяет поле «обеспечение глобального уникального КУ». При использовании функции «обеспечение глобального уникального КУ» поле «значение параметра» должно иметь общую длину, равную одному октету, которому присвоены значения, приведенные в табл. 15.

Таблица 15

Биты	Назначение
8 и 7	Код формата КУ, двоичное 11
6	Зарезервировано
5	Упорядочение относительно транзитной задержки
4	Наличие перегрузки
3	Транзитная задержка относительно стоимости
2	Вероятность необнаруженных ошибок относительно транзитной задержки
1	Вероятность необнаруженных ошибок относительно стоимости

Бит 5 устанавливается в единицу для указания на то, что решения о маршрутизации должны по возможности отдавать предпочтение передаче всех ПБД в адрес конкретного ПДУСУ по одному маршруту (для того чтобы сохранять последовательность) по отношению к минимизации транзитной задержки. Значение нуль (0) означает, что решения о маршрутизации должны по возможности отдавать предпочтение низкой транзитной задержке по отношению к сохранению последовательности.

Бит 4 устанавливается в ноль логическим-объектом-сетевого-уровня, передающим ПБД. Он устанавливается промежуточной системой в единицу для того, чтобы показать, что этот ПБД поступил в перегруженную промежуточную систему и логический-объект-сетевого-уровня — получатель должен выполнить соответствующее действие. Как только бит наличия перегрузки будет установлен промежуточной системой, он не может быть сброшен никакой другой промежуточной системой, которую далее проходит ПБД по пути к пункту назначения.

Бит 3 устанавливается в единицу для указания на то, что принимаемые решения о маршрутизации должны по возможности

отдавать предпочтение снижению транзитной задержки по отношению к минимизации стоимости. Значение 0 показывает, что принимаемые решения должны отдавать предпочтение низкой стоимости по отношению к малой транзитной задержке.

Бит 2 устанавливается в единицу для указания на то, что принимаемые решения о маршрутизации должны по возможности отдавать предпочтение низкой вероятности необнаруженных ошибок относительно малой транзитной задержки. Значение нуль показывает, что принимаемые решения о маршрутизации должны отдавать предпочтение снижению транзитной задержки по отношению к снижению вероятности необнаруженных ошибок.

Бит 1 устанавливается в единицу для указания на то, что принимаемые решения о маршрутизации должны по возможности отдавать предпочтение уменьшению вероятности необнаруженных ошибок по отношению к снижению стоимости. Значение нуль указывает, что решения о маршрутизации должны отдавать предпочтение снижению стоимости по отношению к уменьшению вероятности необнаруженных ошибок.

7.5.7. Приоритет

Значение параметра «приоритет» указывает соответствующий приоритет ПБД. Промежуточные системы, обеспечивающие эту факультативную возможность, должны использовать эту информацию при маршрутизации и упорядочении ПБД для передачи (см. п. 6.17).

Код параметра: 1100 1101.

Длина параметра: 1 октет.

Значение параметра:

от 0000 0000 — нормальное (рекомендуемое)

до 0000 1110 — наивысшее

<все остальные значения зарезервированы>.

Значения от 0000 0001 до 00001110 должны использоваться для ПБД с более высоким значением приоритета. Если промежуточная система не обеспечивает эту факультативную возможность, то все ПБД должны обрабатываться так, как если бы это поле имело значение 0000 0000.

7.6. Часть «данные»

Часть «данные» ПБД состоит из упорядоченного множества октетов, идентичного такому же упорядоченному множеству октетов, определенному для параметра «УСУ-данные-пользователя» примитивов СУ-БЛОК-ДАННЫХ. запрос и индикация. Часть «данные» показана в табл. 16.

Таблица 16

Часть «данные» заголовка ПБД	Номер октета
Данные	$p+1$ . . . z

7.7. ПБД «данные» (ДН)

7.7.1. Структура

ПБД ДН имеет формат, показанный в табл. 17.

7.7.1.1. Фиксированная часть

1). Идентификатор протокола сетевого уровня — по п. 7.2.2.
2). Указатель длины — по п. 7.2.3.
3). Версия/расширение идентификатора протокола — по п. 7.2.4.
4). Время существования — по п. 7.2.5.
5). ФСР, ФДС, ФИО — по п. 7.2.6.
6). Код типа — по п. 7.2.7.
7). Длина сегмента — по п. 7.2.8.
8). Контрольная сумма — по п. 7.2.9.

7.7.1.2. Адреса

См. п. 7.3.

7.7.1.3. Сегментирование

См. п. 7.4.

7.7.1.4. Факультативные возможности

См. п. 7.5.

7.7.1.5. Данные

См. п. 7.6.

7.8. Инертный протокол сетевого уровня

Формат инертного протокола сетевого уровня приведен в табл. 18.

7.8.1. Идентификатор протокола сетевого уровня

Значение поля «идентификатор протокола сетевого уровня» выражается в двоичном коде одними нулями (0000 0000).

7.8.2. Часть «данные»

Часть «данные» может содержать любое число октетов вплоть до числа на единицу меньшего максимального, которое может быть размещено в параметре «ПСТ-данные-пользователя» нижерасположенного примитива ПСТ-БЛОК-ДАННЫХ. Следовательно, инертный протокол сетевого уровня может быть использован только в том случае, если длина параметра «УСУ-данные-пользователя» в примитиве СУ-БЛОК-ДАННЫХ ограничена числом,

меньшим или равным значению длины параметра «ПСТ-данные-пользователя» минус единица (см. п. 7.6).

Таблица 17

ПБД - данные	Номер октета
Идентификатор протокола сетевого уровня	1
Указатель длины	2
Версия/расширение идентификатора протокола	3
Время существования	4
ФСР ФДС ФИО Тип	5
Длина сегмента	6,7
Контрольная сумма	8,9
Указатель длины адреса получателя	10
Адрес получателя	11 $m-1$
Указатель длины адреса отправителя	m
Адрес отправителя	$m+1$ $n-1$
Идентификатор блока данных	n, $n+1$
Смещение сегмента	$n+2$, $n+3$
Общая длина	$n+4$, $n+5$
Факультативные возможности	$n+6$ p
Данные	$p+1$ z

Таблица 18

Инертный протокол сетевого уровня	Номер октета
Идентификатор протокола сетевого уровня	1
Данные	2 . . . 2—n

7.9. ПБД «информирование об ошибке» (ОШ)

7.9.1. Структура

Формат ПБД «информирование об ошибке» показан в табл. 19.

7.9.1.1. Фиксированная часть

Фиксированная часть ПБД «информирование об ошибке» формируется таким же образом, как и новый (исходный) ПБД «данные». В табл. 20. даны ссылки на предыдущие разделы, где описывается кодирование полей, образующих фиксированную часть.

7.9.1.2. Адреса

Адрес получателя определяет наименование логического-объекта-сетевого-уровня — отправителя аннулированного ПБД. Адрес отправителя определяет наименование логического-объекта-сетевого-уровня промежуточной или оконечной системы, инициирующего ПБД «информирование об ошибке» (см. п. 7.3).

7.9.1.3. Факультативные возможности

См. п. 7.5.

7.9.1.4. Причина аннулирования

Этот параметр действителен только для ПБД «информирование об ошибке».

Код параметра: 1100 0001.

Длина параметра: два октета.

Значение параметра: тип ошибки, представленный в двоичном коде.

Значения данного параметра приведены в табл. 21.

Первый октет значения параметра содержит код типа ошибки. Если ошибка в аннулируемом ПБД «данные» может быть локализована в конкретном поле, то номер первого октета этого поля содержится во втором октете поля параметра «причина аннулирования». Если ошибка не может быть локализована в конкретном поле или ошибка обнаружена в контрольной сумме, то во втором октете поля параметра «причина аннулирования» содержится значение нуль.

Таблица 19

ПБД «информирование об ошибке»	Номер октета
Идентификатор протокола сетевого уровня	1
Указатель длины	2
Версия/расширение идентификатора протокола	3
Время существования	4
ФСР=0 ФДС=0 Зарезервировано Тип	5
Длина сегмента	6, 7
Контрольная сумма	8, 9
Указатель длины адреса получателя	10
Адрес получателя	11 . . . $m-1$
Указатель длины адреса отправителя	m
Адрес отправителя	$m+1$ . . . $n-1$
Факультативные возможности	n . . . $p-1$
Причина аннулирования	p . . . $q-1$
Часть «данные» ПБД «информирование об ошибке»	q . . . z

Таблица 20

ПБД» «информирование об ошибке. Поле фиксированной части

Поле	Пункт
Идентификатор протокола сетевого уровня	7.2.2
Указатель длины	7.2.3
Версия/расширение идентификатора протокола	7.2.4
Время существования	7.2.5
ФСР, ФДС, ФИО *(всегда установлены в нуль)*	6.10
Код типа	7.2.7
Длина сегмента	7.2.8
Контрольная сумма	7.2.9

Таблица 21

Значения параметра «причина аннулирования»

Значение параметра	Класс ошибки	Содержание
0000 0000	Общий	Причина не определена
0001		Протокольная процедурная ошибка
0010		Неправильная контрольная сумма
0011		ПБД аннулирован из-за перегрузки
0100		Синтаксическая ошибка заголовка (не поддается грамматическому разбору)
0101		Сегментирование необходимо, но не разрешено
0110		Принят неполный ПБД
0111		Продублированная факультативная возможность
1000 0000	Адрес	Адрес получателя недоступен
0001		Адрес получателя неизвестен
1001 0000	Маршрутизация со стороны отправителя	Не определена ошибка маршрутизации со стороны отправителя
0001		Синтаксическая ошибка в поле маршрутизации со стороны отправителя
0010		Неизвестен адрес в поле маршрутизации со стороны отправителя
0011		Маршрут неприемлем
1010 0000	Время существования	Истекло время существования в процессе передачи блока данных
0001		Истекло время существования во время сборки

Продолжение

Значение параметра	Класс ошибки	Содержание
1011 0000	ПБД аннулирован	Не определена необеспеченная факультативная возможность
0001		Не обеспечена версия протокола
0010		Не обеспечена факультативная защита
0011		Не обеспечена факультативная маршрутизация со стороны отправителя
0100		Не обеспечена регистрация факультативного маршрута
1100 0000	Сборка	Вмешательство в сборку

7.9.1.5. Часть «данные» ПБД «информирование об ошибке»

Это поле содержит весь заголовок аннулированного ПБД «данные» и может содержать либо полностью всю часть «данные» аннулированного ПБД, либо частично, либо вообще ничего не содержать из этой части.

8. ОБЕСПЕЧЕНИЕ УСЛУГ НИЖЕРАСПОЛОЖЕННОГО УРОВНЯ

Функции сходимости, зависимые от подсети, могут выполняться для обеспечения услуг нижерасположенного уровня в режиме без-установления-соединения в тех случаях, когда реальная подсеть по существу не обеспечивает тех услуг нижерасположенного уровня в режиме без-установления-соединения, на которые рассчитывает данный протокол. Если подсеть обеспечивает свойственные ей услуги в режиме с-установлением-соединения, то функция сходимости, зависимая от подсети, обеспечивает их преобразование в требуемые услуги нижерасположенного уровня в режиме без-установления-соединения. Функции сходимости, зависимые от подсети могут потребоваться также в тех случаях, когда не выполняются функции, ожидаемые от услуг нижерасположенного уровня. В некоторых случаях такая ситуация может потребовать, чтобы определенный протокол (т. е. протокол, охватывающий явные обмены протокольной управляющей информацией между равноправными логическими-объектами-сетевого-уровня) действовал в роли протокола сходимости, зависимого от подсети (ПСЗП). Однако могут быть и такие случаи, когда набор функций, необходимых для выполнения роли ПСЗП, состоит просто из набора правил манипуляций услугами нижерасположенного уровня (без обмена протокольной управляющей информацией между равноправными логическими-объектами-сетевого-уровня).

8.1. Пункты подключения к подсети

Параметры адреса отправителя и получателя в примитиве ПСТ-БЛОК-ДАННЫХ определяют пункты подключения к подсети (ям) общего или частного пользования. Адреса пунктов подключения к подсети (ППП) определяются уполномоченным каждой отдельной подсети. Синтаксис и семантики ППП не определяются настоящим стандартом.

8.2. Качество услуг подсети

Для каждой передачи в режиме без-установления-соединения во время инициации действий примитива ПСТ-БЛОК-ДАННЫХ запрашиваются некоторые меры качества услуг. Эти запрашиваемые меры (или значения параметров и факультативные возможности) основываются на априорных сведениях об услугах, доступных из подсети. Сведения о характере и типе доступной услуги обычно приобретаются до привлечения услуг в режиме без-установления-соединения.

Параметры «качество услуг», идентифицированные для услуг нижерасположенного уровня в режиме без-установления-соединения, в некоторых случаях могут быть получены непосредственно из тех параметров или преобразованы в те параметры, которые идентифицированы в услугах сетевого уровня в режиме без-установления-соединения. В частности, могут использоваться следующие параметры, определенные в ГОСТ 34.951—92:

а) транзитная задержка;

б) защита от несанкционированного доступа;

в) определители стоимости;

г) вероятность необнаруженных ошибок.

Примечание. Для тех реальных подсетей, которым не свойственно обеспечивать качество услуг в виде параметра, вопрос сохранения семантики запрашиваемой услуги является частным решением. В частности, возможны случаи, когда запрошенное качество услуг не может быть обеспечено. В таких случаях должна предприниматься доставка ПБД с любым доступным качеством услуг.

В общем случае либо ФСЗП, либо сама подсеть может выполнять функции, связанные с конкретными запросами КУ. Эти функции могут факультативно выбираться протоколом ПСУ-БУС. Соответствующие параметры КУ подсети классифицируются следующим образом:

а) параметры КУ, при которых ФСЗП или сама подсеть выполняет специально разработанные функции с целью обеспечения информации для функции «маршрутизация ПБД» ПСУ-БУС;

б) параметры КУ, при которых ФСЗП или сама подсеть выполняет специально разработанные функции для обеспечения желаемого КУ;

в) параметры КУ, при которых ФСЗП или сама подсеть может быть привлечена для выполнения любой из двух вышеуказанных функций а) или б).

Определение значений этих параметров КУ содержится в пп. 8.2.1—8.2.4.

8.2.1. Транзитная задержка

Транзитная задержка — это промежуток времени между выполнением примитива ПСТ-БЛОК-ДАННЫХ. запрос и соответствующего примитива ПСТ-БЛОК-ДАННЫХ. индикация. Значения этого промежутка времени вычисляются на основе успешно переданных СБДП. Передача СБДП считается успешной в том случае, если СБДП, выданный передающей ФСЗП, доставлен адресуемой ФСЗП. Расчет транзитной задержки основывается на длине СБДП 512 октетов и определяется в единицах 500 мс.

Транзитная задержка определяется ФСЗП до обработки подсетью любых данных пользователя. Механизм, с помощью которого информация о транзитной задержке передается функции «маршрутизация ПБД» протокола ПСУ-БУС, определяется как частное решение. Транзитная задержка может быть либо измерена, либо оценена. Описываемые здесь ФСЗП не обеспечивают никаких средств измерения или оценки транзитной задержки помимо средств, обеспечиваемых нижерасположенной подсетью.

Примечания:

1. При необходимости измерения транзитной задержки протокол ПСЗП, ориентированный на ограничение времени прохождения блоков СБДП через подсеть, должен использоваться до обработки любых запросов данных с целью определения фактической задержки.

Транзитная задержка в пределах конкретной подсети может изменяться. Если транзитная задержка подлежит измерению, то для сохранения точности измерений при любой информации маршрутизации, обеспечиваемой логическим-объектом-сетевого-уровня, может оказаться необходимым периодически повторять измерения транзитной задержки.

2. При отсутствии лучших способов измерения транзитная задержка может быть оценена путем передачи СБДП (с помощью некоторого однозначно идентифицируемого ПБД, запрашивающего ответ) и измерения промежутка времени между выполнением примитива ПСТ-БЛОК-ДАННЫХ. запрос и соответствующего примитива ПСТ-БЛОК-ДАННЫХ.индикация. Это приводит к такой завышенной оценке задержки, что можно ожидать правильной работы ПСУ-БУС. Если транзитная задержка оценивается, то желательно, чтобы результаты оценки были завышенными, а не заниженными с тем, чтобы неточности в ее оценке не помешали ПСУ-БУС аннулировать те ПБД, у которых истекло заданное время существования.

3. Те подсети, которые используют протокол ГОСТ 34.950, способны динамически определять, будет ли транзитная задержка, доступная при установлении соединения, соответствовать запрашиваемому значению транзитной задержки. Для обеспечения этой динамической возможности функция ФСЗП может использовать выбор/индикацию транзитной задержки и средства согласования

межконцевой транзитной задержки X.25 ППУ. Согласование транзитной задержки функцией ФСЗП или автоматом ПДП прозрачно для автомата ПСУ-БУС.

4. В подсетях, использующих протокол ГОСТ 34.950, значение транзитной задержки, обеспечиваемое ПСУ-БУС, должно учитывать любую задержку обработки или ожидания в очереди, возникающую в результате попыток ФСЗП установить виртуальный канал.

8.2.2. Защита от несанкционированного доступа

В настоящее время не существует рекомендации, касающейся способа защиты от пассивного наблюдения, модификации, замены, добавления или вычеркивания ПСТ-данных-пользователя.

8.2.3. Вероятность необнаруженных ошибок

Вероятность необнаруженных ошибок вычисляется как отношение потерянных, продублированных или неправильно доставленных СБДП к общему числу СБДП, переданных ФСЗП за период измерения. Механизм, с помощью которого значение вероятности необнаруженных ошибок сообщается функции «маршрутизация ПБД» ПСУ-БУС, является частным вопросом.

Вероятность необнаруженных ошибок становится известной для ФСЗП до обработки любых данных пользователя подсетью либо на основе хранимой в ФСЗП последовательности измерений вероятности необнаруженных ошибок, либо в результате получения информации от поставщика нижерасположенных услуг.

Примечание. Для подсетей, обеспечивающих услуги в режиме с-установлением-соединения, вероятность необнаруженных ошибок определяется для каждого отдельного соединения.

8.2.4. Определители стоимости

Попытка соблюсти ограничения, налагаемые пользователем УСУ, с помощью параметра качества услуг *определители стоимости* осуществляется функцией «маршрутизация ПБД», привлекаемой протоколом ПСУ-БУС. В приемлемых случаях информация, относящаяся к тарифу(ам), оцениваемым числом переданных пакетов или числом используемых соединений, передается функцией «маршрутизация ПБД» протокола ПСУ-БУС. Механизм реализации этого процесса является частным вопросом.

Примечание. Функция «маршрутизация ПБД», привлекаемая протоколом ПСУ-БУС, потребуется для выполнения следующих стоимостных оценок, если:

а) при обработке рассматриваемого СБДП не происходит увеличения стоимости и тарифы устанавливаются в зависимости от числа переданных пакетов;

б) не происходит увеличения стоимости и к конкретному адресату нет доступного соединения, а тариф определяется подсетью в зависимости от числа используемых соединений (например для установленного виртуального канала, времени удержания виртуального канала и т. д.);

в) максимально приемлемая стоимость была определена для процесса обработки СБДС и она вероятно завышена, то функция «маршрутизация ПБД» должна выдать результат, показывающий, что протокол ПСУ-БУС должен пытаться

доставить СБДС по некоторому альтернативному маршруту. Если альтернативный маршрут нельзя отыскать, то может быть привлечена локальная функция для уведомления пользователя УСУ о неспособности поставщика УСУ доставить данный СБДС (и, возможно, последующие СБДС) при существующих ограничениях.

8.3. Данные пользователя подсети

Параметр «ПСТ-данные-пользователя» представляют собой упорядоченный набор октетов и передается в «прозрачном» режиме между определенными пунктами подключения подсети.

Нижерасположенные услуги, на которые ориентируется ПСУ-БУС, необходимы для обеспечения СБД длиной не менее 512 октетов.

Если известно, что минимальные длины СБД, обеспечиваемые всеми подсетями, участвующими в передаче конкретного ПБД, достаточно велики для того, чтобы не требовалось сегментирования, то может использоваться несегментируемое подмножество протокола.

Данные, полученные из подсети, у которой идентификатор протокола соответствует настоящему протоколу (первый октет равен 1000 0001) должны обрабатываться в соответствии с настоящим стандартом.

Примечание. Данные с другим идентификатором протокола должны игнорироваться, поскольку они могут быть переданы изделием, обеспечивающим дополнительные протоколы, предназначенные для использования совместно с настоящим стандартом.

8.4. Функции сходимости, зависимые от подсети

8.4.1. Общая модель

Общая модель обеспечения услуг нижерасположенного уровня, на которые рассчитывает протокол, в сочетании с реальной подсетью, использующей протокол доступа к подсети без-установления-соединения, выглядит следующим образом. Выработка примитива ПСТ-БЛОК-ДАННЫХ.запрос протоколом ПСУ-БУС приводит к выработке функцией сходимости, зависимой от подсети соответствующего специфичного для подсети примитива БЛОК-ДАННЫХ.запрос. Прием специфичного для подсети примитива БЛОК-ДАННЫХ.индикация, связанного с передачей блока данных без-установления-соединения к его получателю, побуждает ФСЗП сгенерировать для ПСУ-БУС примитив ПСТ-БЛОК-ДАННЫХ.индикация.

Общая модель обеспечения услуг нижерасположенного уровня, на которые рассчитывает ПСУ-БУС, в сочетании с реальной подсетью, использующей протокол доступа к подсети в режиме соединения, выглядит следующим образом. Выработка примитива ПСТ-БЛОК-ДАННЫХ.запрос протоколом ПСУ-БУС делает доступным

соединение (логический канал, логическое звено или их эквивалент) для передачи параметра «ПСТ-данные-пользователя». Если соединение не может быть доступным, то примитив ПСТ-БЛОК-ДАННЫХ.запрос аннулируется. Прием специфичного для подсети ПБД, содержащего «ПСТ-данные-пользователя», побуждает ФСЗП сгенерировать для ПСУ-БУС примитив ПСТ-БЛОК-ДАННЫХ.индикация.

В тех случаях, когда реальная подсеть ориентирована на использование протокола доступа к подсети либо в режиме без-установления- соединения, либо в режиме с-установлением-соединения, обеспечение услуг нижерасположенного уровня, на которые рассчитывает ПСУ-БУС, достигается путем использования альтернативного режима без-установления-соединения.

8.4.2. Функции сходимости, зависимые от подсети, используемые подсетями типа ЛВС ГОСТ 28907

В ГОСТ 28907 определены два класса управления логическим звеном (УЛЗ). Класс 1 обеспечивает только услуги режима без-установления-соединения. Класс 2 обеспечивает услуги как режима без-установления-соединения, так и режима с-установлением-соединения. Для станций, соответствующих любому из этих двух классов услуг, используется услуга режима без-установления-соединения и без подтверждения с целью обеспечения услуг нижерасположенного уровня, на которые рассчитывает настоящий протокол.

Услуги в режиме без-установления-соединения без подтверждения, описываемые в ГОСТ 28907 — это как раз то, что требует ПСУ-БУС. Параметры такой услуги, за исключением КУ, приведены в табл. 22.

Таблица 22

Примитивы услуг подуровня УЛЗ

Примитивы	Параметры
ЗД-БЛОК-ДАННЫХ .запрос индикация	ЗД-адрес-отправителя, ЗД-адрес-получателя; ЗД-приоритет, ЗД-данные

Функции сходимости, зависимые от подсети, выполняют преобразование услуг режима без-установления-соединения и без подтверждения, обеспечиваемых подсетью класса 1 или 2 УЛЗ, в услуги нижерасположенного уровня, на которые рассчитывает

ПСУ-БУС. Преобразование происходит следующим образом. Генерация примитива ПСТ-БЛОК-ДАННЫХ.запрос протоколом ПСУ-БУС приводит к генерации примитива ЗД-БЛОК-ДАННЫХ.запрос функцией сходимости, зависимой от подсети. Соответствующий примитив ЗД-БЛОК-ДАННЫХ.индикация побуждает ФСЗП выработать примитив ПСТ-БЛОК-ДАННЫХ.индикация для ПСУ-БУС. Для обеспечения такого преобразования услуг никакого обмена явной управляющей информации протокола ПСЗП между логическими-объектами-сетевого-уровня не происходит.

Адреса, используемые в примитивах ПСТ-БЛОК-ДАННЫХ.запрос и индикация, представляют собой 7-октетные адреса станции ЛВС, описанные в ГОСТ 28907, и состоящие из 6-октетного адреса подуровня управления доступом к физической среде (УДС) плюс одноокетный адрес ПДУ подуровня УЛЗ.

Примечание. Для того, чтобы обеспечить услуги нижерасположенного уровня, на которые рассчитывает настоящий протокол, эти услуги должны обеспечивать минимальную длину сервисного блока данных 512 октетов. Если протокол ГОСТ 28907 не налагает никаких ограничений на длину СБД, то минимальное требование к УДС состоит в том, чтобы он мог передавать ПБД «ненумерованная информация» (НИ), содержащие информационное поле длиной 128 октетов. Поэтому в таких случаях на ФСЗП налагается дополнительное ограничение, обусловленное необходимостью передачи в ПБД НИ, по меньшей мере, 512 октетов данных пользователя.

8.4.3. Функции сходимости, зависимые от подсети, используемые подсетями ГОСТ 34.950

Услуги режима с-установлением-соединения, которые обеспечиваются подсетями, использующими протокол пакетного уровня Х.25, определенный в ГОСТ 34.950, обрабатываются функцией ФСЗП и таким образом виртуальный канал становится доступным для передачи параметра «ПСТ-данные-пользователя» вслед за генерацией примитива ПСТ-БЛОК-ДАННЫХ.запрос протоколом ПСУ-БУС. В общем случае (см. п. 8.4.3.4) для обеспечения такого преобразования услуг не происходит никаких обменов явной управляющей информацией протокола ПСЗП между равноправными логическими-объектами-сетевого-уровня на протяжении действия фазы данных.

Параметры «ПСТ-адрес-получателя» и «ПСТ-адрес-отправителя» в примитивах ПСТ-БЛОК-ДАННЫХ.запрос и индикация являются адресами ООД Х.121, используемыми подсетью Х.25.

Если подсеть Х.25 не обеспечивает информации вызывающего ООД, то в примитиве ПСТ-БЛОК-ДАННЫХ.индикация содержится нулевой параметр «ПСТ-адрес-отправителя». ФСЗП должна помещать адрес собственного ООД в поле «вызывающее ООД» па-

кета X.25 «запрос вызова» в том случае, когда сама подсеть не включает этот параметр, но разрешает ООД включать его.

Примечание. Некоторые подсети, использующие ППУ X.25, реализуют другие схемы адресации, отличающиеся от X.121. Использование таких схем адресации (например соответствующих рекомендациям Е.163 и Е.164 МККТТ) не запрещается.

Параметр «ПСТ-данные-пользователя» содержит данные пользователя, максимальная длина которых определена администратором подсети. Нижерасположенные услуги, на которые рассчитывает настоящий протокол, требуют, чтобы подсеть могла обеспечивать максимальную длину сервисного блока данных 512 октетов.

Примечание. Бит М может использоваться в тех случаях, когда подсеть X.25 не может непосредственно обеспечить минимальную длину пакета в 512 октетов, а также в таких ситуациях, когда длина СБД превышает требуемую минимальную длину (например при использовании несегментируемого подмножества протокола).

8.4.3.1. Вопросы установления соединения

Механизм открытия виртуального канала до передачи параметра «ПСТ-данные-пользователя» и синхронизация этого процесса представляют собой частный вопрос. Открытие виртуального канала может инициироваться при:

а) получении СБДП, подлежащего передаче через подсеть X.25 в момент отсутствия доступного виртуального канала;

б) наличии локальной очереди вопросов, ожидающей, когда существующий виртуальный канал достигнет предельного размера, при котором необходимо обеспечить доступность дополнительного виртуального канала, если возможно, для поддержания запрашиваемого КУ;

в) явном вмешательстве со стороны управления системой.

Если определено, что необходимо обеспечить новый виртуальный канал, то вызывающая ФСЗП выполняет все функции, связанные с установлением виртуального канала. Вызываемая ФСЗП выполняет действия, связанные с приемом вызова, но не генерирует примитив ПСТ-БЛОК-ДАННЫХ.индикация до тех пор, пока не будет завершено установление соединения, когда может осуществляться обмен пакетом (ами) «данные» X.25. В общем случае прием пакетов X.25 «данные», содержащих «ПСТ-данные-пользователя», побуждает ФСЗП сгенерировать примитив ПСТ-БЛОК-ДАННЫЕ. индикация для протокола ПСУ-БУС. Пакеты X.25 «повторная установка» при их приеме не влияют на действие ФСЗП. К необходимым процедурам, выполняемым для правильной работы ППУ X.25, относятся следующие.

8.4.3.2. Вопросы завершения соединения

Выбор механизмов определения момента завершения виртуального канала после передачи параметра «ПСТ-данные-пользователя» функцией ФСЗП является частным вопросом. Примерами ситуаций, которые могут завершить виртуальный канал, являются:

а) истечение тайм-аута после передачи одного или нескольких ПБД (см. п. 8.4.3.4);

б) необходимость использования специального интерфейса для открытия альтернативного виртуального канала от локального логического-объекта-сетевого-уровня к другому удаленному логическому-объекту-сетевого-уровня;

в) явное вмешательство со стороны управления системой;

г) завершение виртуального канала по инициативе поставщика.

Если определено, что виртуальный канал должен быть завершен, то ФСЗП выполняет все функции, относящиеся к завершению соединения. Игнорируются все пакеты, за исключением пакета «подтверждение завершения» или «индикация завершения». Те же действия предпринимаются при приеме пакета «индикация завершения». В таких случаях ФСЗП будет сохранять данные пользователя, переданные с примитивами ПСТ-БЛОК-ДАННЫХ.запрос при попытках установить новый канал; однако ФСЗП должна аннулировать данные пользователя, если существует вероятность превышения транзитной задержки, сообщаемой протоколу ПСУ-БУС.

Примечание. Для правильной работы, описываемой здесь ФСЗП, не требуется, чтобы виртуальные каналы динамически открывались или закрывались. Не запрещается использовать постоянные виртуальные каналы (ПВК) или удерживать виртуальные каналы в открытом состоянии, начиная с инициации системы.

8.4.3.3. Различимость протоколов.

Первый октет поля «данные вызывающего пользователя» пакета «запрос вызова» должен устанавливаться в значение, показывающее, что виртуальный канал должен использоваться для обеспечения услуг нижерасположенного уровня, на которые рассчитывает настоящий протокол. Это значение определяется в п. 7.2.2.

8.4.3.4. Длительности тайм-аутов

Тайм-ауты могут использоваться для определения момента освобождения виртуального канала (например, если виртуальный канал длительное время оставался незанятым) или момента открытия дополнительных виртуальных каналов (например при наличии чрезмерно большей очереди блоков данных, ожидающих исходного логического канала).

В некоторых реализациях может быть отдано предпочтение освобождению виртуального канала после некоторого периода его незанятости. Если для этой цели используется тайм-аут, то он при-

меняется следующим образом. Если виртуальный канал доступен для передачи СБДП, то отсчет тайм-аута начинается со значения, представляющего собой максимальный период времени, в течение которого виртуальный канал может оставаться незанятым. Каждый раз, когда блок данных передается услугой нижерасположенного уровня, тайм-аут устанавливается в это исходное значение. Если в очереди на обработку блоков данных нет и этот тайм-аут истек, то виртуальный канал освобождается.

Выбор значений тайм-аута является частным вопросом.

Примечания:

1. При наличии очень большой очереди блоков данных, ожидающих исходного логического канала, могут быть открыты дополнительные виртуальные каналы. Длительности тайм-аутов для определения времени завершения таких дополнительных виртуальных каналов могут быть короче длительностей подобных тайм-аутов для исходного виртуального канала. Длительность тайм-аутов может быть также постоянной величиной. В конкретных реализациях может быть отдано предпочтение закрытию всех дополнительных виртуальных каналов, если очередь подлежащих передаче блоков данных достигнет некоторого предельного значения (возможно нулевого).

2. Длительности тайм-аутов выбирают на основе критериев экономичности и особенностей конкретной реализации. Если уполномоченный данной подсети не налагает никаких тарифов за длительность пребывания виртуального канала в открытом состоянии, а существуют тарифы за открытие виртуальных каналов, то длительность тайм-аута может выбираться такой, чтобы виртуальный канал оставался открытым на длительный период времени. Длительности тайм-аутов могут изменяться также в зависимости от времени суток, нагрузки трафика, усредненной за последний период, или других факторов.

8.4.3.5. Разрешение конфликтов из-за пользования виртуальным каналом

Две ФСЗП могут одновременно попытаться установить виртуальные каналы друг с другом. Желательно иметь возможность обнаруживать такие ситуации и исключать один виртуальный канал, сохраняя другой во избежание излишних затрат на установление соединения.

Если подсеть обеспечивает адрес вызывающего ООД, то такой конфликт можно обнаружить. Конфликт происходит, когда от ООД поступает входящий вызов в то время, как ожидается подтверждение на ранее инициированное соединение с этим же ООД.

Если адрес вызывающего ООД не обеспечивается сетью, то конфликты не обнаруживаются.

Решение вопроса, какой из виртуальных каналов должен быть сохранен при появлении конфликта, принимается на основе соглашения. Соглашение основывается на сравнении адресов вызываемого и вызывающего ООД X.25. Виртуальный канал, инициируе-

мый ФСЗП и имеющий наибольший адрес ООД, является тем каналом, который сохраняется.

При приеме пакета X.25 «запрос вызова» и неполучении подтверждения на предыдущий пакет «запрос вызова», выданный по тому же адресу ООД, ФСЗП должна выполнить процедуру разрешения конфликта вызовов, которая описывается как последовательность нижеперечисленных шагов:

а) адрес ООД локальной ФСЗП следует сравнить с адресом ООД удаленной ФСЗП. Если адреса имеют различную длину, то меньший из них дополняется нулями до длины большего с наиболее значащего (левого) конца адреса;

б) сравнение должно выполняться, начиная с цифры наименьшей значимости (правой), и продолжаться до цифры наибольшей значимости (левой);

в) как только обнаруживается, что цифры, имеющие одинаковую позицию в каждом адресе, имеют различные значения, сравнение прекращается;

г) адрес, представленный числом с меньшим значением (где 0 — наименьшее, а 9 — наибольшее), рассматривается как младший адрес;

д) если локальная ФСЗП имеет меньший адрес, то она должна освободить инициированный ею виртуальный канал и воспринять виртуальный канал, инициированный удаленной ФСЗП;

е) если локальная ФСЗП имеет больший адрес, она должна освободить виртуальный канал, инициированный удаленной ФСЗП, и продолжать ожидать принятия инициированного ею виртуального канала.

Если запрос на установление нового виртуального канала принимается при наличии установленного виртуального канала, то новый виртуальный канал должен быть воспринят, а существовавший до этого — освобожден.

Примечание. Эта процедура необходима для обеспечения быстрого восстановления в ситуации инициируемого поставщиком завершения виртуального канала, когда обе ФСЗП не получают уведомления об этом действии в одно и то же время.

8.4.3.6. Использование группы виртуальных каналов

В некоторых ситуациях желательно использовать несколько виртуальных каналов X.25 между двумя логическими-объектами-сетевого-уровня, например для того, чтобы увеличить пропускную способность или устойчивость. В этом случае каждый виртуальный

канал отдельно наблюдаем со стороны ПСУ-БУС и обеспечивает четко различимые услуги. Каждый канал обеспечивается отдельной парой независимых ФСЗП. Необходимо уметь различать эти независимые виртуальные каналы во избежание неправильного обнаружения конфликтов.

Там, где требуется группа виртуальных каналов, различимость между ними обеспечивается во время установления соединения с помощью передачи двухоктетного указателя подключения к подсети (УПП) в поле «данные пользователя» пакета X.25 «запрос вызова». Если в пакете X.25 «индикация вызова» данные пользователя отсутствуют (кроме протокольного дискриминатора, закодированного в октете 1), то принимающая ФСЗП должна действовать так, как если бы УПП был явно передан с нулевым значением. При необходимости передать указатель подключения к подсети поле «данные пользователя» пакета X.25 «запрос вызова» должно устанавливаться в соответствии с табл. 23.

Таблица 23

Кодирование указателя подключения к подсети

Различимость протоколов	Указатель длины	Версия УПП	Значение УПП
Октет 1	Октет 2	Октет 3	Октеты 4 и 5
0000 0000 или 1000 0001 (см. п. 7.2.2)	0000 0100	0000 0010	(см. ниже)

Примечание. Описанные выше процедуры были определены для удовлетворения следующих критериев:

а) непредвиденные продублированные виртуальные каналы должны быстро обнаруживаться и освобождаться;

б) должна быть обеспечена возможность иметь группу виртуальных каналов между двумя логическими-объектами-сетевого-уровня при наличии необходимости, например по соображениям пропускной способности или избыточности и

в) в общем случае, когда требуется только один виртуальный канал, должна требоваться только минимальная протокольная управляющая информация (предпочтительно нулевая).

Октеты 1—3 устанавливаются в указанные значения. Октеты 4 и 5 содержат указатель подключения к подсети. Октет 4 содержит младший разряд октета УПП, октет 5 — старший разряд.

Процедура разрешения конфликта, рассмотренная в п. 8.4.3.5, должна выполняться только в том случае, если два виртуальных канала переносят (явно или неявно) один и тот же УПП.

Значения УПП могут выбираться взаимодействующими ФСЗП произвольно. Если требуется определенное число виртуальных

каналов, но нет предварительного соглашения относительно используемых значений ПСЗП, то должны использоваться значения от нуля до значения, на единицу меньшего числа требуемых виртуальных каналов.

8.4.3.7. Приоритет

Как часть своих операций по управлению виртуальными каналами ФСЗП может выполнять функцию назначения приоритетов относительно примитивов ПСТ-БЛОК-ДАННЫХ.запрос, чтобы определить приоритет в виде параметра КУ. В частности, ФСЗП может открыть новый виртуальный канал для обслуживания трафика с более высоким приоритетом или закрыть существующий виртуальный канал для освобождения логического канала или ресурсов локальной системы, предоставив их для обработки трафика с более высоким приоритетом при отсутствии для него других ресурсов.

8.4.3.8. Элементы протокола ГОСТ 34.950

Перечисленные ниже элементы протокола ГОСТ 34.950 необходимы для обеспечения нижерасположенных услуг, на которые рассчитывает настоящий протокол:

а) услуги виртуального соединения;
б) передача данных (без бита подтверждения доставки и процедур прерывания передачи);
в) процедуры управления потоком;
г) пакеты «управление потоком» и «повторная установка»;
д) пакеты «установление соединения» и «завершение соединения»;
е) пакеты «данные ООД и АКД»;
ж) процедуры повторного пуска;
з) пакеты повторного пуска;
и) тайм-ауты АКД;
к) временные ограничители ООД;
л) кодирование пакетов, генерируемых сетью X.25.

Нижеперечисленные элементы протокола являются желательными, но необязательными:

м) согласование параметра управления потоком;
н) выбор и индикация транзитной задержки;
о) согласование класса пропускной способности.

Все другие услуги и средства являются факультативными.

Примечание. Обязательные элементы протокола не мешают работе ФСЗП через подсеть, использующей версию X.25 1980 г.

9. СООТВЕТСТВИЕ

Для соответствия настоящему стандарту требуется способность инициировать, манипулировать и принимать ПБД в соответствии

с полным протоколом (в отличие от несегментируемого подмножества или подмножества «инертный протокол сетевого уровня»).

Кроме того, соответствие настоящему стандарту требует обеспечения функций протокола, описанных в разд. 6. Обеспечение факультативных функций, описанных в п. 6.19 и перечисленных в табл. 23, должно удовлетворять требованиям, описанным там же. Ниже в п. 9.1 приведены исключения из этих требований.

Соответствие настоящему стандарту обязывает также придерживаться структуры и кодирования ПБД согласно разд. 7. Только если вышеперечисленные требования удовлетворены, имеет место соответствие настоящему стандарту.

9.1. Обеспечение функций соответствия

В табл. 24 функции разд. 6 подразделяются на категории по типам систем, обеспечивающих данную функцию.

Таблица 24

Функция	Тип системы		
	Передача	Продвижение	Прием
Формирование ПБД	О	(примечание 1)	(примечание 1)
Разборка ПБД	О	—	О
Анализ формата заголовка	—	О	О
Управление временем существования ПБД	—	О	Ф
Маршрутизация ПБД	—	О	—
Продвижение ПБД	О	О	(примечание 1)
Сегментирование ПБД	О	(примечание 2)	—
Сборка ПБД	—	Ф	О
Аннулирование ПБД		О	О
Информирование об ошибках	О	О	О
Обнаружение ошибок в заголовке	О	О	О
Защита	—	(примечание 4)	(примечание 4)
Полная маршрутизация со стороны отправителя	—	(примечание 5)	—
Полная регистрация маршрута	—	(примечание 5)	—
Частичная маршрутизация со стороны отправителя	—	(примечание 5)	—
Частичная регистрация маршрута	—	(примечание 5)	—
Приоритет	—	(примечание 5)	—
Обеспечение КУ	—	(примечание 5)	
Информирование о перегрузке	—	(примечание 5)	—
Заполнение	—	(примечание 3)	(примечание 3)

Условные обозначения:
«О» — обязательная функция;
«Ф» — факультативная реализация в соответствии с изложенным в тексте;
«—» — не используется.

Примечания:

1. Обеспечение функций «формирование ПБД» и «продвижение ПБД» необходимо для генерации ПБД «информирование об ошибках».

2. Функция «сегментирование ПБД» в общем случае является обязательной для промежуточной системы. Однако система, которая должна подключаться только к таким подсетям, каждая из которых обеспечивает одинаковую максимальную длину СБД (как например, идентичные локальные вычислительные сети), не будет нуждаться в выполнении этой функции и, следовательно, в ее реализации.

Если эта функция не реализована, она должна рассматриваться как часть спецификации реализации.

3. Правильная трактовка функции «заполнение» не требует операций обработки. Реализация, претендующая на соответствие стандарту, должна обеспечивать данную функцию в такой степени, чтобы игнорировать этот параметр, где бы он ни появился.

4. Эта функция может обеспечиваться или не обеспечиваться. Если реализация не обеспечивает эту функцию, а она выбрана в ПБД, то этот ПБД должен быть аннулирован, сгенерирован ПБД ОШ и направлен исходному логическому-объекту-сетевого-уровня, если флаг «информирование об ошибках» установлен и условия п. 6.10.4 соблюдены.

5. Эта функция может обеспечиваться или не обеспечиваться. Если реализация не обеспечивает эту функцию, а она выбрана в ПБД, то функция не выполняется и ПБД обрабатывается точно так, как если бы эта функция не была выбрана, ПБД не должен аннулироваться исходя из этой причины.

ПРИЛОЖЕНИЕ А
Справочное

ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ

А.1. Введение

Операции настоящего протокола (ПСУ-БУС) моделируются в виде конечного автомата, который действует под управлением переменной, принимающей три значения. Поведение автомата определяется относительно отдельных независимых ПБД. Переход состояния автомата происходит под действием элементарного события на одном из трех интерфейсов:

а) интерфейсе с транспортным уровнем, определенным сервисными примитивами ГОСТ 34.951;

б) интерфейсе с услугами нижерасположенного уровня, на которые рассчитывает протокол, определенный примитивом ПСТ-БЛОК-ДАННЫХ в п. 5.5 настоящего стандарта;

в) интерфейсе с зависимой от реализации функцией тайм-аута, определенной примитивами ТАЙМ-АУТ, описанными в п. 5.6 настоящего стандарта.

Кроме того, переходы состояний автомата могут быть обусловлены появлением внутренних условий автомата.

Элементарные события определены в п. А.3. Появление элементарного события само по себе недостаточно для того, чтобы вызвать переход; возможно потребуется выполнение других «разрешающих» условий, прежде чем сможет произойти данный переход. Разрешающие условия выражаются булевыми переменными, которые зависят от значений параметров, относящихся к соответствующему элементарному событию (т. е. параметров некоторого примитива), и от значений локально-обеспечиваемых переменных.

С одним элементарным событием может быть связано несколько разрешающих условий и, следовательно, несколько возможных переходов. В каждом таком случае разрешающие условия взаимно исключают друг друга, поэтому для любой конкретной комбинации элементарного события и значений параметра может произойти только один переход состояния.

Каждому переходу соответствует действие или «выход». Действия представляют собой изменения значений локальных переменных и последующее выполнение необходимых функций в количестве от нуля до нескольких. Работа конечного автомата полностью определена в п. А.4 путем определения действий, относящихся к каждому возможному переходу.

А.2. Значения переменной

Протокольная переменная имеет три значения:

исходное — автомат создается в исходном состоянии. Никакой переход не может установить автомат в исходное состояние;

сборка — автомат находится в состоянии сборка в течение всего периода времени, когда он осуществляет сборку сегментов ПБД в полный ПБД;

завершено — конечным состоянием автомата является состояние завершено. Когда автомат входит в состояние завершено, он прекращает свое существование.

А.3. Элементарные события

Элементарное событие состоит в передаче единицы информации через интерфейс. Описание элементарного события определяет примитив (например СУ-БЛОК-ДАННЫХ.запрос, описанный в ГОСТ 34.951) и границу услуг, на которой он привлекается (например граница услуг сетевого уровня). Направление

потока информации через эту границу вытекает из определения каждого из примитивов.

А.3.1. Примитивы СУ.БЛОК-ДАННЫХ ≈ запрос и индикация

Элементарные события СУ.БЛОК-ДАННЫХ_запрос и СУ.БЛОК-ДАННЫХ_индикация происходят на границе услуг сетевого уровня. Примитивы, описывающие эти элементарные события, определены в стандарте ГОСТ 34.951:

СУ.БЛОК-ДАННЫХ_запрос (УСУ-адрес-отправителя,
УСУ-адрес-получателя,
УСУ-качество-услуг,
УСУ-данные-пользователя);

СУ.БЛОК-ДАННЫХ_индикация (УСУ-адрес-отправителя,
УСУ-адрес-получателя,
УСУ-качество-услуг,
УСУ-данные-пользователя).

Параметры примитивов СУ.БЛОК-ДАННЫХ_запрос и СУ.БЛОК-ДАННЫХ_индикация в совокупности называются сервисными блоками данных сетевого уровня.

А.3.2. Примитивы ПСТ.БЛОК-ДАННЫХ_запрос и индикация

Элементарные события ПСТ.БЛОК-ДАННЫХ_запрос и ПСТ.БЛОК-ДАННЫХ индикация происходят на абстрактном интерфейсе, который предположительно существует между автоматом ПСУ-БУС и обеспечиваемой услугой нижерасположенного уровня. Примитивы, описывающие эти элементарные события, определены в п. 5.5:

ПСТ.БЛОК-ДАННЫХ_запрос (ПСТ-адрес-отправителя,
ПСТ-адрес-получателя,
ПСТ-качество-услуг,
ПСТ-данные-пользователя);

ПСТ.БЛОК-ДАННЫХ_индикация (ПСТ-адрес-отправителя,
ПСТ-адрес-получателя,
ПСТ-качество-услуг,
ПСТ-данные-пользователя).

Параметры примитивов ПСТ.БЛОК-ДАННЫХ_запрос и ПСТ.БЛОК-ДАННЫХ_индикация в совокупности называются сервисными блоками данных подсети.

Значение параметра «ПСТ-данные-пользователя» может представлять исходный или производный ПБД.

А.3.3. Элементарные события услуги СТ.ТАЙМ-АУТ

Элементарные события СТ.ТАЙМ-АУТ происходят на интерфейсе между рассматриваемым здесь протоколом и его локальной средой. Они определены в п. 5.6:

СТ.ТАЙМ-АУТ_запрос (время, имя, индекс);

СТ.ТАЙМ-АУТ_аннулирование (имя, индекс);

СТ.ТАЙМ-АУТ_ответ (имя, индекс).

А.4. Операции конечного автомата

В настоящем приложении в формализованном виде определяется абстрактный автомат, который обеспечивает простой пример услуг сетевого уровня в режиме без-установления-соединения путем использования ПСУ-БУС. Следует подчеркнуть, что такая формализованная спецификация никоим образом не ограничивает внутренние операции или структуру любой фактической реализации. Например, не требуется, чтобы отдельные сегменты программы, описываемые в переходах состояний, фактически выглядели как часть конкретной реализации. Такая формализованная спецификация полезна тем, что она пытается устранить любые неясности или двусмысленности, с которыми можно столкнуться при спецификации стандарта по протоколу.

Такая формализованная спецификация определяет поведение отдельного конечного автомата, который отражает поведение протокола, соответствующее отдельному независимому запросу услуги. Предполагается, что любая конкретная реализация будет в состоянии обеспечить поведение, соответствующее поведению нескольких одновременно работающих конечных автоматов.

Предполагается, что для этой спецификации сборка выполняется оконечной системой получателя.

Когда автомат ПСУ-БУС получает из транспортного уровня примитив СУ-БЛОК-ДАННЫХ запрос, то параметры, относящиеся к этому запросу, используются для формирования протокольного блока данных в соответствии с изложенным в разд. 6. Параметры «УСУ-адрес-получателя» и «УСУ-адрес-отправителя» примитива СУ-БЛОК-ДАННЫХ запрос используются для получения АИПСУ, подлежащей передаче в виде параметров «адрес ПДУСУ отправителя» и «адрес ПДУСУ получателя» данного ПБД. Параметр «УСУ-данные-пользователя» определяет подлежащие передаче данные пользователя. Остальные поля заголовка ПБД получаются из локальной информации, информация о состоянии и параметра «УСУ-качество-услуг».

Параметр «УСУ-качество-услуг» определяет КУ, запрошенное передающим пользователем УСУ. Вопрос о том, обеспечивается это КУ непосредственно самим ПСУ-БУС либо с помощью средств КУ, предлагаемых каждой услугой нижерасположенного уровня, подлежащей прохождению, решается до выдачи исходного примитива ПСТ-БЛОК-ДАННЫХ.запрос функцией «маршрутизация ПБД» автомата ПСУ-БУС. В это время определяется также, какие факультативные параметры (если они применяются) должны использоваться в помощь при обработке этого ПБД, а также решается вопрос о необходимости сегментирования ПБД.

Как только соответствующая подсеть выбрана, функция «продвижение ПБД» автомата ПСУ-БУС выдает примитив ПСТ-БЛОК-ДАННЫХ.запрос. Когда ФСЗП получателя (идентифицируемая пунктом ППП, указанным в параметре «ПСТ-адрес-получателя» примитива ПСТ-БЛОК-ДАННЫХ.запрос) принимает СБДП, она информирует обслуживаемый ею автомат ПСУ-БУС о доставке СБДП с помощью соответствующего примитива ПСТ-БЛОК-ДАННЫХ.индикация.

Анализируется заголовок ПБД и если адрес ПДУСУ, закодированный в поле «адрес получателя» ПБД, соответствует одному из адресов в ПДУСУ (если они имеются), обеспечиваемых автоматом ПСУ-БУС, то происходит разборка ПБД и выдается примитив СУ-БЛОК-ДАННЫХ индикация для пользователя-УСУ, подключенного к этому ПДУСУ. Если определено, что ПБД не достиг своего получателя, то выполняются функции «маршрутизация» и «продвижение ПБД». Этот процесс продолжается до тех пор, пока ПБД не поступит к адресату или пока не истечет время существования ПБД, в случае чего ПБД аннулируется.

А.4.1. Определения типов и констант

```
Const
    НУЛЬ = 0,
    макс_данные_пользователя = 64512;
    ЗАКОНЧЕНО = 255;
type
    тип_данных_тайм_аута = ...;
    тип_ид_протокола_сетевого_уровня = (ид_протокола_ИСО_8473);
    тип_ид_версии = (версия 1);
    тип_тп_пбд = (ДН, ОШ);
    тип_адр_ПДУСУ = ...;                (* адреса ПДУСУ, проходящие через границу услуг сетевого уровня *)
```

```
тип_адр_АИПСУ = ... ;          (* адреса, передаваемые в ПБД *)
тип_адр_ПСТ = ... ;            (* адреса в услугах нижерасположенного
                                  уровня, используемых данным протоко-
                                  лом *)
тип_наименований_рм = ... ;    (* содержит список адресов в параметре
                                  «регистрация маршрута» *)
тип_качества_услуги = ... ;    (* параметр КУ, передаваемый через
                                  границу услуг сетевого уровня *)
тип_КУ_ПСТ = ... ;             (* параметр КУ (если используется), пере-
                                  даваемый нижерасположенным услугам,
                                  используемым данным протоколом *)
тип_данных = ... ;             (* данные пользователя. Концептуально
                                  это равноценно последовательности двоич-
                                  ных бит переменной длины *)
тип_буфера = ... ;             (* ресурсы памяти, используемые при пе-
                                  редаче и приеме данных пользователя *)
тип_факультативных_воз-        (* запоминание факультативных частей
можностей = ... ;                 заголовка ПБД *)
тип_ид_подсети = ... ;         (* локальный идентификатор конкретной
                                  услуги нижерасположенного уровня, ис-
                                  пользуемой данным протоколом. В общем
                                  случае здесь возможны многие услуги ни-
                                  жерасположенной подсети (или звена дан-
                                  ных *)

тип_сбдс = record
  ап: тип_адр_ПДУСУ;
  ао: тип_адр_ПДУСУ;
  ку: тип_качества_услуг;
  данные: тип_данных;
  end;
тип_результата_маршрутизации = record
  ид_подсети: тип_ид_подсети;
  ап_пст: тип_адр_ПСТ;
  ао_пст: тип_адр_ПСТ;
  длина_сегмента: integer;
  end;
тип_ошибки = (ОТСУТСТВИЕ_ОШИБОК,
              СЛИШКОМ_МНОГО_ДАННЫХ_ПОЛЬЗОВАТЕЛЯ,
              ПРОТОКОЛЬНАЯ_ПРОЦЕДУРНАЯ_ОШИБКА,
              НЕПРАВИЛЬНАЯ_КОНТРОЛЬНАЯ_СУММА,
              ПЕРЕГРУЗКА,
              СИНТАКСИЧЕСКАЯ_ОШИБКА_ЗАГОЛОВКА,
              СЕГМ_НЕОБХОДИМА_И_НЕ_РАЗРЕШЕНА,
              НЕПОЛНЫЙ_ПБД,
              ДУБЛИРОВАННАЯ_ФАКУЛЬТАТИВНАЯ_ВОЗМОЖ-
              НОСТЬ,
              АДРЕСАТ_НЕДОСТУПЕН,
              АДРЕСАТ_НЕИЗВЕСТЕН,
              НЕОПРЕДЕЛЕННАЯ ОШИБКА_МАРШРУТИЗАЦИИ_СО_
              СТОРОНЫ_ОТПР,
              СИНТАКСИЧЕСКАЯ_ОШИБКА_В_ПОЛЕ_
              МАРШРУТИЗАЦИИ_СО_СТОРОНЫ_ОТПР,
              НЕИЗВЕСТЕН_АДРЕС_ПРИ_МАРШРУТИЗАЦИИ_СО_
              СТОРОНЫ_ОТПР,
```

НЕПРИЕМЛЕМ_ПУТЬ_МАРШРУТИЗАЦИИ_СО_
СТОРОНЫ_ОТПР,
ИСТЕКЛО_ВРЕМЯ_СУЩЕСТВОВАНИЯ_ПРИ_ПЕРЕДАЧЕ,
ИСТЕКЛО_ВРЕМЯ_СУЩЕСТВОВАНИЯ_ПРИ_СБОРКЕ,
НЕОПРЕДЕЛЕННАЯ_НЕОБЕСПЕЧЕННАЯ_
ФАКУЛЬТАТИВНАЯ_ВОЗМОЖНОСТЬ,
НЕОБЕСПЕЧЕННАЯ_ВЕРСИЯ_ПРОТОКОЛА,
НЕОБЕСПЕЧЕННАЯ_ФАКУЛЬТАТИВНАЯ_ЗАЩИТА,
НЕОБЕСПЕЧЕННАЯ_ФАКУЛЬТАТИВНАЯ_
МАРШРУТИЗАЦИЯ_СО_СТОРОНЫ_ОТПР,
НЕОБЕСПЕЧЕННАЯ_ФАКУЛЬТАТИВНАЯ_
РЕГИСТРАЦИЯ_МАРШРУТА,
ВМЕШАТЕЛЬСТВО_В_СБОРКУ);

```
тип_подробных_данных_рм = record
  дл_рм : integer;
  кодтипа_рм : тип_кодтипа_рм;
  смещение_рм : integer;
  наименование_рм : тип_наименований_рм;
  end;
тип_кодтипа_рм = (ЧАСТИЧНАЯ_РЕГИСТРАЦИЯ, ПОЛНАЯ РЕГИСТ-
РАЦИЯ);
тип_пбд = record
  ид_псу : тип_ид_протокола_сетевого_уровня;
  удз : integer;
  ид_версии : тип_ид_версии;
  время_существования : integer;
  ср : boolean;
  дс : boolean;
  флаг_ош : boolean;
  тп_пбд : тип_тп_пбд;
  дл_сегм : integer;
  контрольная_сумма : integer;
  дл_ап : integer;
  ап : тип_адр_АИПСУ;
  дл_ао : integer;
  ао : тип_адр_АИПСУ;
ид_бд : факультативная_возможность      (* имеется только если
integer;                                  ср = TRUE; *)
сс : факультативная_возможность         (* имеется только если
integer;                                  ср = TRUE; *)
общ_дл : факультативная_возмож-         (* имеется только если
ность integer;                            ср = TRUE; *)
факультативные_возможности : тип_факультативных_возможностей;
данные : тип_данных;
end;
```

А.4.2. Определение канала

```
channel пункт_доступа_к_сетевому_уровню  (пользователь, поставщик);
by пользователь:
    БЛОК_ДАННЫХ_запрос (УСУ_адрес_получателя : тип_адр_ПДУСУ;
                        УСУ_адрес_отправителя : тип_адр_ПДУСУ;
                        УСУ_качество_услуг : тип_качества_услуг;
                        УСУ_данные_пользователя : тип_данных);
by поставщик:
    БЛОК_ДАННЫХ_индикация (УСУ_адрес_получателя: тип_адр_ПДУСУ;
```

```
                                    УСУ_адр_отправителя: тип_адр_ПДУСУ;
                                    УСУ_качество_услуг: тип_качества_услуг;
                                    УСУ_данные_пользователя: тип_данных);
channel — пункт_доступа_к_подсети (пользователь, поставщик);
by пользователь:
    БЛОК_ДАННЫХ_запрос (ПСТ_адрес_получателя: тип_адр_ПСТ;
                        ПСТ_адрес_отправителя: тип_адр_ПСТ;
                        ПСТ_качество_услуг: тип_КУ_ПСТ;
                        ПСТ_данные_пользователя: тип_пбд);
by поставщик:
    БЛОК_ДАННЫХ_индикация (ПСТ_адрес_получателя: тип_адр_ПСТ;
                           ПСТ_адрес_отправителя: тип_адр_ПСТ;
                           ПСТ_качество_услуг: тип_КУ_ПСТ;
                           ПСТ_данные_пользователя: тип_пбд);
channel пункт_доступа_к_системе (пользователь, поставщик);
by пользователь:
    ТАЙМ-АУТ_запрос (время : integer;
                     ид_тайм-аута : тип_имени_тайм-аута;
                     индекс : integer);
    ТАЙМ-АУТ_аннулирование (ид_тайм-аута : тип_имени_тайм-аута);
                           индекс : integer);
by поставщик.
    ТАЙМ-АУТ_ответ (ид_тайм-аута : тип_имени_тайм-аута;
                    индекс : integer).
```

А.4.3. Определения из формализованного описания автомата

```
module протокольный_автомат_сетевого_уровня_в_режиме_БУС
    (СУ: пункт_доступа_к_сетевому_уровню (поставщик) common queue,
    ПСТ: array [тип_ид_подсети] of
            пункт_доступа_подсети (пользователь) common queue;
    СТ: пункт_доступа_к_системе (пользователь) individual queue);
body ИП for ПРОТОКОЛЬНОГО_АВТОМАТА_СЕТЕВОГО_УРОВНЯ_В_
РЕЖИМЕ_БУС;
var
    сбдс : тип_сбдс;
    ПБД : тип_пбд;
    буф_прм тип_буфера;
state : (ИСХОДНОЕ, СБОРКА, ЗАВЕРШЕНО);
function получить_время_существования (ап : тип_адр_ПДУСУ);
                                       ку : тип_качества_услуг)
                                       тип_время_существования;
primitive;
(* Данная функция выдает в ответ значение времени существования для использования в ПБД, основываясь на адресе получателя и запрошенном качестве услуг*).
function получить_сегм_разрешено (ап : тип_адр_ПДУСУ;
                                  ку : тип_качества_услуг;
                                  дд : integer): boolean;
primitive;
(* Данная функция выдает в ответ булево значение для использования в поле «сегментирование разрешено» ПБД. Это значение может зависеть от адреса получателя, запрошенного качества услуг и длины данных пользователя*).
function длина (данные : тип_данных) : integer;
primitive;
```

(* Данная функция выдает в ответ длину специфицированных данных в октетах*).
function длина_буф (буф : тип_буфера) : integer;
primitive;
(* Данная функция выдает в ответ длину данных в октетах, содержащихся в специфицированном буфере*).
function получить_флаг_ош (сбдс : тип_сбдс) : boolean;
primitive;
(* Данная функция выдает в ответ булево значение для использования в виде флага «информирование об ошибке» в ПБД, который передает специфицированный СБДС. Если ПБД подлежит аннулированию в некоторый последующий момент, то информирование об ошибке может передаваться обратно только в том случае, если булево значение равно TRUE*).
function получить_дл_АИПСУ (адр : тип_адр_ПДУСУ) : integer;
primitive;
(* Данная функция выдает в ответ длину адресной информации протокола сетевого уровня, соответствующую специфицированному адресу ПДУСУ*).
function получить_АИПСУ (адр : тип_адр_ПДУСУ) : тип_адр_АИПСУ;
primitive;
(* Данная функция выдает в ответ адреса, закодированные в заголовке протокола, или адресную информацию протокола сетевого уровня, соответствующую специфицированному адресу ПДУСУ*).
function получить_факультативные_возможности (ап : тип_адр_ПДУСУ;
ку : тип_качества_
услуг) : тип_факультативных_возможностей;
primitive;
(* Данная функция выдает в ответ поле «факультативные возможности» для ПБД, основываясь на специфицированном адресе и качестве услуг*).
получить_дл_заголовка (дл_ап : integer;
дл_ао : integer;
ср : boolean;
факультативные_возможности : тип_
факультативных_возможностей) : integer;
primitive;
(* Данная функция выдает в ответ длину заголовка в октетах. Длина заголовка зависит от длин адресов отправителя и получателя, наличия сегментируемой части заголовка и длины факультативной части*).
function получить_нд_блока_данных (ап : тип_адр_АИПСУ) : integer;
primitive;
(* Данная функция выдает в ответ идентификатор блока данных, который является уникальным для специфицированного адреса получателя*).
function организовать_буфер (данные : тип_данных) : тип_буфера;
primitive;
(* Данная функция помещает специфицированные данные во вновь организованный буфер. Способ управления буфером является частным вопросом. Вновь организованный буфер определяется в ответе как значение функции*).
function проверить_параметры (уда : integer;
ср : boolean;
ап : тип_адр_АИПСУ;
факультативные_возможности : тип_
факультативных_возможностей;
длина_данных : integer) : тип_ошибки;
primitive;
(* Данная функция определяет действительность параметров, связанных с мар-

шрутизацией и продвижением ПБД. Если они действительны, то в ответ выдается результат ОТСУТСТВИЕ_ОШИБОК и вызывается функция «маршрутизация» для определения маршрута и длины сегмента; в противном случае эта функция сообщает в ответ о наличии ошибки*).
function получить_контрольную_сумму (пбд : тип_пбд) : integer;
primitive;
(* Данная функция выдает в ответ 16-битовое целочисленное значение, которое должно быть помещено в поле «контрольная сумма» ПБД. Если средство «контрольная сумма» не выбрано, то функция выдает в ответ значение ноль*).
маршрутизация (удз : integer;
ср : boolean;
ап : тип_адр_АИПСУ;
факультативные_возможности : тип
факультативных_возможностей;
длина_данных : integer): тип_результата_маршрутизации;
primitive;
(* Данная функция определяет маршрут, по которому должен следовать сегмент ПБД, а также длину сегмента. Определение маршрута и длины сегмента может осуществляться взаимозависимо и производиться на основе длины заголовка, флага «сегментирование разрешено», адреса получателя, некоторых параметров (например, маршрутизация со стороны отправителя), содержащихся в факультативной части заголовка ПБД и длины данных. Функция «маршрутизация» выдает в ответ структуру данных, определяя подсеть, по которой должен передаваться сегмент, ППП отправителя и получателя, подлежащий использованию в подсети, и длину сегмента. Функция «маршрутизация» может быть привлечена только в том случае, если функция «проверка_параметров» уже определена и передаваемые параметры действительны*).
function извлечь (буф : тип_буфера; октеты : integer) : тип_данных;
primitive;
(* Данная функция выдает в ответ определенное число данных из определенного буфера. Число выдаваемых октетов ограничивается требованием, чтобы ни один сегмент, меньший восьми октетов, не мог быть генерирован передающим логическим-объектом-сетевого-уровня. Число выдаваемых октетов равно: 1) специфицированному числу; 2) числу данных, остающихся в буфере или 3) целому числу октетов, большему восьми, но не превышающему требования пп. 1) и 2). Выдаваемые в ответ данные удаляются из буфера и фактическое число октетов, извлекаемое из буфера, выдается в октетах параметра*).
function получить_ку_пст (ид_подсети : тип_ид_подсети;
факультативные_возможности : тип_факультативных
-возможностей): тип_КУ ПСТ;
primitive;
(* Данная функция выдает в ответ КУ, обеспечиваемое конкретной подсетью. Эта информация используется для определения значений параметров (если они используются) «обеспечение КУ», «защита» и «приоритет» в факультативной части ПБД*).
function получить_длину_локального_адр_АИПСУ : integer;
primitive;
(* Данная функция выдает в ответ длину локального адреса, используемую в заголовке ПБД
function получить_локальный_адр_АИПСУ : тип_адреса_АИПСУ;
primitive;
(* Данная функция выдает в ответ локальный адрес, используемый в заголовке протокола*).

```
function получить_адр_ПДУСУ (адрес : тип_адр_АИПСУ;
                             дл : integer) : тип_адр_ПДУСУ;
primitive;
(* Данная функция выдает в ответ адрес ПДУСУ, соответствующий адресной
информации протокола сетевого уровня определенной длины*).
function локальный_адр АИПСУ (адрес : тип_адр_АИПСУ) : boolean;
primitive;
(* Данная функция выдает в ответ булево значение TRUE только в том случае,
если определенный адрес ПДУСУ определяет локальный адрес*).
function получить_ку (факультативные_возможности : тип_факультативных_
                      возможностей): тип_качества_услуг
primitive;
(* Данная функция определяет в возможной степени качество услуг, достигну-
тое для конкретного ПБД, основываясь на качество услуг и другой информации,
содержащейся в факультативной части заголовка ПБД*).
function полный_блок_данных (буфер : тип_буфера) : boolean;
primitive;
(* Данная функция выдает в ответ булево значение, которое определяет, при-
няты ли все производные ПБД блока ПБД, содержащегося в определенном бу-
фере*).
function оценочное_истекшее_время : integer;
primitive;
(* Данная функция выдает в ответ оценочное значение истекшего времени, воз-
растающего по 500 мс, после того, как ПБД был передан предыдущим равно-
правным логическим-объектом-сетевого-уровня. Оценка включает в себя время,
затраченное на передачу, и все значения времени, затраченного в буферах внут-
ри локальной системы. Оценка не обязательно должна быть точной, но пред-
почтительна переоценка, чем недооценка, так как недооценка истекшего времени
может свести на нет действие функции «время существования*).
procedure очистить_буфер (буфер : тип_буфера);
primitive;
(* Данная процедура освобождает определенный буфер*)
procedure распределить_ресурсы_сборки (общ_дл_пбд : integer;
факультативные_возможности : тип_факультативных_возможностей);
primitive;
(* Данная процедура распределяет ресурсы, необходимые для сборки ПБД
конкретной общей длины. Вопрос о том, привлекать ли эту процедуру один раз
(при приеме первого производного ПБД или исходного ПБД) либо каждый раз
при приеме производного ПБД, является частным вопросом. Если ПБД должен
быть аннулирован для распределения ресурсов и в подлежащем аннулированию
ПБД флаг ОШ установлен, то генерируется информирование об ошибке. Реше-
ние об аннулировании ПБД и выбор ПБД для аннулирования может учиты-
вать значения параметра «приоритет» в поле ПБД «факультативные возмож-
ности»*).
procedure освободить_ресурсы_сборки;
primitive;
(* Данная процедура освобождает ресурсы, которые были предварительно рас-
пределены функцией «распределение_ресурсов_сборки»*).
procedure объединить_сегм (буфер : тип_буфера;
                           сс : integer;
                           данные : тип_данных);
primitive;
(* Данная процедура объединяет определенные данные в определенный буфер,
основываясь на определенном смещении сегмента данных*).
```

function передать_ош_при_перегрузке (пбд : тип_пбд) : boolean;
primitive;
(* Данная функция выдает в ответ булево значение TRUE, если должно быть передано информирование об ошибке, когда указанный блок данных аннулируется из-за перегрузки. Если в ответ выдается значение TRUE, то поле «флаг_ош» аннулированного блока данных должно по-прежнему проверяться до того, как будет передана информирование об ошибке*).
function получить_ош_время_существования (ап : тип_адр_АИПСУ) : integer;
primitive;
(* Данная функция выдает в ответ значение времени существования, которое должно использоваться в ПБД ОШ, основываясь на адресе получателя и определенном качестве услуг*).
function получить_ош_поля_данных (ошибка : тип_ошибки;
пбд : тип_цбд) : тип_данных;
primitive;
(* Данная функция выдает в ответ поле «данные» для ПБД «информирование об ошибке». Это поле «данные» должно включать заголовок аннулированного ПБД и может факультативно содержать дополнительные данные пользователя*).
function получить_ош_факультативных_возможностей (ошибка : тип_ошибки;
ап : тип_адр_АИПСУ;
факультативные_
возможности : тип_факультативных_возможностей): тип_факультативных_возможностей;
primitive;
(* Данная функция выдает в ответ поле «факультативные возможности» ПБД «информирование об ошибке», основываясь на причине аннулирования ПБД, адресе получателя и поле «факультативные возможности» аннулированного ПБД. Взаимосвязь факультативных возможностей ПБД «данные» с ПБД ОШ рассмотрена в п. 6.10.4. Поле «факультативные возможности» всегда содержит параметр «причина аннулирования», полученный из значения параметра «ошибка». Если факультативная возможность, выбранная в исходном ПБД, не может быть обеспечена системой, обрабатывающей ПБД «информирование об ошибке», то ошибка сообщается обратно в параметре «ошибка», не генерируется никаких ПБД ОШ, в противном случае значение ОТСУТСТВИЕ_ОШИБОК сообщается обратно в параметре «ошибка» и обработка ПБД ОШ продолжается. Выбор факультативных возможностей, отличных от тех, которые выбраны в ПБД, вызывающих генерацию ПБД ОШ, является частным вопросом*).
function получить_подробные_данные_рм (факультативные_возможности
: тип_факультативных_возможностей;
ошибка : тип_ошибки): тип_
подробных_данных_рм;
primitive;
(* Данная функция отыскивает с помощью факультативных возможностей параметр «регистрация маршрута». Если он не найден, то результирующее поле «дл_рм» устанавливается в нуль. В противном случае поле «дл_рм» устанавливается равным параметру «длина» и остальная часть результирующего поля устанавливается равной содержимому поля «значение параметра»*).
procedure послать_информирование_об_ошибке (ош_пбд : тип_пбд);
primitive;
(* Данная процедура посылает определенный тип ПБД «информирование об ошибке» (ОШ) локальному логическому объекту, который обрабатывает информацию об ошибке*).
procedure вставить_адрес (подробные_данные_м : тип_подробных_данных_рм);

```
                                    дл_поля : integer);
primitive;
(* Данная процедура обрабатывает параметр «регистрация маршрута», содер-
жащийся в подробных_данных_м. Сначала все наименования логических—объ-
ектов—сетевого—уровня перемещаются в списке вверх, оставляя пространство
в начале, длина которого равна значению «дл_поля». После этого происходит
вставка в это место поля нового наименования логического-объекта-сетевого-
уровня длиной один октет, содержащего длину наименования локального логи-
ческого-объекта-сетевого-уровня, за которым следует само наименование*);
procedure установить_подробные_данные_рм (факультативные_возможности:
                                    тип_факультативных_возможностей;
                                                подробные_данные_м
                                    тип_подробных_данных_рм);
primitive;
(* Данная процедура изменяет значение параметра «регистрация маршрута» в
поле «факультативные возможности»*).
procedure зарегистрировать_маршрут (факультативные_возможности : тип_
                                                факультативных_возможностей);
var
    подробные_данные_м : тип_подробных_данных_рм;
    длина_поля_наименования : integer;
begin
подробные_данные_м : = получить_подробные_данные_рм (факультативные_
                                                возможности);
if подробные_данные.длина_рм>0 and подробные_данные_м.смещение
    _рм<>ЗАКОНЧЕНО then
begin
    длина_поля_наименования := 1+(получить_дл_локального_адр_АИПСУ);
if ((подробные_данные_м.смещение_рм+длина_поля_адр) = подробные
    данные_м.длина_рм)>1
then подробные_данные_м.смещение_рм := ЗАКОНЧЕНО;
else
begin
    подробные_данные_м.смещение_рм :=подробные_данные_м.
    смещение_рм+длина_поля_наименования;
    вставить_адрес (подробные_данные_м, длина_поля_наименования);
end;
    установить_подробные_данные_рм (факультативные_возможности,
    подробные_данные_м);
      end;
    end;
procedure передать_информирование_об_ошибке (ошибка: тип_ошибки;
                                                пбд : тип_пбд);
var ош_пбд: тип_пбд;
begin
if (пбд.флаг_ош) then
begin
    ош_пбд.ид_всу := ид_настоящего_протокола;
    ош_пбд.ид_вер := версия 1;
    ош_пбд.время_существования := получить_ош_время_существования
                                                (пбд.ао);
    ош_пбд.ср := FALSE;
    ош_пбд.дс := FALSE;
    ош_пбд.флаг_ош := FALSE;
```

```
    ош_пбд.тп_пбд := ОШ;
    ош_пбд.дл_ап := пбд.дл_ао;
    ош_пбд_ап := пбд.ао;
    ош_пбд.дл_ао := получить_дл_локального_адр_АИПСУ;
    ош пбд.ао: = получить_локальный_адр_АИПСУ;
    ош_пбд.факультативные_возможности := получить_ош_факультативных_
                                  _возможностей (ошибка,
                                                 ош_пбд.ап, пбд.факультатив-
                                  ные_возможности);
if (ошибка = ОТСУТСТВИЕ_ОШИБОК) then
begin
    ош_пбд.удз := получить_дл_заголовка (ош_пбд.дл_ап,
                                         ош_пбд.дл_ао,
                                         ош_пбд.ср,
                                         ош_пбд.факультативные_воз-
                                         можности);
    ош_пбд.данные := получить_ош_поля_данных (ошибка, пбд);
if (локальный_адр_АИПСУ (ош.пбд.ап)) then
    послать_информирование_об_ошибке (ош.пбд)
else передать_пбд (ош_пбд);
    end;
  end;
end;
procedure передать_пбд (пбд : тип_пбд);
var
    результирующий_мрш : тип_результирующего_маршрута;
    код_ошибки : тип_ошибки;
    передать_буф : тип_буфера;
    извлекаемые_октеты : integer;
    дополнительные_сегм : boolean;
    ку_пст : тип_КУ_ПСТ;
begin
передать_буф := организовать_буфер (данные.пбд);
дополнительные_сегм := пбд.дс;
repeat
    begin
    код_ошибки := проверить_параметры (пбд. узд,
                                       пбд.ср,
                                       пбд.ап,
                                       пбд.факультативные_возможности,
                                       длина (пбд.данные));
if (код_ошибки = ОТСУТСТВИЕ_ОШИБОК) then
begin
    результирующий_мрш := маршрут (пбд.узд,
                                   пбд. ср,
                                   пбд.ап,
                                   пбд.факультативные_возможности,
                                   длина (пбд.данные));
    извлекаемые_октеты := длина_сегмента.результирующего_маршрута —
                          пбд. узз;
    данные.пбд := извлечь (передать_буф, извлекаемые_октеты);
    пбд.дл_сегм := пбд.узд+ длина (данные.пбд);
    if (длина_буфера (передать_буфер) = НУЛЬ) then
      пбд. дс := дополнительные_сегм;
else
```

```
    пбд.дс := TRUE;
    пбд.контрольная_сумма := получить_контрольную сумму (пбд);
    ку_пст := получить_ку_пст (результирующий_маршрут.ид_подсети,
                                пбд.факультативные_возможности);
    ВЫХОД ПСТ [результирующий_маршрут.ид_подсети], БЛОК_ДАННЫХ_
    _запрос
      (результирующий_мрш.ап_пст,
      результирующий_мрш.ао_пст,
      ку_пст,
      пбд);
if (пбд.ср) then пбд.ес := пбд. ес+извлекаемые_октеты,
  end
else
if (код_ошибки = ПЕРЕГРУЗКА) then
if (передать_ош_при_перегрузке (пбд)) then
    передать_информирование_об_ошибке (ПЕРЕГРУЗКА, пбд);
else
    передать_информирование_об_ошибке (код_ошибки, пбд);
end;
until (длина_буфера (данные_буфера) = НУЛЬ) or (код_ошибки <>
                                                ОТСУТСТВИЕ_ОШИБОК);

end;
initialize
begin
to ИСХОДНОЕ;
end;
(* Начать переходы*)
trans *переход #1*)
from ИСХОДНОЕ to ЗАВЕРШЕНО
when СУ.БЛОК_ДАННЫХ_запрос
provided not локальный_адр_ПДУСУ (УСУ_адрес_получателя)
begin
сбдс.ап := УСУ_адрес_получателя;
сбдс.ао := УСУ_адрес_отправителя;
сбдс.ку := УСУ_качество_услуг;
сбдс.данные := УСУ_данные_пользователя;
пбд.ид_псу := ид_настоящего_протокола;
пбд.ид_вер := версия 1;
пбд.время_существования := получить_время_существования (сбдс.ап,
                                сбдс. ку);
пбд.ср := получить_сегм_разрешено (сбдс.ап,
                                сбдс. ку,
                                длина (сбдс.данные));
пбд.дс := PALSE;
пбд.флаг_ош := получить_флаг_ош(сбдс);
пбд.тп_пбд := ДН;
пбд.дл_ап := получить_дл_АИПСУ (сбдс.ап);
пбд.ап := получить_АИПСУ (сбдс.ап);
пбд.дл_ао := получить_дл_АИПСУ (сбдс.ао);
пбд.ао := получить_АИПСУ (сбдс.ао);
пбд.факультативные_возможности := получить_факультативные_возможности
                                                (сбдс.ап, сбдс.ку);

пбд.данные := сбдс.данные;
пбд.удз := получить_длину_заголовка (пбд.дл_ап,
```

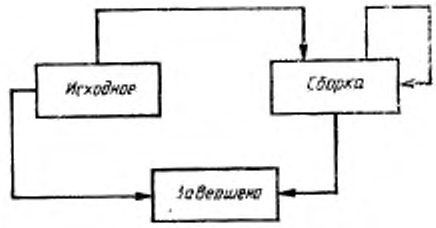
```
                                        пбд.дл_ао,
                                        пбд.ср,
                                        пбд.факультативные_возможности);
if (пбд.ср) then
begin
пбд.нд_бд := получить_нд_блока_данных (пбд.ап);
пбд.сс := НОЛЬ;
пбд.общ_длина := пбд.узд+длина (пбд.данные);
end;
if (длина (пбд.данные) > макс_данные_пользователя) then
    передать_информирование_об_ошибке (СЛИШКОМ_МНОГО_ДАННЫХ_
    ПОЛЬЗОВАТЕЛЯ, пбд)
else
    передать_пбд(пбд);
end;
                                                    (* переход # 2*)
from ИСХОДНОЕ
to ЗАВЕРШЕНО
cohen СУ.БЛОК_ДАННЫХ_запрос
provided локальный_адр_ПДУСУ (УСУ_адрес_получателя)
begin
сбдс.ап := УСУ_адрес_получателя;
сбдс.ао := УСУ_адрес_отправителя;
сбдс.ку := УСУ_качество_услуг;
сбдс_данные := УСУ_данные_пользователя;
ВЫХОД СУ.БЛОК_ДАННЫХ_индикация (сбдс. ап,
                                 сбдс.ао,
                                 сбдс.ку,
                                 сбдс.данные);
end;
                                                    (* переход # 3*)
from ИСХОДНОЕ to ЗАВЕРШЕНО
when ПСТ.[нд_подсети].БЛОК_ДАННЫХ_индикация
provided (локальный_адр_АИПСУ (ПСТ_данные_пользователя.ап)  and
        ПСТ_данные_пользователя.сс = НУЛЬ and
        not ПСТ_данные_пользователя.дс)
begin
пбд := ПСТ_данные_пользователя;
if (пбд.тп_пбд = ДН) then
    ВЫХОД СУ.БЛОК_ДАННЫХ_индикация (получить_адр_ПДУСУ
                                    (пбд.ап, пбд.дл_ап),
                                    получить_адр_ПДУСУ (пбд.ао,
                                    пбд.дл_ао),
                                    получить_ку (пбд.факультатив-
                                    ные_возможности),
                                    пбд.данные);
else
    послать_информирование_об_ошибке (пбд);
end;
from ИСХОДНОЕ to СБОРКА
when ПСТ [нд_подсети].БЛОК_ДАННЫХ_индикация
provided локальный_адр_АИПСУ (ПСТ_данные_пользователя.ап) and
    ((ПСТ данные_пользователя.сс > НУЛЬ) or (ПСТ_данные_пользова-
     теля.дс))
```

```
begin
пбд := ПСТ_данные_пользователя;
распределитель_ресурсы_сборки (пбд.общ_длина, пбд.факультативные_
                               возможности);
освободить_буфер (прм_буф);
объединить_сегм (прм_буф,
                 пбд.сс,
                 пбд.данные);
ВЫХОД СТ.ТАЙМ_АУТ_запрос (пбд.время_существования,
                           тайм_аут_времени_существования,
                           НУЛЬ);
end;
                                                    (* переход # 5*)
from ИСХОДНОЕ to завершено
when ПСТ [нд_подсети].БЛОК_ДАННЫХ_индикация
provided not локальный_адр_АИПСУ (ПСТ_данные_пользователя.ап)
begin
пбд := ПСТ_данные_пользователя;
if (пбд.время_существования > оценочное_истекшее_время) then
begin
пбд.время_существования := пбд.время_существования — оценочное_
                           истекшее_время;
регистрация_маршрута (пбд.факультативные_возможности);
передать_пбд(пбд);
end
else
передать_информирование_об_ошибке (ВРЕМЯ_СУЩЕСТВОВАНИЯ_
ИСТЕКЛО_ПРИ_ПЕРЕХОДЕ, пбд);
end;
                                                    (*переход # 6*)
from СБОРКА to СБОРКА
cohen ПСТ [нд_подсети].БЛОК_ДАННЫХ_индикация
provided (ПСТ_данные_пользователя.нд_бд = пбд.нд_бд and
          ПСТ_данные_пользователя.дл_ап = пбд.дл_ап and
          ПСТ_данные_пользователя.ап = пбд.ап and
          ПСТ_данные_пользователя.дл_ао = пбд.дл_ао and
          ПСТ_данные_пользователя.ао = пбд.ао)
begin
          объединить_сегм (прм_буф,
                           ПСТ_данные_пользователя.сс,
                           ПСТ_данные_пользователя.данные);
end;
                                                    (* переход # 7*)
from СБОРКА to ЗАВЕРШЕНО
provided полный_блок_данных (прм_буф) no delay
begin
if пбд.тп_пбд = ДН then
     ВЫХОД СУ.БЛОК_ДАННЫХ_индикация (получить_адр_ПДУСУ
                                     (пбд.ап, пбд.дл_ао),
                                     получить_адр_ПДУСУ (пбд_ао,
                                     пбд.дл_ао),
```

```
                                        получить_ку (пбд.факультатив-
                                        ные_возможности),
                                        извлечь (прм_буф, длина_буф
                                        (прм_буф))),
else
    послать_информирование_об_ошибке (пбд);
ВЫХОД СТ.ТАЙМ-АУТ_аннулирование (тайм-аут_времени_существования,
                                  НУЛЬ);
освободить_ресурсы_сборки;
end;
                                                    (* переход # 8*)
from СБОРКА to ЗАВЕРШЕНО
when СТ.ТАЙМ-АУТ_ответ
begin
передать_информирование_об_ошибке (ВРЕМЯ_СУЩЕСТВОВАНИЯ_
                                    ИСТЕКЛО_ПРИ_СБОРКЕ, пбд);
end.
```

А.5. Диаграмма переходов состояний

Диаграмма переходов состояний конечного автомата приведена на черт. 2.



Черт. 2

ПРИЛОЖЕНИЕ В
Справочное

ВСПОМОГАТЕЛЬНЫЙ ТЕХНИЧЕСКИЙ МАТЕРИАЛ

В.1. Время существования блока данных

Можно выделить две основные цели обеспечения в протоколе, определенном настоящим стандартом, функции «время существования ПБД». Первая цель — предотвратить неограниченное зацикливание в сети протокольных блоков данных; если алгоритм маршрутизации должен гарантировать, что зацикливание данных будет происходить очень редко, то поле «время существования ПБД» обеспечит дополнительное ограничение на длительность такого зацикливания.

Другая цель обеспечения функции «время существования» — предусмотреть средства, с помощью которых инициирующий логический-объект-сетевого-уровня может ограничивать максимальное время существования СБДС. Класс 4 протокола транспортного уровня (ГОСТ 34.961) исходит из того, что установлено конкретное значение максимального времени существования СБДС для того, чтобы предотвратить некоторые ошибочные состояния в фазах установления и завершения соединения транспортного уровня; а именно, если ПБДТ не поступает в течение максимального времени существования СБДС, то нет никаких шансов, что он когда-нибудь поступит. Такое допущение необходимо сделать, даже если сетевой уровень не обеспечивает никакой конкретной верхней границы времени существования СБДС; однако для класса 4 протокола транспортного уровня более просто допустить потерю ПБДТ, чем их запоздалое поступление и по этой причине предпочтительно аннулировать запоздалые ПБДТ, чем доставлять их. Следует заметить, что время существования СБДС не связано непосредственно с повторной передачей потерянных ПБДТ. Оно более полезно для различия старых (продублированных) ПБДТ и новых ПБДТ.

Максимальное время существования СБДС должно обеспечиваться для логического объекта протокола транспортного уровня в единицах времени для того, чтобы его можно было использовать при определении значений тайм-аутов транспортного уровня (логический-объект-транспортного-уровня не может подсчитывать «шаги»).

При отсутствии любой гарантированной верхней границы принято оценивать значение максимального времени существования СБДС. Это значение часто основывается на наблюдении последних рабочих характеристик и может быть различным для разных отправителей и получателей. Существует два возможных способа выполнения требования по ограничению максимального времени существования СБДС: 1) обеспечить механизм в транспортном уровне для опознавания и аннулирования старых ПБДТ или 2) определять время существования в единицах времени. Второй способ требует, чтобы промежуточные системы уменьшали поле «время существования» на значение, представляющее собой верхний предел времени, истекшего после поступления ПБД в предыдущую промежуточную систему. Транспортный уровень полагается на то, что сетевой уровень аннулирует СБДС (и, следовательно, ПБДТ), время существования которых истекло.

Основной недостаток применения первого способа состоит в том, что логические -объекты-транспортного-уровня (экземпляры) создаются при необходимости и освобождаются, когда их задача выполнена; следовательно, по своему характеру они являются временными. Для того чтобы определить, устарел ли данный ПБДТ, должны быть назначены функции, распознающие и аннулирую-

щие старые ПБДТ (и они всегда должны существовать), дополнительно к тем, которые выполняются каждым логическим-объектом-транспортного-уровня — экземпляром. Такие функции чрезвычайно сложны и создают нетривиальную перегрузку в работе транспортного уровня.

Для правильной работы автомата, связанного с обеспечением услуг в режиме без-установления-соединения, не требуется сведений о состоянии предыдущего соединения. Поскольку для обеспечения доставки старых СБДС (и, следовательно, старых ПБДТ) на транспортный уровень никаких дополнительных механизмов, кроме необходимых для правильного ограничения времени существования ПБДС, не требуется, то выгоднее, чтобы сетевой уровень аннулировал те ПБДС, время существования которых истекло и оставил для транспортного уровня проблемы потерянных ПБДТ (второй способ).

В.1.1. Определение значения времени существования ПБДС

Для каждой промежуточной системы нет никакой необходимости вычитать точное значение времени, которое истекло с тех пор, как ПБДС (содержащий ПБДТ или его сегмент) поступил в предыдущую промежуточную систему. Если точное значение нельзя получить, то достаточно вычесть верхний допуск оценки из фактически затраченного времени. В большинстве случаев промежуточная система может просто вычитать постоянное значение, зависящее от типичных, близких к максимальным, задержек, которые имеют место в конкретных услугах нижерасположенного уровня. Более точное измерение может потребоваться для тех подсетей, которые имеют как относительно большую максимальную задержку, так и относительно большие вариации длительностей задержки.

В качестве примера предположим, что конкретная ЛВС имеет короткие средние задержки при общих длительностях задержек в диапазоне 1—5 мс и случайных задержках до 20 мс. В этом случае, хотя относительный диапазон задержек может быть и больше в 20 раз, нет необходимости измерять фактическую задержку ПБДС. Постоянное значение 20 мс (или более) можно вычесть для всех ПБДС. Точно также, если отдельный участок спутникового канала имел задержку в диапазоне 0,5—0,6 с, то большее значение всегда можно использовать.

Если третья подсеть вносила обычные задержки в диапазоне 0,1—1 с, но иногда доставляла ПБДС после задержки 15 с, то подключенная к этой подсети промежуточная система могла бы счесть необходимым определение фактических затрат времени на доставку ПБДС. Даже в этом последнем примере для промежуточных систем более полезно определить, когда задержки становятся слишком большими, и аннулировать очень старые ПБДС, оставив для протокола транспортного уровня задачу обнаружения потерянных ПБДТ.

Кроме временных задержек внутри каждой подсети, важно рассмотреть временные задержки внутри промежуточных систем. Для тех шлюзовых устройств, которые могут задерживать некоторые блоки данных на значительные периоды времени, было бы относительно просто соответственно уменьшить время их существования.

В.2. Управление временем существования сборки

Для того, чтобы обеспечить предельное время существования СБДС и эффективно управлять буферами сборки на сетевом уровне, функция «сборка», описанная в разд. 6, должна управлять временем существования сегментов, представляющих частично собранные ПБД. В этом разделе рассматриваются методы ограничения времени существования сборки и предлагаются некоторые руководства относительно функции «сборка».

Когда сегменты ПБД поступают в адресуемый логический-объект-сетевого-уровня, они накапливаются в буфере до тех пор, пока ПБД не будет полностью принят, скомпанован и передан функции «разборка ПБД». Протокол не гарантирует доставку ПБД, поэтому возможна потеря некоторых сегментов ПБД или

их задержка, в результате чего весь ПБД не сможет быть собран за приемлемый отрезок времени. Например, в случае потери сегмента ПБД этот сегмент может быть потерян навсегда. Для предотвращения этого существует несколько возможных методов:

а) тайм-ауты сборки для каждого ПБД;

б) расширение функции «управление временем существования ПБД»;

в) объединение тайм-аутов повторной передачи на транспортном уровне.

Каждый из этих методов рассматривается в последующих разделах.

В.2.1. Метод а

Данный метод назначает «время существования сборки» для каждого ПБД, принимаемого и идентифицируемого его идентификатором блока данных. Это время является локальным, реальным, которое назначается функцией «сборка» и уменьшается, когда некоторые, но не все, сегменты ПБД буферизуются адресуемым логическим-объектом-сетевого-уровня. По истечении тайм-аута все сегменты ПБД аннулируются, освобождая буферы сборки и исключая возможность перепутать «очень старые» ПБД с новыми, содержащими тот же идентификатор блока данных. Для правильного функционирования этого метода тайм-ауты должны назначаться таким образом, чтобы предотвратить явление «помеха сборки» (рассматриваемое ниже). В частности, должны соблюдаться следующие правила:

а) время существования сборки должно быть значительно меньше максимального времени существования ПБД сети (чтобы предотвратить путаницу старых и новых блоков данных);

б) время существования должно быть меньше длительности тайм-аутов повторной передачи протокола транспортного уровня минус среднее время передачи по сети. Если это условие не выполняется, то добавляются дополнительные буферы для хранения данных, которые уже были повторно переданы протоколом транспортного уровня.

В.2.2. Метод б

Данный метод возможен в том случае, если действия функции «управление временем существования ПБД» основываются на реальном или виртуальном времени, а не на подсчете шагов. При этом методе поле «время существования» каждого сегмента ПБД блока данных по-прежнему должно уменьшаться функцией «сборка» адресуемого логического-объекта-сетевого-уровня так же, как если бы ПБД все еще находился в процессе передачи (это фактически имеет место). Если время существования любого сегмента частично собранного ПБД истекает, то все сегменты этого ПБД аннулируются. Этот метод привлекателен тем, что действия настоящего протокола по доставке были бы одинаковыми для сегментируемых и несегментируемых ПБД.

В.2.3. Метод в

Данный метод согласует время существования сборки непосредственно с тайм-аутами повторной передачи протокола транспортного уровня и требует, чтобы управление транспортным уровнем доводило до сведения управления сетевым уровнем (и, следовательно, функции «сборка») значения тайм-аутов повторной передачи для каждого отправителя, из которого он ожидает получать трафик. Когда сегмент ПБД принимается от отправителя, то длительность тайм-аута повторной передачи за вычетом времени предполагаемой передачи составляет время существования сборки этого ПБД. Если этот тайм-аут истечет раньше, чем будет собран весь ПБД, то все сегменты этого ПБД аннулируются. Этот метод привлекателен тем, что он обеспечивает низкую вероятность удержания тех сегментов ПБД, которые уже были повторно переданы логическим-объектом-транспортного-уровня отправителем. Его недостатком, однако, является зависимость эффективной работы от надежности действий протокола транспортного уровня. Если тайм-ауты повторной передачи установлены неправиль-

но, то возможно, что все ПБД будут аннулированы слишком быстро и протокол транспортного уровня не будет действовать.

Б.3. Эффективность функции «обнаружение ошибок заголовка»

Б.3.1. Общие положения

Формула контрольной суммы, используемой для обнаружения ошибок в заголовке, такова, что она легко вычисляется в программных и микропрограммных средствах, используя только две операции сложения на каждый октет заголовка. Она обладает эффективностью обнаружения ошибок, приближающейся (но не совсем одинаковой) к эффективности тех методов, которые используют вычисления, на которое затрачивается значительно больше времени или пространства (например циклический контроль полиномом). В этом разделе рассматривается эффективность данной функции обнаружения ошибок.

Контрольная сумма состоит из двух октетов, каждый из которых может принимать любое значение, кроме нуля. То есть каждый октет может принимать 255 различных значений. Вычисление двух октетов осуществляется таким образом, что значение каждого из них не зависит от значения другого, поэтому контрольная сумма имеет в общем $255 \times 255 = 65025$ значений. Если все возможные случаи искажения рассматривать как равновероятные, то имеется только один шанс из 65025, что контрольная сумма будет иметь правильное значение при любом конкретном искажении. Это соответствует 0,0015 % всех возможных ошибок.

В остальной части этого раздела рассматриваются конкретные классы ошибок, которые могут возникать. Есть надежда, что функция обнаружения ошибок окажется более эффективной или, по меньшей мере, не менее эффективной относительно этих классов ошибок по сравнению с общим видом ошибок.

Б.3.2. Ошибки изменения бита

Прежде всего рассмотрим те классы ошибок, которые связаны с изменением битов, но не со вставкой или выпадением битов.

Пакет ошибок длиной b представляет собой такое искажение заголовка, когда все измененные биты (по количеству не более b) находятся в пределах одного промежутка последовательно передаваемых битов, длиной b бит. Обычно предполагается, что контрольные суммы хорошо справляются с пакетами ошибок длиной, не превышающей число битов в параметре «обнаружение ошибок заголовка (16 для заголовка ПБД). Параметр «обнаружение ошибок заголовка ПБД» фактически не может обнаруживать только 0,000019 % всех ошибок, где каждый отдельный пакет ошибок длиной 16 бит или меньше встречается с равной вероятностью. В частности, он не может обнаруживать 8-битовые пакеты, в которых октет нуль изменен на октет 255 (все биты равны 1) или наоборот. Аналогично, он не может обнаруживать перестановки двух смежных октетов только в том случае, если один из них нуль, а другой 255.

Метод обнаружения ошибок заголовка ПБД, как и следовало ожидать, обнаруживает все те ошибки, которые вызваны изменением только одного бита.

Необнаруженные ошибки, вызванные изменением только двух битов, должны появляться только в том случае, если два бита отделены друг от друга на большое расстояние (и даже при этом очень редко). Метод обнаружения ошибок заголовка ПБД выявляет все двойные битовые ошибки, при которых расстояние между двумя измененными битами меньше 2040 бит = 255 октетов. Поскольку такое расстояние превышает максимальную длину заголовка, то все двойные битовые ошибки обнаруживаются.

Способность обнаружения двойных битовых ошибок является преимуществом алгоритма контрольной суммы, используемой данным протоколом, по отношению к простому суммированию по модулю 65536 заголовка, разделенного на поля по 16 битов. Такое простое суммирование не сможет охватить все такие

двойные битовые ошибки. Фактически двойные битовые ошибки с расстоянием менее 16 бит могут не обнаруживаться.

Случай, когда сама контрольная сумма из-за ошибки целиком установлена в нуль, рассмотрена в п. В.3.4.

В.3.3. Ошибки вставки/выпадания битов

Хотя ошибки, связанные со вставками или выпаданиями битов, в общем случае обнаруживаются с более или менее одинаковой вероятностью, что и все другие виды общих ошибок, но, по меньшей мере, один класс таких ошибок представляет особый интерес. Если октеты, каждый из которых равен нулю или 255, добавляются в таких точках, что простая сумма C_0 в текущем вычислении (описанном в п. В.3.4) оказывается равной нулю, то ошибка становится необнаруживаемой. Это представляет первостепенный интерес, так как существуют два момента в вычислении, при которых такое значение суммы является нередким, а часто ожидаемым, а именно, в начале и в конце. То есть, если заголовку предшествуют или за ним следуют добавленные октеты, каждый из которых равен нулю или 255, то ни одна ошибка не будет обнаружена. Оба случая рассматриваются по отдельности.

Появление ошибочных октетов в начале заголовка полностью нарушает структуру полей заголовка, обусловливая тем самым их неправильную интерпретацию. В частности, первый такой введенный октет рассматривается как идентификатор протокола сетевого уровня, что, вероятно, исключает какие бы то ни было сведения о том, что этот блок данных относится к настоящему протоколу, и тем самым исключает любые попытки вычислить контрольную сумму или привлечь другие методы ее вычисления.

Появление ошибочных октетов в конце заголовка при отсутствии других ошибок невозможно, так как поле «длина» однозначно определяет место окончания заголовка. Включение или удаление октетов в конце заголовка требует изменения значения октета, определяющего длину заголовка. Такое изменение предполагает, что значение вычисленной суммы в конце заголовка не будет иметь критического значения нуль и, следовательно, эта ошибка в общем случае обнаруживается точно с такой же вероятностью, как и любая ошибка вообще.

Появление ошибочного октета в середине заголовка представляет основной интерес, если добавленный октет имеет значение 0 или 255, и если переменная C_0 при этом оказывается равной нулю. В большинстве случаев эта ошибка полностью нарушает анализ структуры заголовка, что может привести к аннулированию блока данных. Кроме того, при отсутствии любых других ошибок последний октет заголовка будет рассматриваться как данные. Это, в свою очередь, обусловит окончание заголовка в ошибочном месте. И наоборот, если заголовок может быть проанализирован правильно, может оказаться, что последнее поле будет пропущено. Даже в том случае, когда последнее поле является факультативным заполнителем и, следовательно, не является обязательным, длина поля функции «заполнение» окажется несовместимой с полем «длина заголовка» и, следовательно, ошибка может быть обнаружена.

В.3.4. Ошибки, приводящие к невозможности вычисления контрольной суммы

Использование функции «обнаружение ошибок заголовка» факультативно. Вариант ее неиспользования указывается нулевым значением параметра «контрольная сумма». Это создает возможность того, что оба октета параметра «контрольная сумма» (ни один из которых не генерируется в значении 0) могут принять нулевое значение. Фактически это привело бы к необнаружению ошибки контрольной суммой в связи с невозможностью выполнения проверки. Возможна одна из трех ситуаций:

а) пакет ошибок длиной 16 бит, устанавливающий всю контрольную сумму в нуль. Такая ошибка не может быть обнаружена; однако это возможно, если такой пакет будет размещен в конкретной позиции заголовка (Вычисление его

влияния на полную обнаруживаемость пакетов ошибок зависит от длины заголовка);

б) обнаруживаются все одиночные битовые ошибки. Так как оба октета поля «контрольная сумма» должны быть ненулевыми при использовании контрольной суммы, то ни одна одиночная битовая ошибка не может установить контрольную сумму в нуль;

в) если каждый из двух октетов параметра «контрольная сумма» имеет значение, выражаемое степенью два, причем только один бит в каждом октете равен единице (1), то обнуление параметра «контрольная сумма» может привести к необнаружению двойной битовой ошибки. Более того, два измененных бита имеют расстояние меньше шестнадцати (16) и могут быть смежными. Это, очевидно, отличается от описанной ранее полной обнаруживаемости.

В тех случаях, когда какая-то администрация сильно озабочена возможностью случайного обнуления контрольной суммы среди блоков данных в своей области сетевой адресации, эта администрация может наложить ограничение, состоящее в том, что все блоки данных отправителя и получателя, находящиеся в пределах данной области сетевой адресации, должны использовать функцию «обнаружение ошибок заголовка». Любые блоки данных, которые не могут быть аннулированы, также не будут разрешены вне области сетевой адресации. Это защищает от ошибок, появляющихся в области сетевой адресации, и может защитить все блоки данных, у которых адреса отправителя и получателя находятся в пределах области сетевой адресации, даже если маршрут данных между всеми такими парами проходит через другие области сетевой адресации (несмотря на ошибки вне защищенной области сетевой адресации).

ПРИЛОЖЕНИЕ С
Справочное

АЛГОРИТМЫ ФУНКЦИИ «ОБНАРУЖЕНИЕ ОШИБОК В ЗАГОЛОВКЕ ПБД»

С.1. Символы, используемые в алгоритмах

C_0, C_1 — переменные, используемые в алгоритме;

i — номер (т. е. позиция) октета в заголовке (позиция первого октета $i=1$);

O_i — значение октета i заголовка ПБД;

n — номер (т. е. позиция) первого октета параметра «контрольная сумма» ($n=8$);

L — длина заголовка ПБД в октетах;

X — значение октета «один» параметра «контрольная сумма»;

Y — значение октета «два» параметра «контрольная сумма».

С.2. Соглашения по арифметическим действиям

Сложение выполняется одним из следующих способов:

а) арифметическое по модулю 255;

б) восьмибитовое арифметическое дополнение до единиц, при котором каждая из переменных в значении минус нуль (т. е. 255) должна рассматриваться так, как если бы она имела значение плюс нуль (т. е. 0).

С.3. Алгоритм выработки параметров «контрольная сумма»

Сформировать полный заголовок ПБД со значением поля параметра «контрольная сумма», равным нулю:

А: $C_0 \leftarrow C_1 \leftarrow 0$.

Б: Последовательно обработать каждый октет заголовка ПБД от $i=1$ до L путем

$$C_0 \leftarrow C_0 + O_i;$$
$$C_1 \leftarrow C_1 + C_0.$$

В: Вычислить:

$$X \leftarrow (L-8)C_0 - C_1 \quad (\text{мод } 255);$$
$$Y \leftarrow (L-7)(-C_0) + C_1 \quad (\text{мод } 255).$$

Г: Если $X=0$, то $X \leftarrow 255$.

Д: Если $Y=0$, то $Y \leftarrow 255$.

Е: Поместить значения X и Y в октеты 8 и 9 соответственно.

С.4. Алгоритм проверки параметра «контрольная сумма»

А: Если оба октета 8 и 9 заголовка ПБД равны нулю (все биты сброшены), то контрольная сумма вычислена правильно; если только один из двух октетов, но не оба, равен нулю, то контрольная сумма вычислена неправильно; в остальных случаях инициировать.

$$C_1 \leftarrow C_0 \leftarrow 0.$$

Б: Последовательно обработать каждый октет заголовка ПБД от $i=1$ до L путем

$$C_0 \leftarrow C_0 + O_i;$$
$$C_1 \leftarrow C_1 + C_0$$

В: Если после обработки всех октетов $C_0 = C_1 = 0$ (мод 255), то контрольная сумма вычислена правильно, в противном случае она вычислена неверно.

С.5. Алгоритм подгонки параметра «контрольная сумма» при изменении октета

Данный алгоритм модифицирует контрольную сумму при изменении октета (например поля «время существования»). Предположим, что значение k октета заменено на Z, равное новому значению минус прежнее значение.

Если X и Y указывают значения контрольной суммы, содержащиеся в октетах n и $n+1$ соответственно, то модификация X и Y выполняется следующим образом:

если $X = 0$ и $Y = 0$, то ничего не предпринимается, если $X = 0$ или $Y = 0$, то контрольная сумма неверна, кроме того:

$$X \leftarrow (k-n-1)Z + X \quad (\text{мод } 255),$$

$$Y \leftarrow (n-k)Z + Y \quad (\text{мод } 255).$$

Если $X = 0$, то $X \leftarrow 255$; если $Y = 0$, то $Y \leftarrow 255$.

Для данного протокола $n = 8$. Если измененным октетом является поле «время существования», то $k = 4$. В случае, когда время существования уменьшено на один элемент ($Z = -1$), то присвоение операторов новым значениям X и Y в непосредственно предшествующем алгоритме упрощается:

$$X \leftarrow X + 5 \quad (\text{мод } 255),$$

$$Y \leftarrow Y - 4 \quad (\text{мод } 255).$$

Примечание. Для получения этого результата предположим, что к октету k добавлено значение Z, тогда к X и Y добавляются значения Z_x и Z_y. Для того чтобы параметр «контрольная сумма» удовлетворял условию п. 6.11 как до, так и после добавления этих значений, должны выполняться следующие условия:

$$Z + Z_x + Z_y = 0 \quad (\text{мод } 255)$$

и

$$(L-k+1)Z + (L-n+1)Z_x + (L-n)Z_y = 0 \quad (\text{мод } 255).$$

Решая эти уравнения одновременно, получаем:

$$Z_x = (k-n-1)Z,$$

и

$$Z_y = (n-k)Z.$$

ИНФОРМАЦИОННЫЕ ДАННЫЕ

1. **ПОДГОТОВЛЕН и ВНЕСЕН** Техническим Комитетом ТК 22 «Информационная технология»

2. **УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ** Постановлением Комитета Российской Федерации по стандартизации, метрологии и сертификации от 28.12.92 № 1572

   Настоящий стандарт подготовлен методом прямого применения международного стандарта ИСО 8473—88 «Системы обработки информации. Взаимосвязь открытых систем. Протокол для обеспечения услуг сетевого уровня в режиме без установления соединения» и полностью ему соответствует

3. **Срок проверки** — 1999 г., периодичность проверки — 5 лет

4. **ВВЕДЕН ВПЕРВЫЕ**

5. **ССЫЛОЧНЫЕ НОРМАТИВНО-ТЕХНИЧЕСКИЕ ДОКУМЕНТЫ**

Обозначение отечественного НТД, на который дана ссылка	Обозначение соответствующего международного стандарта	Номер раздела, пункта, приложения, в котором дана ссылка
ГОСТ 28906—91	ИСО 7498—84 с Доп. 1	0, 2, 3.1
ГОСТ 34.961—91	ИСО 8073—88	Приложение В
ГОСТ 34.950—92	ИСО 8208—87	1, 2, 3.6, 8.2.1, 8.4.3, 8.4.3.8
ГОСТ 34.951—92	ИСО 8348—87 с Доп. 1	0, 1, 2, 5.4, 6.13, приложение А
—	ИСО 8348/Доп. 2—87*	2, 5.3.7, 7.3.11
—	ИСО 8509—87*	2, 3.2
—	ИСО 8648—88*	5.1
ГОСТ 28907—91	ИСО 8802/2—89	1, 2, 8.4.2
ГОСТ 34.913.3—91	ИСО 8802/3—89	2
—	ИСО 9074—84*	2

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат ТК 22 «Информационная технология».

СОДЕРЖАНИЕ

0. Введение 1
1. Назначение и область применения 2
2. Ссылки 3
Часть 1. Общие положения 4
3. Определения 4
4. Сокращения 6
5. Краткое описание протокола 7
Часть 2. Спецификация протокола 12
6. Функции протокола 12
7. Структура и кодирование ПБД 29
8. Обеспечение услуг нижерасположенного уровня 47
9. Соответствие 59
Приложение А. Формализованное описание 62
Приложение В. Вспомогательный технический материал 78
Приложение С. Алгоритм функции «обнаружение ошибок в заголовке ПБД» 84
Информационные данные 86

Редактор *В. М. Лысенкина*
Технический редактор *В. Н. Прусакова*
Корректор *А. И. Зюбан*

Сдано в набор 21.01.93 Подп. в печ. 20.05.93. Усл. печ. л. 5,11. Усл. кр.-отт. 6,24
Уч.-изд. л. 5,83. Тир. 359 экз. С 212.

Ордена «Знак Почета» Издательство стандартов, 107076, Москва, Колодезный пер., 14.
Калужская типография стандартов, ул. Московская, 256. Зак. 181

gost.e-ecolog.ru