МИНИСТЕРСТВО СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНКОМСВЯЗЬ РОССИИ)

МИНИСТЕРСТВО ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ЗАРЕГИСТРИРОВАНО
Регистрационный № 40066
от "10" декабря 2015 г.

ПРИКАЗ

23.07.2015 № 279

Москва

Об утверждении Требований к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий

В целях реализации требований части 2 статьи 64 Федерального закона от 7 июля 2003 г. № 126-ФЗ «О связи» (Собрание законодательства Российской Федерации, 2003, № 28, ст. 2895; № 52, ст. 5038; 2004, № 35, ст. 3607; № 45, ст. 4377; 2005, № 19, ст. 1752; 2006, № 6, ст. 636; № 10, ст. 1069; № 31, ст. 3431, ст. 3452; 2007, № 1, ст. 8; № 7, ст. 835; 2008, № 18, ст. 1941; 2009, № 29, ст. 3625; 2010, № 7, ст. 705; № 27, ст. 3408; № 31, ст. 4190; 2011, № 9, ст. 1205; № 25, ст. 3535; № 27, ст. 3880; № 29, ст. 4291; № 30, ст. 4590; № 45, ст. 6333; № 49, ст. 7061; № 50, ст. 7351, ст. 7366; 2012, № 31, ст. 4322, ст. 4328; № 53, ст. 7578; 2013, № 19, ст. 2326; № 27, ст. 3450; № 43, ст. 5451; № 44, ст. 5643; № 48, ст. 6162; № 49, ст. 6339, ст. 6347; № 52, ст. 6961, ст. 5038; 2014, № 6, ст. 560; № 14, ст. 1552; № 19, ст. 2302; № 26, ст. 3366; № 30, ст. 4229), пунктов 4, 6, 11 Правил взаимодействия операторов связи с уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность, утвержденных постановлением Правительства Российской Федерации от 27 августа 2005 г. № 538 (Собрание законодательства Российской Федерации, 2005, № 36, ст. 3704; 2007, № 48, ст. 6010; 2008, № 42, ст. 4832; 2013, № 15, ст. 1804),

ПРИКАЗЫВАЮ:

1. Утвердить прилагаемые Требования к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий (далее – Требования).

2. Операторы почтовой связи обязаны обеспечить выполнение Требований не позднее одного года со дня утверждения плана мероприятий по внедрению технических средств, предусмотренного Правилами взаимодействия операторов связи с уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность, утвержденными постановлением Правительства Российской Федерации от 27.08.2005 № 538.

3. Направить настоящий приказ на государственную регистрацию в Министерство юстиции Российской Федерации.

4. Контроль за исполнением настоящего приказа возложить на заместителя Министра связи и массовых коммуникаций Российской Федерации М.Я. Евраева.

Министр Н.А. Никифоров

УТВЕРЖДЕНЫ
приказом Минкомсвязи России
от 23.07.2015 г. № 279

Требования
к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий

1. Общие требования к информационным системам, содержащим базы данных абонентов оператора связи и предоставленных им услугах связи

Технические средства оперативно-розыскных мероприятий, содержащие базы данных операторов почтовой связи (далее – ОПС[1]), необходимые для обеспечения проведения оперативно-розыскных мероприятий на сетях почтовой связи, предназначены для накопления, хранения, обработки и предоставления уполномоченным государственным органам, осуществляющим в соответствии с Федеральным законом от 12.08.1995 № 144-ФЗ «Об оперативно-розыскной деятельности» (Собрание законодательства Российской Федерации, 1995, № 33, ст. 3349; 1997, № 29, ст. 3502; 1998, № 30, ст. 3613; 1999, № 2, ст. 233; 2000, № 1, ст. 8; 2001, № 13, ст. 1140; 2003, № 2, ст. 167; № 27, ст. 2700; 2004, № 27, ст. 2711; № 35, ст. 3607; 2005, № 49, ст. 5128; 2007, № 31, ст. 4008, ст. 4011; 2008, № 18, ст. 1941; № 52, ст. 6227, ст. 6235, ст. 6248; 2011, № 1, ст. 16; № 48, ст. 6730; № 50, ст. 7366; 2012, № 29, ст. 3994; № 49, ст. 6752; 2013, № 14, ст. 1661; № 26, ст. 3207; № 44, ст. 5641; № 51, ст. 6689) оперативно-розыскную деятельность (далее – уполномоченные органы), информации о пользователях услуг связи, оказанных им услугах связи, а также иной информации.

1.1. Настоящие требования распространяются на ТС ОРМ, используемые и/или подлежащие установке на сетях операторов связи, осуществляющих деятельность в области оказания услуг почтовой связи.

1.2. Устанавливаемые операторами связи на информационные системы операторов почтовой связи ТС ОРМ должны представлять собой АПК, обеспечивающий процессы сбора, регистрации, обработки, хранения и передачи уполномоченным органам информации о пользователях услугами почтовой связи и предоставленных им услугах связи, поступающей от одного или нескольких операторов почтовой связи.

---

[1] Перечень используемых терминов и сокращений содержится в Приложении № 9 к настоящим требованиям.

1.3. ИС ОПС обеспечивают передачу на ТС ОРМ информации обо всех оказанных услугах почтовой связи и пользователях услугами почтовой связи в соответствии с Приложением №1.

2. Общие функциональные (технические) требования

2.1. Построение ТС ОРМ должны обеспечивать:

2.1.1. организацию БД для накопления, хранения в течение трех лет и обработки информации о пользователях услуг связи, оказанных им услугах связи, иной информации;

2.1.2. прием и первичную обработку информации о пользователях услуг связи и оказанных им услугах связи из ИС ОПС от одного или нескольких операторов связи для занесения в БД ТС ОРМ;

2.1.3. добавление в БД ТС ОРМ информации об оказанных услугах почтовой связи, независимо от того, присутствует или нет в БД ТС ОРМ информация об абонентах (пользователях), которым услуги связи предоставляются;

2.1.4. ИС ОПС выполняют передачу информации на ТС ОРМ о пользователях услуг связи и оказанных им услугах почтовой связи не реже одного раза в час;

2.1.5. защиту от несанкционированного доступа к данным ТС ОРМ с помощью программных и технических средств, а также организационных мер;

2.1.6. круглосуточный автоматизированный удаленный доступ к ТС ОРМ со стороны ПУ уполномоченных органов;

2.1.7. прием из ПУ различных видов запросов о пользователях услуг связи и оказанных им услугах связи;

2.1.8. передачу на ПУ из ТС ОРМ информации в соответствии с запросами;

2.1.9. регистрацию и хранение поступающих запросов, удаление запросов уполномоченными пользователями.

2.2. ТС ОРМ должны являться завершенным модульным АПК, предоставляющим возможность оператору связи ввода в эксплуатацию с любого набора услуг, согласно лицензиям на предоставление услуг связи. Модульное построение должно обеспечивать возможность расширения (замены) состава аппаратных и программных средств ТС ОРМ для улучшения эксплуатационных характеристик ТС ОРМ по мере увеличения количества абонентов, возрастания объемов обрабатываемой информации, расширения функций системы без изменения информации, хранящейся в уже сформированных базах данных.

2.3. ТС ОРМ должны содержать ПО мониторинга и управления КТС ТС ОРМ со стороны ПУ.

2.4. ТС ОРМ должны вести системные журналы, включающие в себя:

— информацию о сессиях;
— информацию о запросах на получение данных;
— информацию об ответах на запросы о получении данных;
— информацию об отчетах;
— информацию о текущей конфигурации КТС, системного и прикладного ПО;

— информацию об изменениях в конфигурации КТС, системного и прикладного ПО;

— информацию о неполадках в КТС, системном и прикладном ПО;

— информацию о доступе технического персонала оператора связи к ТС ОРМ;

— информацию о НСД и попытках НСД.

ТС ОРМ не должны фиксировать в системных журналах никакой информации, относящейся к идентификаторам, содержимому поисковых задач ПУ.

2.5. Доступ технического персонала оператора связи к ТС ОРМ должен осуществляться через модуль доступа, который обеспечивает запись команд управления ТС ОРМ и построение парольной системы.

Технический персонал оператора связи должен иметь доступ только к системным журналам ТС ОРМ, системному ПО и КТС, входящим в ТС ОРМ для проведения ремонтных работ.

2.6. ТС ОРМ должны исключать возможность передачи любой информации, касающейся функционирования ТС ОРМ на оборудование ИС ОПС.

2.7. ТС ОРМ независимо от своего функционального состояния не должны нарушать работу ИС ОПС.

2.8. Проведение организационно-технических мероприятий должно обеспечивать защиту ТС ОРМ от несанкционированного доступа:

— производителя ТС ОРМ;

— неавторизованных пользователей;

— технического персонала оператора связи;

— третьих лиц.

2.9. Защита информации должна осуществляться при:

— загрузке информации в ТС ОРМ;

— получении ТС ОРМ запросов;

— обработке информации в ТС ОРМ;

— хранении запросов к ТС ОРМ и результатов их выполнения;

— передаче на ПУ результатов обработки запросов;

— получении ТС ОРМ информации от ИС ОПС.

2.10. К ТС ОРМ должны иметь доступ:

— программные средства ИС ОПС о пользователях услуг связи и оказанных им услугах связи (источник информации);

— технический персонал оператора связи;

— ПУ уполномоченных органов.

2.11. Источник информации должен иметь возможность осуществлять ввод информации в ТС ОРМ в согласованном с поставщиком ТС ОРМ формате в автоматическом и/или автоматизированном режиме.

2.12. Для технического персонала оператора связи ТС ОРМ должны предоставлять следующие функции:

— доступ к журналу ошибочных блоков переданных отчетов и возможность редактирования ошибочных записей соответствующих отчетов;

— доступ к аппаратным и программным компонентам ТС ОРМ для проведения регламентных и ремонтных работ.

2.13. Технический персонал оператора связи, обслуживающий ТС ОРМ, должен иметь доступ только через специальные технические средства защиты.

2.14. Если в связи со структурой организации сети связи ОПС обслуживает одновременно несколько входящих в его структуру территориально-распределенных филиалов по субъектам Российской Федерации (отдельных юридических лиц), то ТС ОРМ могут накапливать информацию об абонентах и оказанных им услугах связи для нескольких филиалов ОПС.

При обращении ПУ с поисковыми задачами к ТС ОРМ ПУ должен иметь возможность явно задавать перечень филиалов, для которых должна выполняться поисковая задача.

При явном указании списка филиалов оператора в параметрах поисковой задачи, запрос в целом должен выполняться в соответствии с временными характеристиками, приведенными в Разделе 3 «Требования, предъявляемые к временным характеристикам ТС ОРМ и БД» настоящих Требований. Обновление в БД ТС ОРМ информации об абонентах и оказанных им услугах связи по каждому филиалу (другому юридическому лицу) должно выполняться в соответствии с п. 2.1.4. настоящих Требований.

2.15. В случае, если какая-либо составная часть информации об абонентах оператора связи хранится в неструктурированном строковом виде (т.е. не разделенная на составные части в соответствии с описанием таких данных об абонентах согласно Приложению № 7), при выполнении запросов об абонентах ТС ОРМ должны осуществлять полнотекстовый поиск по такой неструктурированной информации, используя поисковые критерии задачи.

3. Требования, предъявляемые к временным характеристикам ТС ОРМ и БД

3.1. Время выполнения задач в БД ТС ОРМ поиска информации о пользователях и оказанных им услугах связи не превышает значений, приведенных в Таблице 1.

Требования к временным характеристикам поиска информации в ТС ОРМ разделяются по следующим классам:

— класс 1 (К1) включает критерии:

— идентификатор почтового отправления;

— класс 2 (К2) включает критерии:

— фамилия, имя, отчество (при наличии), наименование организации пользователя услугами почтовой связи;

— почтовый адрес пользователя услугами почтовой связи;

— паспортные данные пользователя услугами почтовой связи;

— номер договора на оказание услуг почтовой связи;

— класс 3 (К3) включает критерии, состоящие из атрибутов записи о почтовом отправлении.

Таблица 1

	Класс параметра запроса	Временной интервал				
		до суток	до 1 месяца	до 6 месяцев	до 1 года	до 3 лет
	К1	< 2 сек.	< 5 сек.	< 15 сек.	< 20 сек.	< 40 сек.
	К2	< 5 сек.	<30 сек.	<60 сек.	<80 сек.	<180 сек.
	К3	< 5 сек.	< 20 сек.	<40 сек.	<60 сек.	< 90 сек.

* Примечание: для отсутствующих в таблице идентификаторов время выполнения задачи в ТС ОРМ может быть больше приведенного.

3.2. ТС ОРМ должны поддерживать возможность выполнения комбинированных запросов. Комбинированными запросами являются поисковые критерии согласно п. 3.1., объединенные логическими операциями.

ТС ОРМ должны поддерживать следующие логические операции для объединения критериев: «И», «ИЛИ» операции группировки критериев «(», «)» и логическое «НЕ».

Использование критериев с использованием операции логического «НЕ» допускается только совместно с другими критериями, не являющимися логическим отрицанием, и объединяется с ними с помощью логических операций объединения критериев. Результирующий критерий должен иметь хотя бы одну логическую операцию «И».

К времени поиска информации в БД ТС ОРМ при выполнении комбинированных поисковых запросов предъявляются следующие требования:

— при выполнении поискового запроса по двум поисковым критериям в зависимости от вида логической операции:

— для операции «И»: T(AB) – максимум (T(A), T(B))*C, но меньше, чем T(A) + T(B), где C – константа (допускается значение больше единицы), T(A), T(B) – время выполнения поисковых запросов по каждому из критериев, T(AB) – время выполнения поискового запроса по критериям, объединенным операцией «И»);

— для операции «ИЛИ»: T(AB) меньше или равно T(A) + T(B), где A, B – поисковые критерии, T(A), T(B) – время выполнения поисковых запросов по каждому из критериев, T(AB) – время выполнения поискового запроса по критериям, объединенным операцией «ИЛИ»);

— для операции «НЕ»: T(A(«НЕ» B)) меньше либо равно T(AB), где A, B – поисковые критерии, T(AB) время выполнения поискового критерия без операции логического «НЕ»;

— при выполнении комбинированного поискового запроса по большему числу критериев (больше двух), требования ко времени поиска должны формироваться аналогично описанному выше способу с учетом операций группировки критериев:

— для последовательного объединения поисковых критериев T(ABC) больше или равно максимум (T(AB), T(C))*C, где T(AB) определяется аналогично варианту комбинирования двух критериев, а C – константа больше единицы;

— для объединения критериев с операциями группировки T(A(BC)) меньше или равно T(A) + T(BC), где T(BC) определяется аналогично варианту комбинирования двух критериев.

Приложение № 1 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

Требования к составу накапливаемой ТС ОРМ информации, получаемой с ИС ОПС

1.1. ТС ОРМ должны позволять накапливать следующую информацию о пользователях услугами почтовой связи:

— для абонента - юридического лица:

— полное наименование юридического лица;

— ИНН;

— ОГРН;

— ОКПО;

— почтовый индекс;

— для юридических лиц, с которыми заключен договор на оказание услуг связи:

— идентификатор филиала оператора связи;

— дата и время заключения договора;

— номер договора;

— дата и время расторжения договора;

— банк абонента для расчетов с ОПС;

— номер счета абонента в банке, используемого для расчетов с ОПС;

— корреспондентский счет;

— КПП;

— БИК;

— информация о контактных лицах, включающая:

— фамилию, имя, отчество (при наличии) контактного лица;

— контактные телефоны;

— список контактных электронных адресов;

— адреса и их почтовые индексы, указанные в заключенном договоре.

Информация накапливается для юридических лиц, заключивших договоры на предоставление услуг почтовой связи.

1.2. ТС ОРМ должны позволять накапливать следующую информацию об отправителе, получателе почтового отправления:

— для абонента - физического лица:

— фамилия, имя, отчество (при наличии);

— дата рождения;
— паспортные данные, в т.ч.:
— вид документа, удостоверяющего личность;
— серия, номер удостоверения личности;
— когда и кем выдано;
— контактные телефоны;
— номер договора;
— контактный адрес абонента;
— для абонента - юридического лица:
— полное наименование юридического лица;
— фамилия, имя отчество (при наличии) отправителя (контактного лица);
— контактные телефоны;
— почтовый индекс;
— номер договора;
— контактный адрес абонента.

1.3. ТС ОРМ должны позволять накапливать следующую информацию об оказанных услугах почтовой связи:

— идентификатор филиала оператора связи;
— идентификатор почтового отправления;
— дата регистрации почтового отправления оператором связи;
— идентификатор региона отправителя;
— идентификатор региона получателя;
— атрибут операции по обработке почтового отправления;
— описание операции по обработке почтового отправления;
— информация об отправителе (юридическое или физическое лицо);
— информация о получателе (юридическое или физическое лицо);
— информация об узле назначения (получателя):
— индекс отделения почтовой связи;
— наименование отделения почтовой связи;
— тип операции по обработке отправления на узле почтовой связи;
— информация об узле обработки почтового отправления:
— индекс отделения почтовой связи;
— наименование отделения почтовой связи;
— адрес доставки отправления;
— дата и время отправления (отправителем, на узле почтовой связи);
— дата и время получения (на узле почтовом связи, получателем);
— информация о почтовом отправлении, в т.ч.
— письмо;
— посылка;
— денежный перевод;
— бандероль;

— секограмма;
— почтовая карточка;
— экспресс-отправлние;
— пакет;
— мешок, международное отправление;
— характеристики отправления, в т.ч.:
— вес почтового отправления в граммах;
— объявленная ценность почтового отправления;
— общее количество составных частей отправления;
— идентификатор составной части отправления;
— информация об оплачивающей отправление третьей стороне (юридическое или физическое лицо).

1.4. ТС ОРМ должны позволять накапливать следующую справочную информацию:

— информация об операторах связи (филиалах оператора связи), обслуживаемых ТС ОРМ, в т.ч.:
— идентификатор оператора (филиала);
— время начала действия;
— время окончания действия;
— описание (наименование) оператора (филиала);
— информация об отделениях почтовой связи оператора (операторов), в т.ч.:
— идентификатор оператора (филиала);
— индекс отделения почтовой связи;
— тип отделения почтовой связи;
— время начала действия;
— время конца действия;
— полное наименование отделения почтовой связи;
— адрес отделения почтовой связи;
— контактные телефоны;
— информация о пункте коллективного доступа, расположенного в отделении почтовой связи;
— информация о типах документов, удостоверяющих личность, в т.ч.:
— идентификатор оператора (филиала);
— идентификатор типа документа;
— время начала действия;
— время конца действия;
— описание (наименование) типа документа.

Приложение № 2 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

Требования, предъявляемые к интерфейсу взаимодействия между ПУ и ТС ОРМ

1.1. ТС ОРМ должны подключаться к ПУ в точках подключения. Интерфейс в точке подключения должен соответствовать спецификации Ethernet 10/100/1000 Base-T либо 100 Base-FX, 1000 Base-LH по согласованию с уполномоченными органами.

1.2. В качестве протокола передачи данных должен использоваться протокол TCP/IP.

1.2.1. Сетевой интерфейс первого уровня с протоколом TCP/IP используется для первичного подключения аппаратуры передачи данных – модемов, маршрутизаторов и т.п.

1.2.2. Для каждой ТС ОРМ внутри сетевого интерфейса первого уровня рекомендуется создавать свою виртуальную сеть VPN (Virtual Private Network) для туннелирования всего рабочего TCP/IP трафика между ТС ОРМ и пунктом управления, которая должна соответствовать спецификации L2TP, IPSec VPN.

1.3. Пропускная способность каналов передачи данных между ТС ОРМ и ПУ в точках подключения соответствует данным, приведенным в Таблице 1 Приложения № 2:

Таблица 1

Количество почтовых отправлений ОПС (миллион/сутки), не более	0,1	1	10	100
Скорость передачи данных (кбит/сек), не менее	1024	2048	10000	10000

1.4. Взаимодействие ТС ОРМ с оборудованием ПУ на транспортном уровне осуществляется по схеме «клиент – сервер»:

— в качестве «сервера» должна выступать ТС ОРМ;

— в качестве «клиента» выступает оборудование ПУ.

1.4.1. Для взаимодействия ТС ОРМ и ПУ организованы четыре канала передачи данных:

— кпд 1 – канал управления;

— кпд 2 – канал данных;

— кпд 3 – канал мониторинга;

— кпд 4 – канал неформатированных данных.

1.4.2. Канал управления (кпд 1) служит для передачи:

— от ПУ в ТС ОРМ запросов (команд);

— от ТС ОРМ на ПУ ответов и «сигналов».

1.4.3. Канал данных (кпд 2) служит для передачи:

— от ТС ОРМ на ПУ блоков данных отчетов генерируемых ТС ОРМ в качестве ответов на запросы из ПУ, «сигнала» Heartbeat;

— от ПУ в ТС ОРМ подтверждений о принятии блоков данных отчетов.

1.4.4. Канал мониторинга (кпд 3) служит для передачи:

— от ПУ в ТС ОРМ запросов о текущей конфигурации оборудования, системного и прикладного ПО ТС ОРМ и запросов на модификацию конфигурации;

— от ТС ОРМ на ПУ ответов на запросы ПУ, «сигналов».

1.4.5. Канал неформатированных данных (кпд 4) служит для передачи:

— от ПУ в ТС ОРМ команд на виды передаваемых неформатированных данных;

— от ТС ОРМ на ПУ неформатированных данных, «сигналов».

1.4.6. кпд 1, кпд 2, кпд 3, кпд 4 представляют собой TCP-соединения, создаваемые для подключения на заранее определенные порты оборудования ТС ОРМ. ТС ОРМ выполняет прослушивание данных портов для создания TCP-соединений с ПУ.

1.5. ПУ выполняет попытки установления подключения с ТС ОРМ в соответствии с задаваемым интервалом по предоставленным ТС ОРМ TCP-портам.

1.5.1. Установление ПУ соединений с ТС ОРМ по каналам кпд 1, кпд 2 осуществляется в следующей последовательности:

— ПУ устанавливает TCP-соединение с ТС ОРМ по порту канала кпд 1;

— выполняется процедура взаимной SSL/TLS аутентификации в соответствии с п. 1.20 настоящего Приложения;

— ПУ устанавливает TCP-соединение с ТС ОРМ по порту канала кпд 2;

— выполняется процедура взаимной SSL/TLS аутентификации в соответствии с п. 1.20 настоящего Приложения

— после успешной аутентификации ПУ выполняет создание сессии с ТС ОРМ;

— ПУ должен ожидать данных и сигналов по кпд 2 только после того, как была создана сессия по каналу кпд 1. При приеме данных и сигналов по кпд 2 при отсутствии установленной сессии по кпд 1 ПУ должен разрывать соединение по кпд 2.

1.5.2. Установление ПУ соединений с ТС ОРМ по каналам кпд 3 и кпд 4 осуществляется независимо друг от друга и от наличия установленных соединений по каналам кпд 1 и кпд 2:

— ПУ устанавливает TCP-соединение с ТС ОРМ по TCP-порту канала кпд 3/кпд 4;

— выполняется процедура взаимной SSL/TLS аутентификации в соответствии с п. 1.20 настоящего Приложения;

— после успешной аутентификации ПУ выполняет создание сессии с ТС ОРМ.

1.6. После успешной аутентификации ПУ посылает на ТС ОРМ команды в соответствии с Приложением № 7.

1.7. ПУ разрывает соединения с ТС ОРМ, если в течении трех периодов приема «сигнала» Heartbeat в каналах не было сетевой активности. При отсутствии подтверждения посланного «сигнала» Heartbeat в течении времени «максимальный размер задержки подтверждения запроса или сигнала» ТС ОРМ разрывает соединения с ПУ по соответствующему каналу.

1.8. ТС ОРМ должны обеспечивать одновременное подключение нескольких ПУ. Максимальное количество одновременно подключенных ПУ для одной ТС ОРМ – 100. ТС ОРМ обслуживает подключенные ПУ независимо друг от друга.

1.9. В ТС ОРМ на одновременном выполнении могут находиться не менее 50 одновременно запущенных задач, осуществляющих подготовку данных.

1.10. Каждому идентифицированному на ТС ОРМ ПУ назначается приоритет выполнения задач. В случае поступления на ТС ОРМ задач от различных ПУ вероятность постановки на выполнение задачи конкретного ПУ зависит от назначенного приоритета данного ПУ и приоритетов других ПУ. Распределение вероятности запуска задач от различных ПУ задается приоритетом каждого конкретного ПУ. Конфигурация по умолчанию должна обеспечивать равномерное распределение вероятности запуска задач от каждого ПУ. ТС ОРМ должны обеспечивать возможность конфигурирования приоритетов ПУ. ТС ОРМ должны обеспечивать одновременное выполнение задач по каждому идентифицированному ПУ:

— по информации об оказанных услугах почтовой связи – не менее 3;

— по пополнению справочников – не менее 1;

— по принадлежности абонентов – не менее 4.

1.11. Единицей обмена в кпд 1, кпд 2, кпд 3 и кпд 4 является «Сообщение» (Message). Форматы «Сообщений» описаны в Приложении № 7 на языке абстрактного описания синтаксиса (ASN.1) согласно ГОСТ Р ИСО/МЭК 8824-1-2001. Способ кодирования сериализованных «Сообщений» соответствует отличительным (DER) по ГОСТ Р ИСО/МЭК 8825-1-2003.

1.12. Интерфейс взаимодействия между ПУ и ТС ОРМ предусматривает наличие следующих видов «Сообщений»:

— «запросы» – передаются от ПУ в ТС ОРМ по кпд 1;

— «ответы» – передаются из ТС ОРМ на ПУ по кпд 1;

— «сигналы» – передаются из ТС ОРМ на ПУ по кпд 1 и кпд 2 (для кпд 2 только «сигнал» Heartbeat);

— «отчеты» – формируются ТС ОРМ в качестве ответов на запросы из ПУ, передаются на ПУ по кпд 2;

— «подтверждения» о принятии «отчетов» – передаются из ПУ в ТС ОРМ по кпд 2;

— «подтверждения» о принятии «сигналов» – передаются из ПУ в ТС ОРМ по кпд 1 и кпд 2 (для кдп 2 только для «сигнала» Heartbeat).

1.13. Интерфейс взаимодействия между ПУ и ТС ОРМ по каналу кпд 3 предусматривает наличие следующих видов «Сообщений»:

— «запросы»;

— «ответы»;

— «сигнал» Heartbeat;

— подтверждения о принятии «сигнала» Heartbeat и ответов.

1.14. Интерфейс взаимодействия между ПУ и ТС ОРМ по каналу кпд 4 предусматривает наличие следующих видов «Сообщений»:

— «запросы» – передаются от ПУ в ТС ОРМ;

— «ответы» – передаются из ТС ОРМ на ПУ;

— «сигналы» – передаются из ТС ОРМ на ПУ;

— «отчеты» – формируются ТС ОРМ в качестве ответов на запросы из ПУ;

— «подтверждения» о принятии «отчетов» – передаются из ПУ в ТС ОРМ;

— «подтверждения» о принятии «сигналов» – передаются из ПУ в ТС ОРМ.

1.15. ТС ОРМ выполняет любые действия, связанные с выдачей информации о пользователях услуг связи и предоставленных им услугах связи, управлением и мониторингом КТС и ПО ТС ОРМ только по «запросам» ПУ.

1.15.1. ТС ОРМ обеспечивает прием и обработку следующих «запросов», передаваемых с ПУ по кпд 1:

— «Запрос на открытие сессии» (ConnectRequest);

— «Запрос на закрытие сессии» (DisconnectRequest);

— «Запрос готовности данных» (DataReadyRequest);

— «Запрос загрузки данных» (DataLoadRequest);

— «Запрос удаления данных» (DataDropRequest);

— «Запрос прерывания загрузки данных» (DataInterruptRequest);

— «Запрос на создание задачи по обработке информации» (CreateTaskRequest);

— «Запрос на постановку/снятие объекта наблюдения на контроль» (UNIControlTaskRequest);

— «Запрос на создание задачи по обработке неформализованных данных» (NonFormalizedTaskRequest).

1.15.2. ТС ОРМ должны обеспечивать одновременную передачу блоков данных отчетов для нескольких завершенных поисковых задач по каналу кпд 2. ТС ОРМ должны обеспечивать возможность многократной передачи отчетов выполненных задач на ПУ.

1.15.3. По запросу ПУ «Запрос загрузки данных» канала кпд 1 ТС ОРМ должны обеспечивать передачу блоков отчетов по каналу кпд 2 по запрошенной задаче без внесения дополнительных временных задержек между операциями по получению записей результата задачи и преобразования их в блоки.

1.15.4. На каждый «запрос» по кпд 1 ТС ОРМ на ПУ посылается «ответ» о принятии к обработке этого запроса. ТС ОРМ обеспечивают посылку по кпд 1 на ПУ следующих «ответов»:

— «Ответ на запрос открытия сессии» (ConnectResponse);

— «Ответ на запрос закрытия сессии» (DisconnectResponse);

— «Ответ на запрос готовности данных» (DataReadyResponse);

— «Ответ на запрос загрузки данных» (DataLoadResponse);

— «Ответ на запрос удаления данных» (DataDropResponse);

— «Ответ на запрос прерывания загрузки данных» (DataInterruptResponse);

— «Ответ на запрос создания задачи» (TaskResponse);

— «Ответ на запрос на постановку/снятие объекта наблюдения на контроль» (UNIControlTaskResponse);

— «Ответ на запрос создания задачи по обработке неформализованных данных» (NonFormalizedTaskResponse).

1.15.5. По «Запросу на создание задачи по обработке информации» ТС ОРМ обеспечивают подготовку и выдачу информации из БД ТС ОРМ для следующих групп задач:

— «Задачи пополнения справочников (нормативно-справочная информация)» (DictionaryTask);

— «Задачи поисков по принадлежности абонентов» (AbonentsTask);

— «Задачи поисков по соединениям абонентов» (ConnectionsTask);

— «Задачи предоставления сведений о наличии данных» (PresenseTask).

В группу «Задачи пополнения справочников (нормативно-справочная информация)» входят запросы на получение информации справочников:

— операторы почтовой связи и их филиалы, связи, обслуживаемые ТС ОРМ;

— справочник типов документов, удостоверяющих личность;

— справочник узлов почтовой связи.

В группу «Задачи поисков по принадлежности абонентов» входят:

— «Задача на поиск информации об идентификаторах абонентов сети оператора связи зарегистрированных на физическое или юридическое лицо» (AbonentsTask).

Поиск информации выполняется в накапливаемой ТС ОРМ информации о юридических лицах, заключивших договоры на предоставление услуг почтовой связи.

В группу «Задачи поисков по соединениям абонентов» входят:

— «Задача на поиск соединений абонентов» (ConnectionsTask).

В случае, если в качестве параметров «задачи на поиск соединений абонентов» указаны только диапазон времени и (опционально) филиал оператора

связи, ТС ОРМ должны сформировать результат выполнения поисковой задачи, содержащий все соединения всех абонентов за указанный диапазон времени.

В группу «Задачи предоставления сведений о наличии данных» (PresenseTask) входят:

— запрос наличия информации по пользователям услугами почтовой связи;

— запрос наличия информации об оказанных услугах почтовой связи;

— запрос наличия справочников.

1.15.6. По «Запросу на создание задачи по обработке неформализованных данных» ТС ОРМ обеспечивают подготовку и выдачу информации из БД ТС ОРМ по следующим видам запросов:

— «запрос получения списка типов сущностей» неформализованных данных ТС ОРМ (GetEntities);

— «запрос получения списка атрибутов сущности» (GetEntityAtttibutes);

— «задача поиска неформализованных данных» (ValidateNonFormalizedTask);

— «задача предоставления сведений о наличии неформализованных данных» (NonFormalizedPresenseTask).

1.15.7. По «Запросу на постановку/снятие объекта наблюдения на контроль»;

— «запрос на создание объекта наблюдения и постановки его на контроль» (CreateUNIRequest);

— «запрос на снятие объекта наблюдения с контроля и удаление объекта наблюдения» (DropUNIRequest).

1.15.8. ТС ОРМ по «задаче поиска неформализованных данных» могут предоставлять доступ ПУ к системным журналам ТС ОРМ.

1.15.9. ТС ОРМ должны обеспечивать выполнение поисковых задач по строковым критериям, содержащими символы маскирования, включающие:

— «*» – обозначает любую комбинацию символов;

— «?» – обозначает любой один возможный символ.

ТС ОРМ должны обеспечивать выполнение поисковых задач по критериям, содержащим последовательность цифр с символом маскирования пробел (« »), означающим любую одну цифру.

Результат выполнения поисковой задачи с критерием, содержащим символы маскирования, должен содержать все записи, соответствующие заданной маске.

1.15.10. В случае возникновения в ТС ОРМ исключительных ситуаций на ПУ передаются следующие «Сообщения» типа «Сигналы», содержащие информацию об уровне важности исключительной ситуации, ее влиянии на сохранность данных и выполнение задач:

— «Нет данных (связи) с ИС ОПС» – посылается в случае отсутствия от ИС ОПС информации в течение трех часов либо при отсутствии связи с ИС ОПС;

— «Перезапуск ПО» (RestartDB);

— «Попытка несанкционированного доступа» (UnauthorizedAccess);

— «Критическая ошибка ПО, потеря данных, дальнейшая работа невозможна» (CriticalError);

— «Серьезная ошибка ПО, потеря данных, но дальнейшая работа возможна» (MajorError);

— «Незначительная ошибка ПО, данные не потеряны, дальнейшая работа возможна» (MinorError);

— «Тестовый пакет» (Heartbeat). Единственный, из «сигналов», передающийся по кпд 1, кпд 2, кпд3, кпд4 в отсутствие иной сетевой активности.

В ответ на «сигналы» поступившие от ТС ОРМ, ПУ должно передавать «подтверждения сигнала» об их приеме.

1.15.11. ТС ОРМ по «Запросу удаления данных» (DataDropRequest) должны:

— прерывать задачу, находящуюся на выполнении;

— удалять данные отчета по завершившейся задаче.

1.16. Требования к функционированию канала кпд 1 приведены в Приложении № 3.

1.17. Требования к функционированию канала кпд 2 приведены в Приложении № 4.

1.18. Требования к функционированию канала кпд 3 приведены в Приложении № 5.

1.19. Требования к функционированию канала кпд 4 приведены в Приложении № 6.

1.20. При установлении соединения ПУ и ТС ОРМ должны быть взаимно аутентифицированы. Аутентификация выполняется установлением SSL/TLS-соединения поверх установленного TCP-соединения между ПУ и ТС ОРМ. Для взаимной аутентификации ПУ и ТС ОРМ предварительно создаются X.509-сертификаты, которые сообщаются ПУ и ТС ОРМ. В случае невозможности аутентифицировать одну из сторон TCP-соединение разрывается. Созданный для ПУ сертификат используется для аутентификации только одного данного ПУ на одной ТС ОРМ по всем каналам передачи данных — кпд 1, кпд 2, кпд 3, кпд 4. ПУ и ТС ОРМ должны использовать TLS версии 1.0. Требования к сертификатам (длины ключей, прочие параметры) должны согласовываться для каждой пары ТС ОРМ и ПУ отдельно.

1.21. Открытие сессии осуществляется выполнением процедуры аутентификации согласно п. 1.20 настоящего Приложения перед началом выполнения всех запросов. Открытие сессии осуществляется посылкой по каналу управления от ПУ к ТС ОРМ «Запроса на открытие сессии» (ConnectRequest).

1.22. «Запрос на открытие сессии» устанавливает следующие параметры сессии:

— «максимальное время отсутствия активности сессии» (session-timeout) – интервал времени, по истечении которого сессия принудительно прерывается от ТС ОРМ;

— «максимальный размер блока данных отчетов» (max-data-length) в строках записей БД;

— «размер окна для канала передачи данных» (data-packet-window-size);

— «максимальная длительность задержки начала передачи блоков отчетов» (data-load-timeout);

— «максимальный размер задержки подтверждения о получении данных» (data-packet-response-timeout);

— «максимальный размер задержки подтверждения запроса или сигнала» (request-response-timeout).

Любое сообщение в соответствии с ASN.1-протоколом взаимодействия ТС ОРМ и ПУ (Приложение № 8) считается сетевой активностью.

1.22.1. ТС ОРМ при получении сообщения «Запрос на открытие сессии», анализируют параметр «размер окна для канала передачи данных», определяет максимально возможный размер окна, не превышающий полученного от ПУ. ТС ОРМ определяют минимальные значения таймаутов из параметров сессии, выбирая их не меньше, чем переданные в сообщении «Запрос на открытие сессии» (ConnectRequest). Рассчитанные значения размеров окна и таймаутов ТС ОРМ передает на ПУ в сообщении «Ответ на открытие сессии» (ConnectResponse). Полученные ПУ значения в сообщении «Ответ на открытие сессии» являются параметрами сессии между ПУ и ТС ОРМ.

1.23. Закрытие сессии осуществляется посылкой по каналу управления или каналу мониторинга от ПУ к ТС ОРМ «Запроса на закрытие сессии» или по истечению допустимого времени отсутствия активности ТС ОРМ, с посылкой сообщения-сигнала «Прерывание текущей сессии по таймауту». При этом ТС ОРМ и ПУ осуществляют разрыв текущих TCP соединений канала управления и канала данных или канала мониторинга или канала неформатированных данных

1.24. ПУ посылает ТС ОРМ «запросы» асинхронно, независимо от получения от ТС ОРМ «ответа» о приеме предыдущего «запроса».

1.25. «Запрос на создание задачи по обработке информации» приводит к созданию в ТС ОРМ задачи по обработке данных в БД ТС ОРМ, которой присваивается номер (идентификатор) задачи, передаваемый в «Ответе на запрос создания задачи» (TaskResponse). Идентификаторы задач генерируются ТС ОРМ независимо от сессий и являются уникальными в данной ТС ОРМ. ТС ОРМ присваивают идентификаторы задачам и выполняет обработку задач независимо для различных ПУ.

1.26. ПУ получает информацию о ходе выполнения и завершения обработки задач в ИС, посылая запрос «Запрос готовности данных». После завершения выполнения задачи, данные, сформированные в результате выполнения задачи, становятся доступными для загрузки в ПУ или для удаления.

1.27. ТС ОРМ при получении «Запрос загрузки данных» по кпд 1 формируют сообщения типа «отчет», состоящие из блоков данных обработанной задачи и передает их на ПУ по кпд 2.

1.28. Количество строк в каждом блоке не превышает параметр «максимальный размер блока данных отчетов», заданный при открытии сессии.

1.29. В каждом последовательном блоке данных из серии указываются идентификатор задачи, сгенерировавшей данный отчет, общее количество блоков в отчете, порядковый номер каждого блока.

1.30. Данные задачи, полученные в одной сессии, могут быть считаны и/или удалены в другой сессии ПУ, инициировавшим данную задачу. Данные по

завершенной задаче доступны между сессиями по тому же идентификатору задачи. При получении задачи «запрос загрузки данных» (DataLoadRequest) от ПУ, не являющегося инициатором данной задачи, ТС ОРМ посылают ответ на «запрос загрузки данных», в котором указывается отсутствие результата исполнения задачи (data-exists), а в поле «краткое описание ошибки» (error-description) записывается расшифровка отказа в доступе к данным задачи. Далее ТС ОРМ посылают на этот ПУ «сигнал» попытки несанкционированного доступа (unauthorized-access) и ожидают его подтверждения.

1.31. ТС ОРМ производят уничтожение данных, сформированных в результате выполнения задачи и самой выполненной задачи после поступления с ПУ запроса на удаление данных.

1.32. В случае возникновения в ТС ОРМ или каналах передачи данных исключительных ситуаций на ПУ передаются «Сообщения» типа «Сигнал».

Приложение № 3 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

Требования, предъявляемые к функционированию канала кпд 1

1.1. Диаграмма состояний переходов ТС ОРМ по кпд 1 приведена на

1.1. Рисунок **1**.

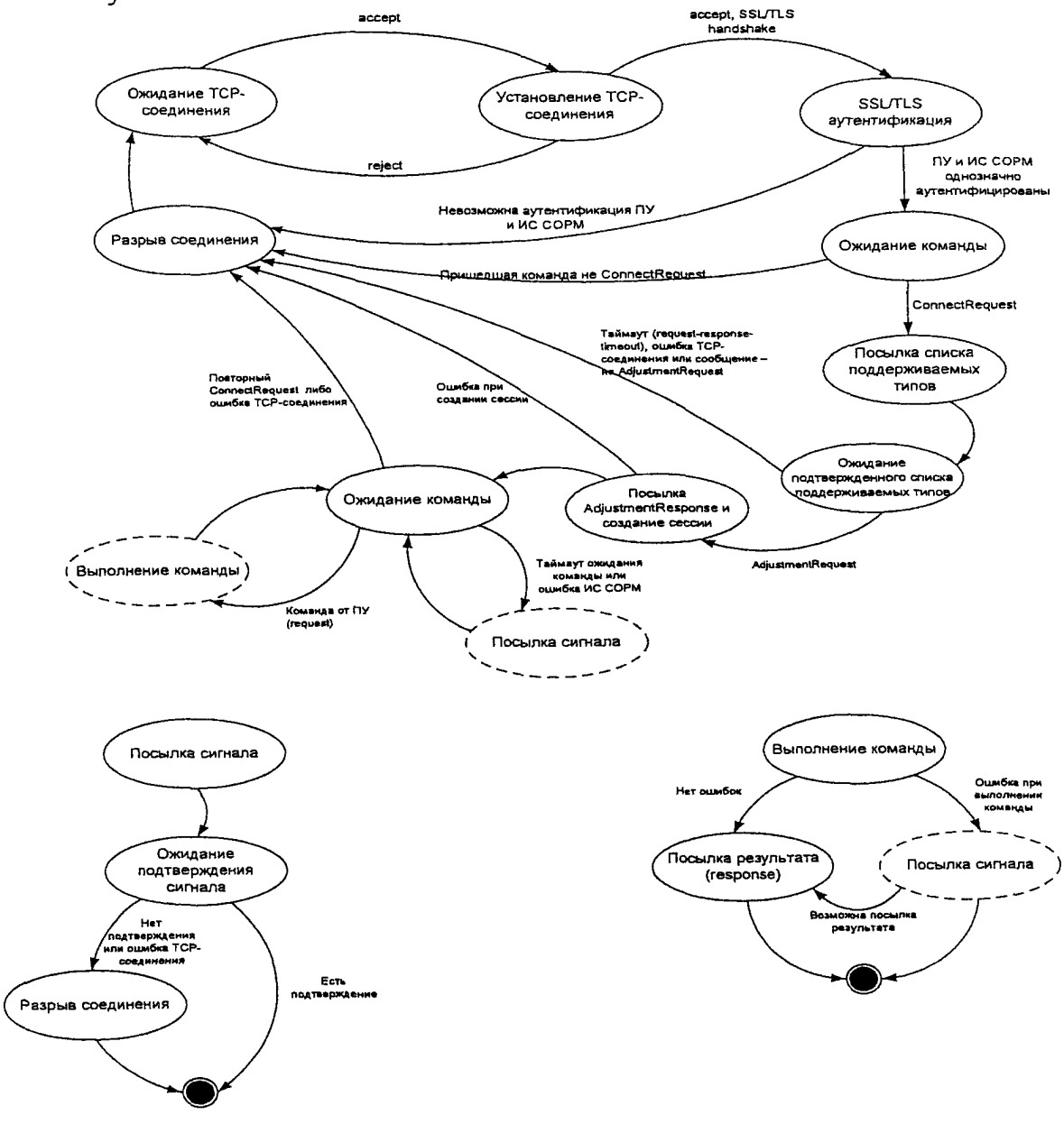

Рисунок 1. Диаграмма состояний перехода ТС ОРМ по кпд 1.

ТС ОРМ по TCP-порту кпд 1 ожидают входящих соединений. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

1.2. Если SSL/TLS-соединения по каналам управления и данных установлены, а сессия не открыта, ТС ОРМ должны реагировать только на сообщение «Запрос на открытие сессии». При попытке посылок каких-либо других сообщений со стороны ПУ ТС ОРМ должны разорвать TCP соединения по каналу управления и каналу данных и перевести канальный уровень подключения в исходное состояние.

1.3. При получении сообщения «Запрос на открытие сессии» ТС ОРМ создают список поддерживаемых ТС ОРМ типов запросов, отчетов и сигналов (в том числе и предыдущих версий) и отсылают его.

1.4. После отсылки списка поддерживаемых типов ТС ОРМ ожидают от ПУ списка поддерживаемых им запросов, отчетов и сигналов. Список поддерживаемых ПУ типов должен являться подмножеством списка типов в ТС ОРМ.

1.5. При получении от ПУ списка поддерживаемых ПУ запросов ТС ОРМ посылают сообщение «Ответ на согласование списка поддерживаемых типов» и создает сессию.

1.6. После создания сессии кпд 1 ТС ОРМ переводятся в режим ожидания команд от ПУ. При поступлении команды со стороны ПУ производится ее выполнение и формируется результат. Результат в виде сообщения «ответа» отправляется на ПУ.

1.7. В случае серьезного сбоя в ТС ОРМ, вызванного причинами, не предусмотренными режимом нормального функционирования системы, по каналу управления передаётся «сигнал» – «Критическая ошибка ПО, потеря данных, дальнейшая работа невозможна», с соответствующим описанием проблемы. Все задачи, которые были в процессе выполнения, когда произошел сбой, а также данные выполненных задач, поврежденные в результате произошедшего сбоя, имеют «признак результата выполнения задачи» (TaskResult) равный значению «ошибка»(error) с соответствующим описанием проблемы. В случае, если для восстановления работоспособности ТС ОРМ требуется их перезагрузка, то по каналу управления выдается прерывание «Перезапуск ПО». В этом случае ТС ОРМ и ПУ закрывают все открытые на текущий момент сессии.

1.8. В случае наличия признаков сбоя или ошибки выполнения конкретной задачи ТС ОРМ в режиме нормального функционирования по каналу управления передаётся прерывание «Серьезная ошибка ПО, потеря данных, но дальнейшая работа возможна», с соответствующим описанием проблемы. Результат выполнения данной задачи имеет «признак результата выполнения задачи» (TaskResult), равный значению «ошибка»(error), с соответствующим описанием проблемы.

1.9. На каждый «сигнал», переданный ТС ОРМ, ПУ должно ответить сообщением «подтверждение сигнала» по кпд 1. Отсутствие подтверждения в течение времени RequestResponseTimeout, которое задается при открытии сессии,

свидетельствует о прерывании соединения и должно вызывать действия, описанные в п. 1.23 настоящего Приложения.

1.10. При отсутствии команд ПУ в течение «максимального времени неактивности» (session-timeout, задается в ConnectRequest), ТС ОРМ посылают на ПУ «сигнал» Heartbeat и ожидает его подтверждения аналогично описанному в п. 1.8 настоящего Приложения.

1.11. Диаграмма состояний переходов ПУ по кпд 1 приведена на 1 настоящего Приложения.

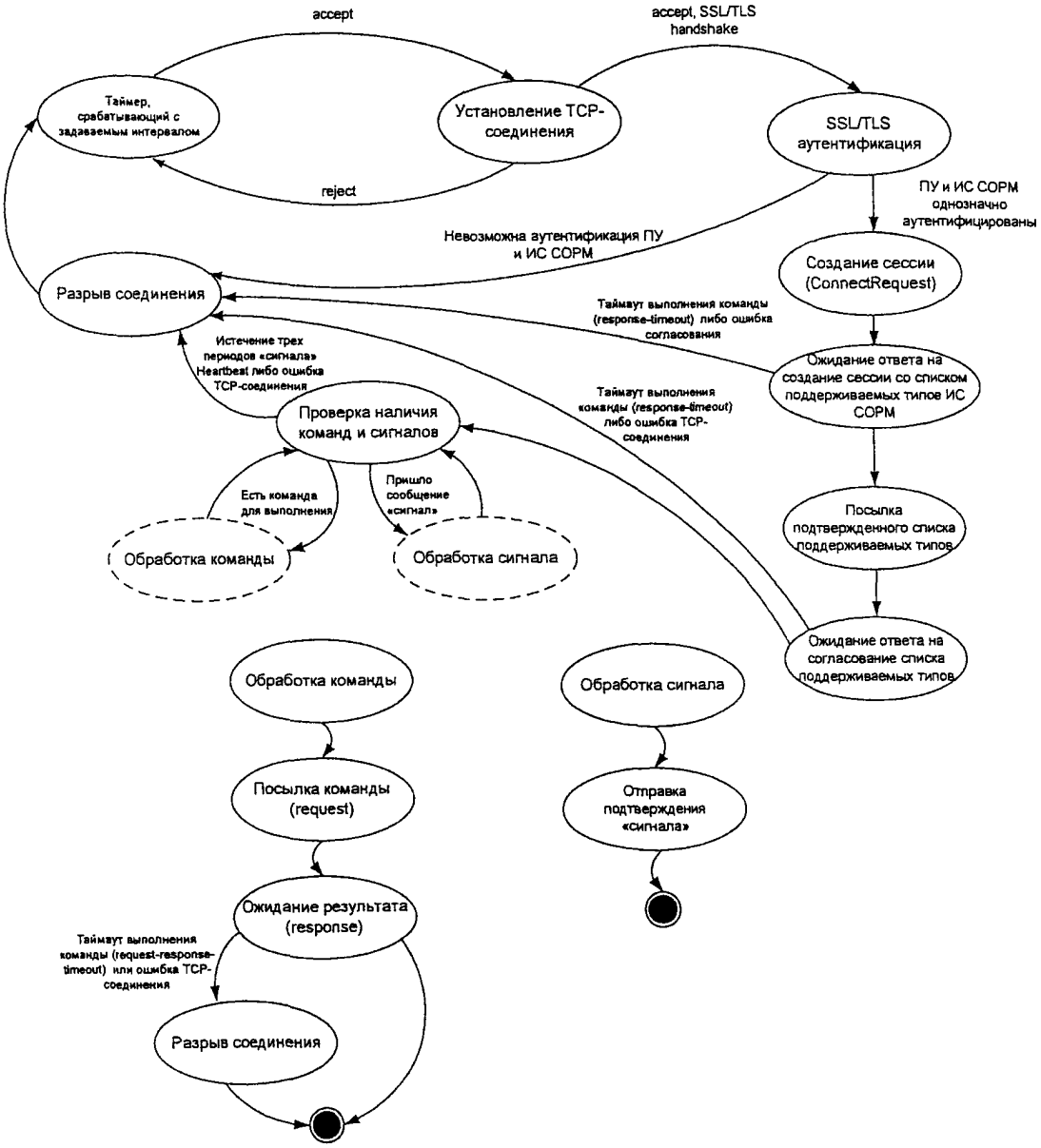

Рисунок 1. Диаграмма состояний перехода ПУ по кпд 1.

1.12. ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ТС ОРМ по заданному порту кпд 1. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

1.13. ПУ ожидает установления соединения по кпд 2.

1.14. После установления TCP-соединения по кпд 2 и прохождения по нему взаимной аутентификации ТС ОРМ и ПУ, ПУ по кпд 1 посылает команду создания сессии (ConnectRequest).

1.15. ПУ ожидает сообщения «ответ» от ТС ОРМ на отправленную команду в течение времени «таймаут ответа на запрос» (request-response-timeout).

1.16. Если сообщение не получено в течение времени «таймаут ответа на запрос», ПУ разрывает соединения с ТС ОРМ по кпд 1 и кпд 2 и переводит их в изначальное состояние (п. 1.12). Время ожидания сообщения «ответ» не зависит от приема сообщений «сигналов», поступающих в этот интервал времени.

1.17. При получении сообщения «Ответ на запрос создания сессии» ПУ создает список поддерживаемых ПУ типов запросов, отчетов и сигналов и отправляет его ТС ОРМ. Список поддерживаемых ПУ типов должен быть подмножеством списка типов ТС ОРМ, полученных в п. 1.15 настоящего Приложения.

1.18. После отправки сообщения «Согласование поддерживаемых типов со стороны ПУ» ПУ ждет ответа на сообщение от ТС ОРМ. Поведение ПУ при ожидании ответа аналогично описанному в п. 1.16 настоящего Приложения.

1.19. Если во время ожидания сообщения «ответ» ТС ОРМ посылают на ПУ сообщение «сигнал» (в т.ч. Heartbeat), то ПУ посылает «подтверждение» о принятии «сигнала» (в т.ч. Heartbeat) и продолжает ожидать сообщение «ответ» на отосланную команду.

1.20. После создания сессии ПУ посылает поступающие команды на ТС ОРМ и ожидает сообщений «ответов» на них в соответствии с п. 1.15, 1.16, 1.19 настоящего Приложения.

1.21. Если при ожидании поступления в ПУ команд ТС ОРМ не посылала «сигнал» Heartbeat в течение трех интервалов «максимального времени неактивности» (session-timeout, задается в ConnectRequest) ПУ разрывает соединения с ТС ОРМ по кпд 1 и кпд 2 и переводит их в изначальное состояние (п. 1.12 настоящего Приложения).

1.22. При поступлении команды «запрос на закрытие сессии» (DisconnectRequest) ПУ отсылает ее на ТС ОРМ, ожидает сообщения «ответ» в соответствии с п. 1.15, 1.16, 1.19 настоящего Приложения, после чего разрывает соединение по кпд 2 и кпд 1 и переводит их в изначальное состояние (п. 1.12 настоящего Приложения).

Приложение № 4 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

Требования, предъявляемые к функционированию канала кпд 2

1.1. ТС ОРМ обеспечивают подключение ПУ и обработку поступающих запросов по каналу кпд 2 в соответствии с приведенными диаграммами состояний переходов.

Диаграмма состояний переходов ТС ОРМ по кпд 2 приведена на

1.2. Рисунок Приложения 4.

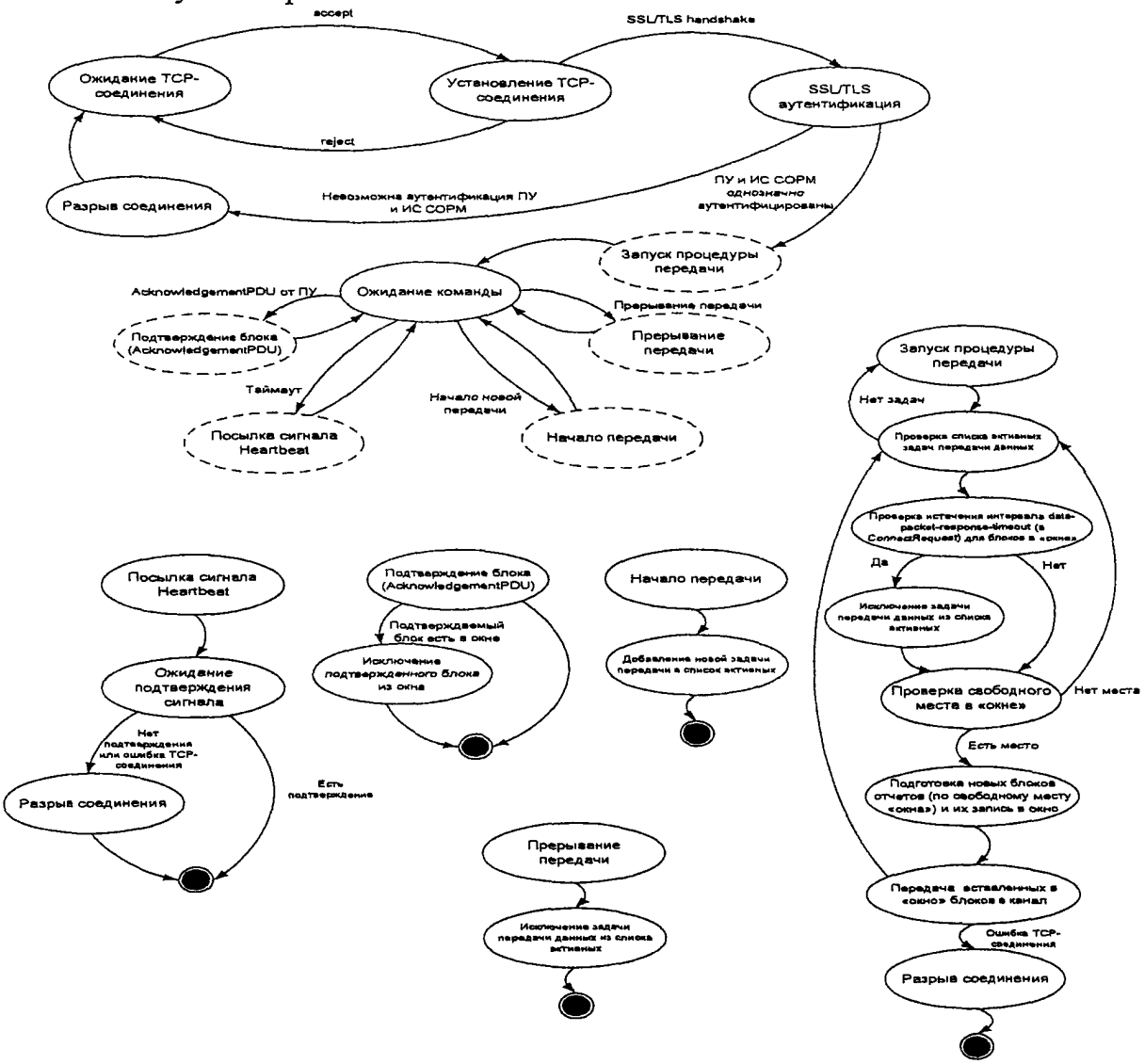

Рисунок 1. Диаграмма состояний перехода ТС ОРМ по кпд 2.

1.3. ТС ОРМ по TCP-порту кпд 2 ожидают входящих соединений. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

1.4. Если на ТС ОРМ была передана «запрос загрузки данных» (DataLoadRequest), ТС ОРМ посылают «ответ на запрос загрузки данных» (DataLoadResponse) по каналу кпд 1 и начинает передачу данных блоков отчетов по кпд 2 при их наличии. ПУ может получить блоки отчетов по кпд 2 до получения «ответа на запрос загрузки данных» (DataLoadResponse) по кпд 1.

1.5. Если количество переданных без получения «подтверждения» о принятии серии блоков «отчетов» по всем задачам, по которым выполняется загрузка на ПУ данных, меньше «окна канала передачи данных» (параметр data-packet-window-size в запросе создания сессии ConnectRequest), то ТС ОРМ выполняют подготовку новых блоков отчетов по загружаемым задачам и посылают их на ПУ. Количество подготовленных и переданных без получения «подтверждения» блоков не должно превышать размера окна канала передачи данных.

1.6. Максимальная задержка подтверждения приема блока данных со стороны ПУ не превышает параметр «таймаут подтверждения приема блока данных отчета» (data-packet-response-timeout), указываемый при создании сессии. Если задержка подтверждения превысила заданное значение, то оставшиеся для передачи блоки данных не отправляются и по каналу управления передается «сигнал» «Незначительная ошибка ПО, данные не потеряны, дальнейшая работа возможна» с соответствующим описанием проблемы, при этом в поле «reference-message» сообщения «сигнал» указывается идентификатор сообщения блока отчета, по которому не поступило подтверждение приема.

1.7. При получении «подтверждения» блока «отчета» ТС ОРМ должны записать информацию об ошибочно принятом ПУ блоке и ошибочных записях в блоке в специальный журнал, передача последующих блоков по задаче на ПУ не прерывается. ТС ОРМ должны предоставлять техническому персоналу оператора связи доступ к журналу с записями об ошибочно принятых на ПУ блоках отчетов и средства исправления ошибочных данных в отчетах. Подтвержденные блоки исключаются из окна канала передачи данных (в окне канала передачи данных остаются только неподтвержденные блоки).

1.8. В случае разрыва TCP/IP соединения кпд 2 при существующем соединении кпд 1 по кпд 1 передаётся прерывание «Незначительная ошибка ПО, данные не потеряны, дальнейшая работа возможна» с соответствующим описанием проблемы, работа в данном случае не прекращается, выполняется установление соединения по кпд 2 в соответствии с п. 1.20 Приложения 2.

1.9. Передача блоков данных может быть прервана в случае получения ТС ОРМ «запроса прерывания загрузки данных».

1.10. Если по кпд 2 не производится передача блоков отчетов в течение «максимального времени неактивности» (session-timeout при создании сессии ConnectRequest), ТС ОРМ посылают на ПУ «сигнал» Heartbeat и ожидает его подтверждения аналогично описанному в п. 1.9 Приложения 3.

1.11. Диаграмма состояний переходов ПУ по кпд 2 приведена на Рисунок Приложения № 4.

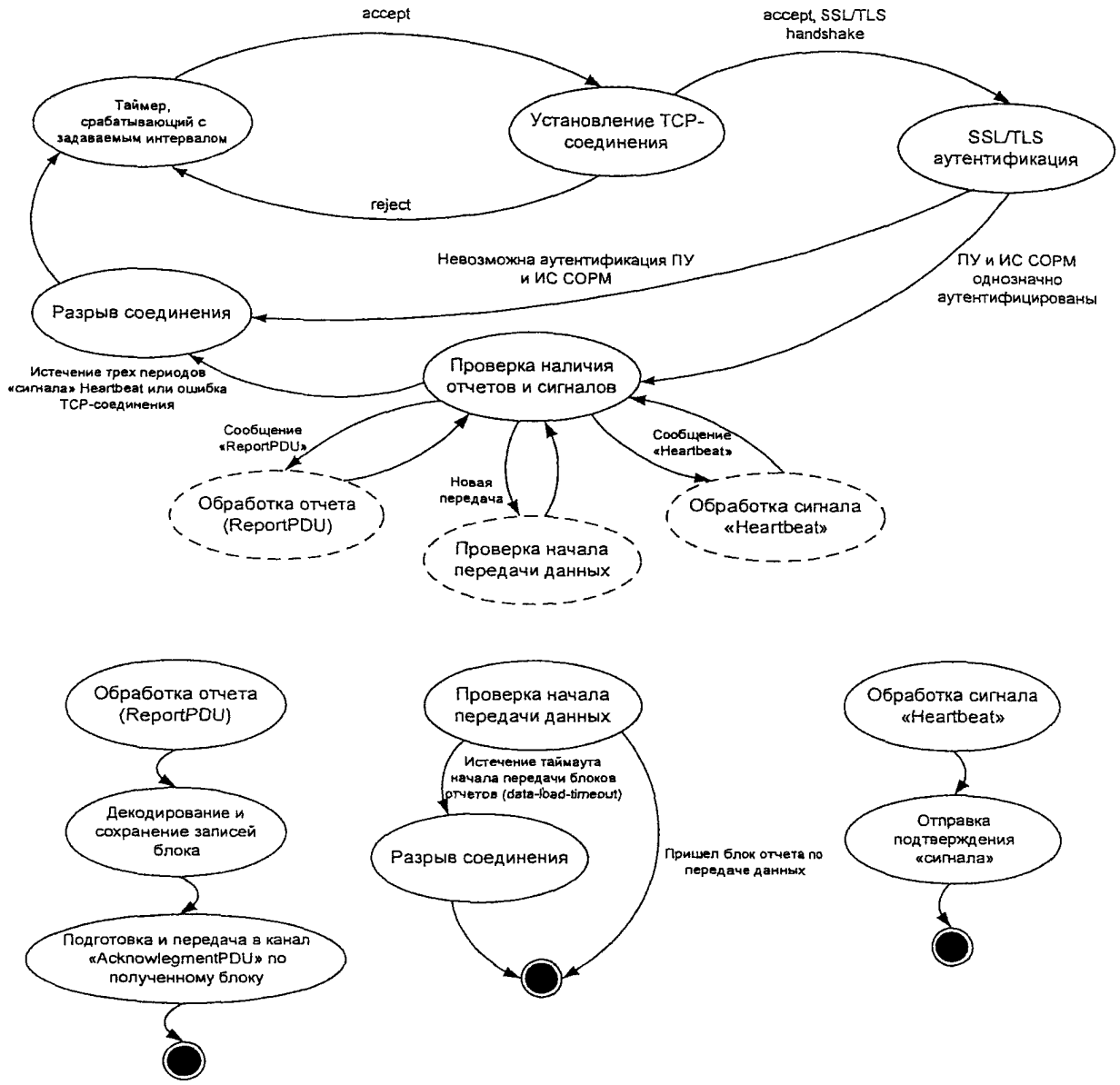

Рисунок 2. Диаграмма состояний перехода ПУ по кпд 2.

1.12. ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ТС ОРМ по заданному порту кпд 2. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

1.13. При поступлении «запроса загрузки данных конкретной задачи» (DataLoadRequest) ПУ ожидает начала передачи данных в течение времени «таймаут начала передачи блоков отчетов» (data-load-timeout в ConnectRequest). Если данные не поступают в течение вышеописанного периода, то ПУ разрывает соединения по каналам кпд 1 и кпд 2 и переводит соединения в изначальное состояние (п. 1.12 Приложения 3, п. 1.12 настоящего Приложения).

1.14. При поступлении блока отчета ПУ производит декодирование полученного блока и сохранение декодированных данных.

1.15. В ответ на переданный блок данных ПУ посылает сообщение «подтверждение» получения блока отчета. Количество последовательно переданных ТС ОРМ блоков данных без подтверждения со стороны ПУ определяется параметром «размер окна канала передачи данных», который согласовывается при создании сессии. При подтверждении блока отчета ПУ может сигнализировать об ошибке декодирования блока. В этом случае в сообщении «подтверждение» приема для ошибочно декодированного блока ПУ, в случае возможности, указывает номер записи в блоке, начиная с которой декодирование не удалось.

1.16. Если при ожидании поступления в ПУ блоков отчетов ТС ОРМ не посылала «сигнал» Heartbeat в течение трех интервалов «максимального времени неактивности» (session-timeout, задается в ConnectRequest), ПУ разрывает соединения с ТС ОРМ по кпд 1 и кпд 2 и переводит их в изначальное состояние (п. 1.12 Приложения № 3, п. 1.12 настоящего Приложения).

Приложение № 5 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

Требования, предъявляемые к функционированию канала кпд 3

1.1. ТС ОРМ обеспечивают подключение ПУ и обработку поступающих запросов по каналу кпд 3 (канал мониторинга).

1.2. ТС ОРМ обеспечивают прием и обработку следующих видов запросов от ПУ по каналу кпд3 (канал мониторинга):

— запрос на получение структуры ТС ОРМ – КТС и списка модулей прикладного ПО ТС ОРМ («GetStructureRequest»);

— запрос на получение конфигурации КТС/модуля ПО ТС ОРМ («GetModuleConfigRequest»);

— запрос на изменение конфигурации КТС/модуля ПО ТС ОРМ («SetModuleConfigRequest»);

— запрос на получение состояния модуля КТС/модуля ПО ТС ОРМ («CheckModuleRequest»).

1.3. На каждый запрос по кпд3 ТС ОРМ посылает ответ на ПУ, содержащий результат обработки соответствующего запроса – «ManagementResponse».

1.4. Диаграммы состояний перехода ТС ОРМ и ПУ по кпд 3 соответствуют диаграммам для кпд 1 (

1.5. Рисунок **1**, Рисунок Приложения № 3).

1.6. ТС ОРМ по TCP-порту кпд 3 ожидают входящих соединений. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

1.7. Если SSL/TLS-соединение по кпд 3 установлено, а сессия не открыта, ТС ОРМ должны реагировать только на сообщение «Запрос на открытие сессии». Создание сессии аналогично описанному в пп. 1.2-1.4 Приложения № 3. При попытке посылок каких-либо других сообщений со стороны ПУ ТС ОРМ должны разорвать TCP соединение по кпд 3 и перевести канальный уровень подключения в исходное состояние.

1.8. После создания сессии ТС ОРМ переводятся в режим ожидания команд от ПУ. Обработка поступающих команд и посылка сигналов производится аналогично описанному в пп. 1.5-1.9 Приложения № 3.

1.9. ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ТС ОРМ по заданному порту. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

1.10. ПУ по кпд 3 посылает команду создания сессии (ConnectRequest).

1.11. Ожидание подтверждения создания сессии, согласование списка поддерживаемых типов, отправка команд, ожидание ответов и обработка полученных от ТС ОРМ сигналов по кпд 3 производится ПУ аналогично описанному в пп. 1.15-1.21 Приложения № 3.

1.12. ТС ОРМ должны обеспечивать получение следующих видов информации о структуре и функционировании ТС ОРМ по запросу ПУ:

— о структуре и составе КТС ТС ОРМ, составе и состоянии интерфейсов взаимодействия КТС ТС ОРМ;

— об установленном на КТС ТС ОРМ общесистемном и специальном программном обеспечении ТС ОРМ, перечне и состоянии программных модулей в составе специального программного обеспечения ТС ОРМ;

— о точках подключения ТС ОРМ к сети оператора связи и интерфейсах ввода информации в ТС ОРМ.

1.13. В части структуры и состава КТС ТС ОРМ, состава и состоянии интерфейсов взаимодействия КТС ТС ОРМ, ТС ОРМ должны по запросу ПУ предоставлять следующую информацию:

— перечень коммутационного и серверного оборудования, средств хранения данных с его идентификацией;

— идентификацию интерфейсов подключения КТС ТС ОРМ друг к другу;

— предоставление следующих параметров для серверного оборудования (на момент формирования ПУ запроса):

— общий и занятый объем оперативной памяти;

— количество сетевых интерфейсов с их идентификацией, текущая нагрузка;

— общее количество процессоров, текущая загрузка;

— общий объем дискового пространства, объем свободного пространства;

— предоставление следующих параметров о технических средствах хранения данных:

— перечень модулей, составляющих средства хранения данных с их идентификацией;

— для каждого входящего в состав средств хранения данных модуля:

— общий объем дискового пространства;

— объем свободного дискового пространства;

— текущее состояние модуля (штатное функционирование, сбой, не функционирует), текстовую расшифровку сбоя.

1.14. В части точек подключения ТС ОРМ к сети оператора связи, интерфейсов ввода информации в ТС ОРМ, ТС ОРМ должны по запросу ПУ предоставлять текущую информацию на момент формирования запроса, содержащую:

— перечень точек подключения к сети связи и точек ввода информации в ТС ОРМ с их идентификацией;

— для каждой точки подключения предоставлять информацию:

— вид поступающих в ТС ОРМ сведений (об абонентах, об оказанных услугах связи);

— состояние точки (штатное функционирование, сбой, не функционирует), текстовую расшифровку сбоя;

— сведения об объеме поступающей информации в секунду, в т.ч.: количество записей, объем (байт);

— период, в течение которого на точку подключения/ввода информации в ТС ОРМ не поступала информация.

1.15. В части состава общесистемного и специального программного обеспечения ТС ОРМ, их текущего состояния, ТС ОРМ по запросу ПУ следующую информацию:

— перечень установленного общесистемного программного обеспечения с его идентификацией;

— предоставление для общесистемного программного обеспечения информации:

— идентификатора КТС, на котором установлены ТС ОРМ;

— наименование общесистемного программного обеспечения;

— текущее состояние (штатное функционирование, сбой, не функционирует), текстовую расшифровку сбоя;

— перечень установленного специального программного обеспечения ТС ОРМ с его идентификацией;

— предоставление для специального программного обеспечения ТС ОРМ информации:

— идентификатора КТС, на котором установлены ТС ОРМ;

— назначение (определяется разработчиком ТС ОРМ);

— текущее состояние (штатное функционирование, сбой, не функционирует), текстовую расшифровку сбоя;

— список контролируемых параметров (определяется разработчиком ТС ОРМ).

1.16. ТС ОРМ должны предоставлять возможность изменения отдельных параметров функционирования КТС ТС ОРМ общесистемного и специального программного обеспечения по запросу ПУ посредством кпд 3.

Приложение № 6 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

Требования, предъявляемые к функционированию канала кпд 4

1.1. ТС ОРМ обеспечивают подключение ПУ и обработку поступающих запросов по каналу кпд 4 (канал неформатированных данных).

1.2. ТС ОРМ обеспечивает прием и обработку следующих видов запросов от ПУ по каналу кпд 4:

— «запрос проверки наличия вида неформатированных данных в ТС ОРМ» (DataTypesRequest);

— «запрос на начало передачи неформатированных данных» (DataStartRequest);

— «запрос на остановку передачи неформатированных данных» (DataStopRequest).

1.3. На каждый запрос по кпд 4 ТС ОРМ посылают ответ на ПУ, содержащий результат обработки соответствующего запроса.

1.4. ТС ОРМ должны накапливать информацию с неформатированными данными в специальном буфере. Данные в буфере упорядочиваются согласно времени их поступления. Длительность временного хранения информации в буфере – 10 суток.

1.5. ТС ОРМ должны записывать в буфер информацию об оказанных услугах связи, абонентах, с которыми заключены договоры на оказание услуг почтовой связи и справочные данные об операторах (филиалах оператора почтовой связи), отделениях почтовой связи и типах документов, удостоверяющих личность, в виде текстовых файлов.

1.6. ТС ОРМ должны записывать информацию в буфер циклически, т.е. при переполнении буфера старая информация в нем перезаписывается.

1.7. Диаграмма состояний переходов ТС ОРМ по кпд 4 приведена на Рисунок Приложения № 6.

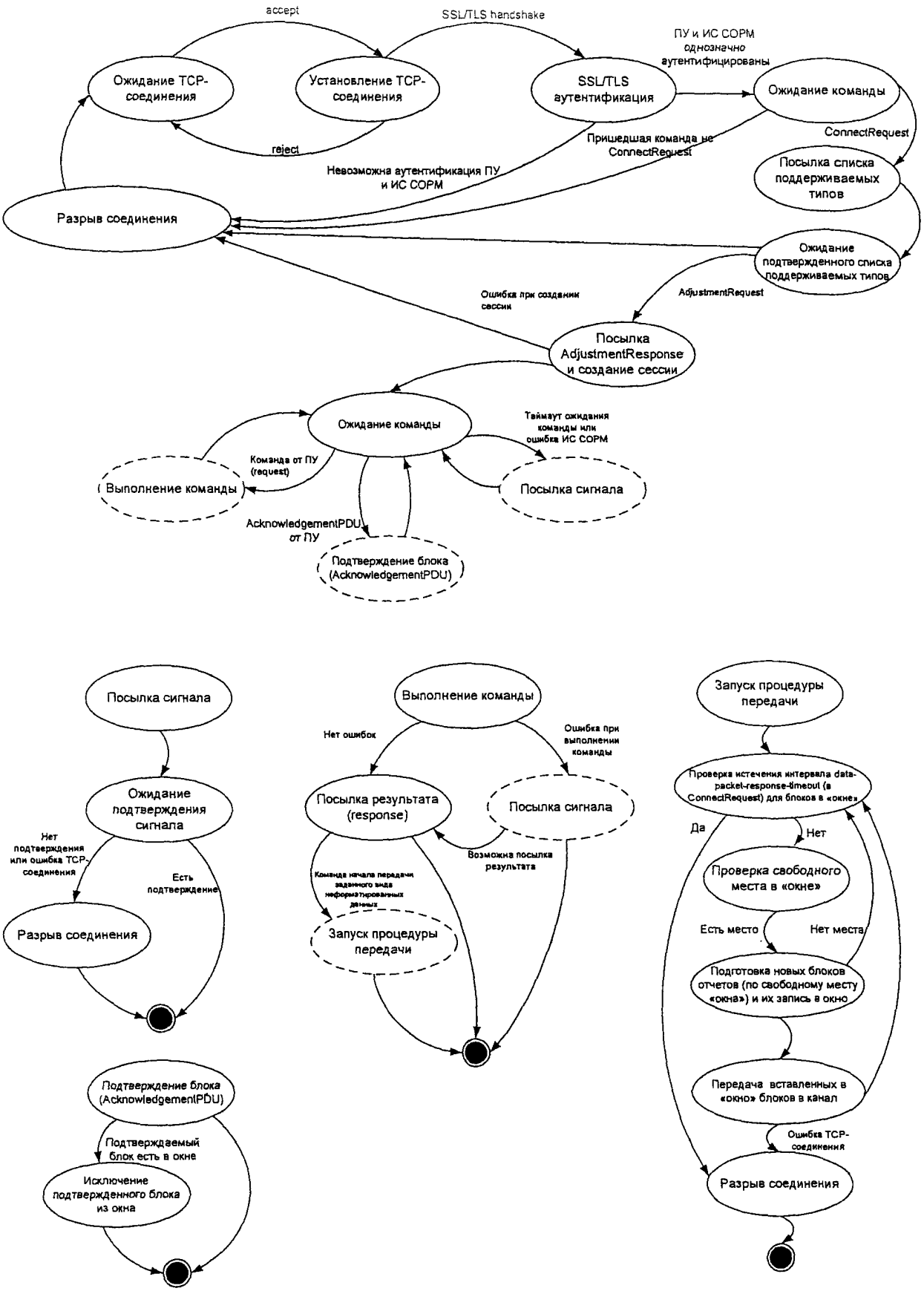

Рисунок 1. Диаграмма состояний перехода ТС ОРМ по кпд 4.

1.8. ТС ОРМ ожидают и устанавливают соединение аналогично описанному в пп. 1.5, 1.6 Приложения № 5.

1.9. После создания сессии ТС ОРМ переводятся в режим ожидания команд. Обработка команд и посылка «сигналов» осуществляется аналогично описанному в пп. 1.5 - 1.9 Приложения № 3, за исключением команд «запрос на начало, остановку передачи неформатированных данных» (DataStartRequest/ DataStopRequest).

1.10. При приеме команды «запрос на начало передачи неформатированных данных» ТС ОРМ, находясь в режиме ожидания команд, отсылает «ответ на запрос начала передачи неформатированных данных» и переводит канал кпд 4 в режим передачи данных.

1.11. При приеме команды «запрос на остановку передачи неформатированных данных» ТС ОРМ, находясь в режиме ожидания команд, посылает «ответ на запрос остановки передачи неформатированных данных» с отрицательным результатом.

1.12. В режиме передачи данных посылка блоков отчетов и их подтверждение производится аналогично описанному в пп. 1.5, 1.7, 1.10 Приложения № 4.

1.13. Максимальная задержка подтверждения приема блока данных со стороны ПУ не превышает параметр «таймаут подтверждения приема блока данных отчета» (data-packet-response-timeout), указываемый при создании сессии. Если задержка подтверждения превысила заданное значение, то оставшиеся для передачи блоки данных не отправляются, ТС ОРМ разрывают соединение по кпд 4 и переводит кпд 4 в изначальное состояние. Передача неформатированных данных соответствующего типа производится из буфера кпд 4 (см. п. 1.5 настоящего Приложения).

1.14. При приеме команды «запрос на остановку передачи неформатированных данных» ТС ОРМ, находясь в режиме передачи данных, посылает «ответ на запрос остановки передачи неформатированных данных», завершает посылку блоков данных и переводится в режим ожидания команд.

1.15. При приеме команды «запрос на начало передачи неформатированных данных» ТС ОРМ, находясь в режиме передачи данных, посылает «ответ на запрос начала передачи неформатированных данных» с отрицательным результатом, при этом передача неформатированных данных не прекращается.

1.16. Исходные данные о соединениях в виде неформатированных данных записываются в буфер независимо от текущего режима работы канала кпд 4 ТС ОРМ.

1.17. Если команда «запрос на остановку передачи неформатированных данных» поступила в момент передачи из буфера файловых данных, то передаваемый файл должен сохраняться в буфере и быть доступным для передачи после посылки команды «запрос на начало передачи неформатированных данных» с учетом ограничений по длительности хранения, указанным в п. 1.5.

1.18. Диаграмма состояний переходов ПУ по кпд 4.

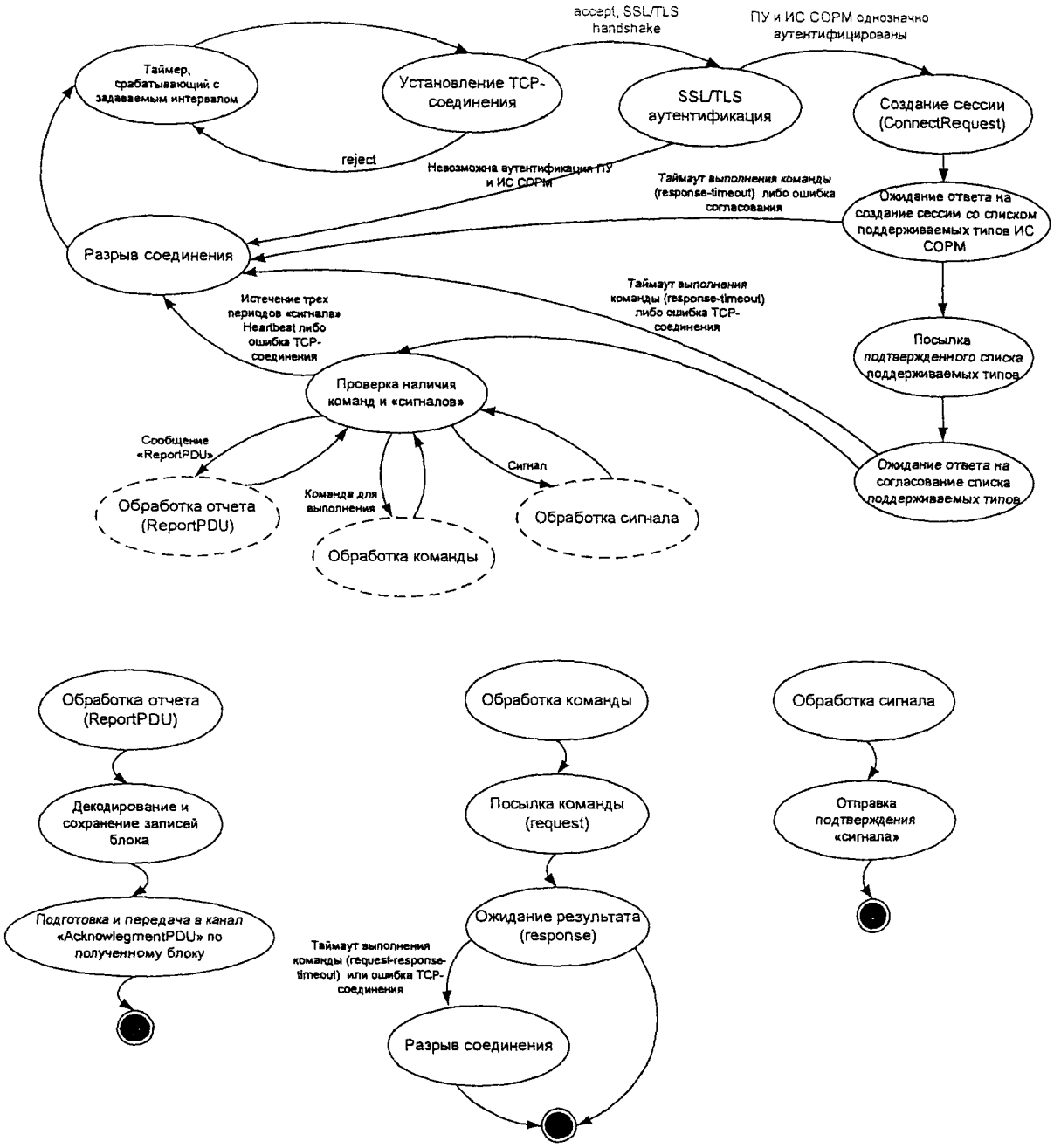

Рисунок 2. Диаграмма состояний перехода ПУ по кпд 4.

1.19. ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ТС ОРМ по заданному порту. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

1.20. ПУ по кпд 4 посылает команду создания сессии (ConnectRequest).

1.21. Ожидание подтверждения создания сессии, согласование списка поддерживаемых типов, отправка команд, ожидание ответов и обработка полученных от ТС ОРМ сигналов по кпд 4 производится ПУ аналогично описанному в пп. 1.15 – 1.21 Приложения № 3, за исключением команд «запрос на начало, остановку передачи неформатированных данных» (DataStartRequest/ DataStopRequest).

1.22. При посылке команды «запрос на начало/остановку передачи неформатированных данных», ПУ ожидает результата (DataStartResponse) аналогично описанному в пп. 1.15 – 1.16 Приложения 3, после чего переводит канал кпд 4 в режим передачи данных.

1.23. В режиме передачи данных ПУ производит прием, декодирование и подтверждение приема данных аналогично описанному в пп. 1.14 – 1.15 Приложения 4.

1.24. При приеме команды «запрос на остановку передачи неформатированных данных» (DataStopRequest) ПУ, находясь в режиме передачи данных, завершает прием блоков данных и переводится в режим передачи команд.

Приложение № 7 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

Требования к форматам сообщений ТС ОРМ

1.1. Структура разделения ASN.1 модулей протокола взаимодействия ПУ и ТС ОРМ приведена на Рисунке 4 Приложения № 7.

1.2. Модуль Classification.asn задает правила, в соответствии с которыми:

— выполняется расширение:

— списка типов запросов к ТС ОРМ;

— списка видов поисковых критериев к ТС ОРМ;

— списка типов отчетов, формируемых ТС ОРМ;

— выполняется ввод новых версий сообщений протокола.

1.3. Структура разделения ASN.1 «Сообщений» протокола приведена на Рисунок .

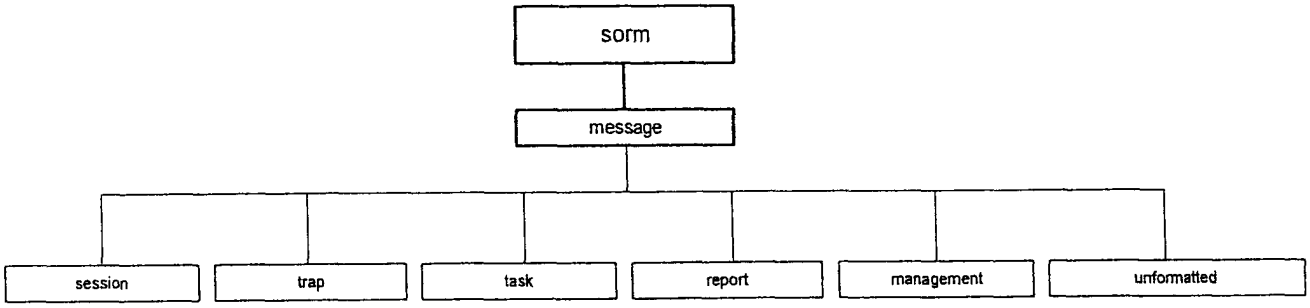

Рисунок 1. Структура видов сообщений протокола взаимодействия ПУ и ТС ОРМ.

1.4. В соответствии с Рисунок , Рисунок в протокольных ASN.1 модулях выполняется подстановка соответствующих версий форматов поисковых критериев, отчетов, справочников, «сообщений» интерфейса взаимодействия ПУ и ТС ОРМ.

1.5. ASN.1–модуль «Classification.asn» содержит кодированные в иерархическом виде идентификаторы:

— видов «Сообщений» верхнего уровня интерфейса взаимодействия ПУ и ТС ОРМ, составляющих кпд 1, кпд 2, кпд 3, кпд 4;

— видов поисковых критериев для формирования задач к ТС ОРМ;

— видов форматов отчетов, формируемых ТС ОРМ.

1.6. Соответствующие идентификаторы используются в других ASN.1-модулях интерфейса взаимодействия ПУ и ТС ОРМ (Рисунок настоящего Приложения с модулями), при этом идентификатор определяет конкретную версию и расширения формата соответствующего элемента (поисковых критериев, отчетов, справочников «сообщений» – в соответствии с Рисунок , Рисунком 3 настоящего Приложения).

1.7. Предоставленный ТС ОРМ при создании сессии перечень идентификаторов и согласованное из него ПУ подмножество в целом определяют конкретные возможности взаимодействия ПУ и ТС ОРМ в соответствии с выбранными идентификаторами.

1.8. Расширение интерфейса взаимодействия ПУ и ТС ОРМ обеспечивается введением новых идентификаторов, определяющих соответствующие расширенные элементы (поисковые критерии, отчеты, справочники, «сообщения»). Кодирование новых вводимых идентификаторов элементов осуществляется в соответствии со структурами на Рисунок , Рисунок настоящего Приложения и стандартным кодированием ASN.1-модуля «Classification.asn».

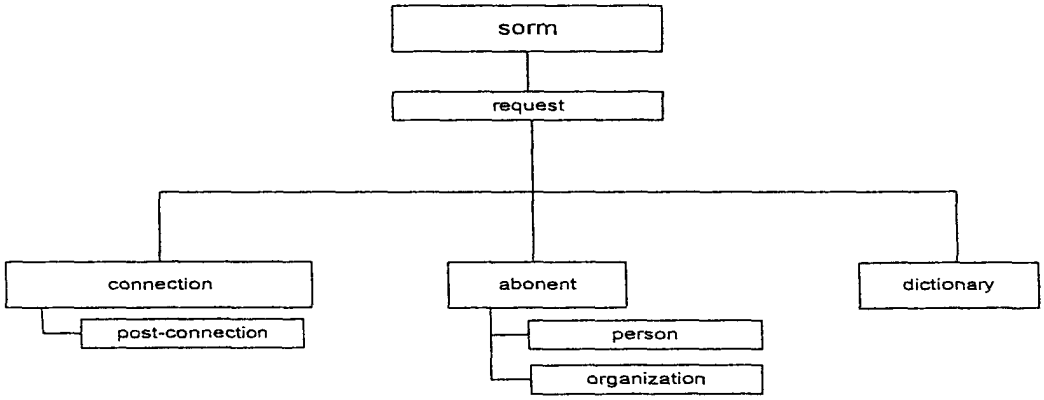

Рисунок 2. Структура разделения поисковых критериев кпд 1.

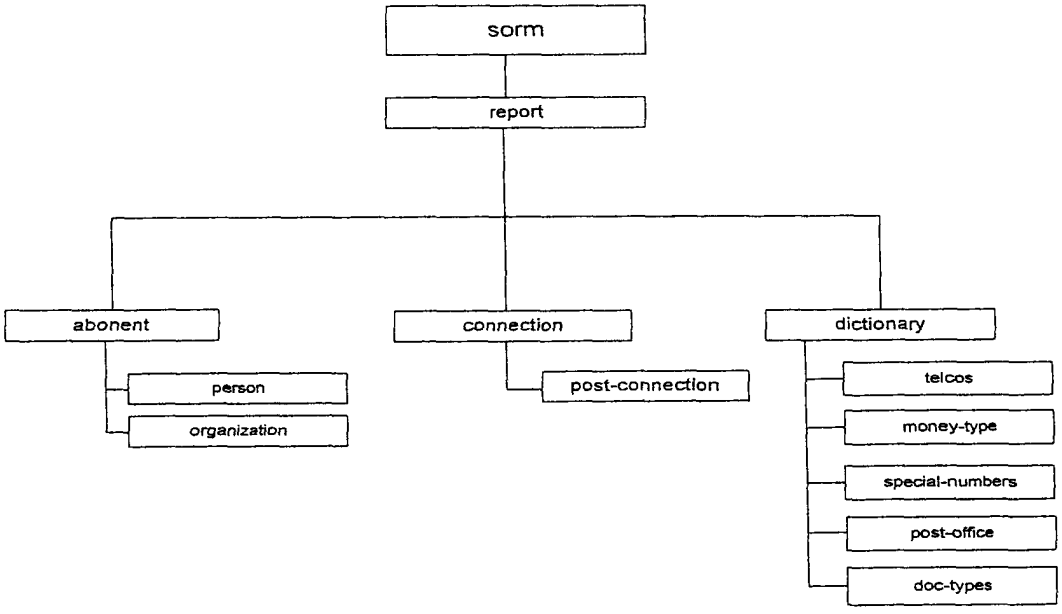

Рисунок 3. Структура разделения видов отчетов кпд 2.

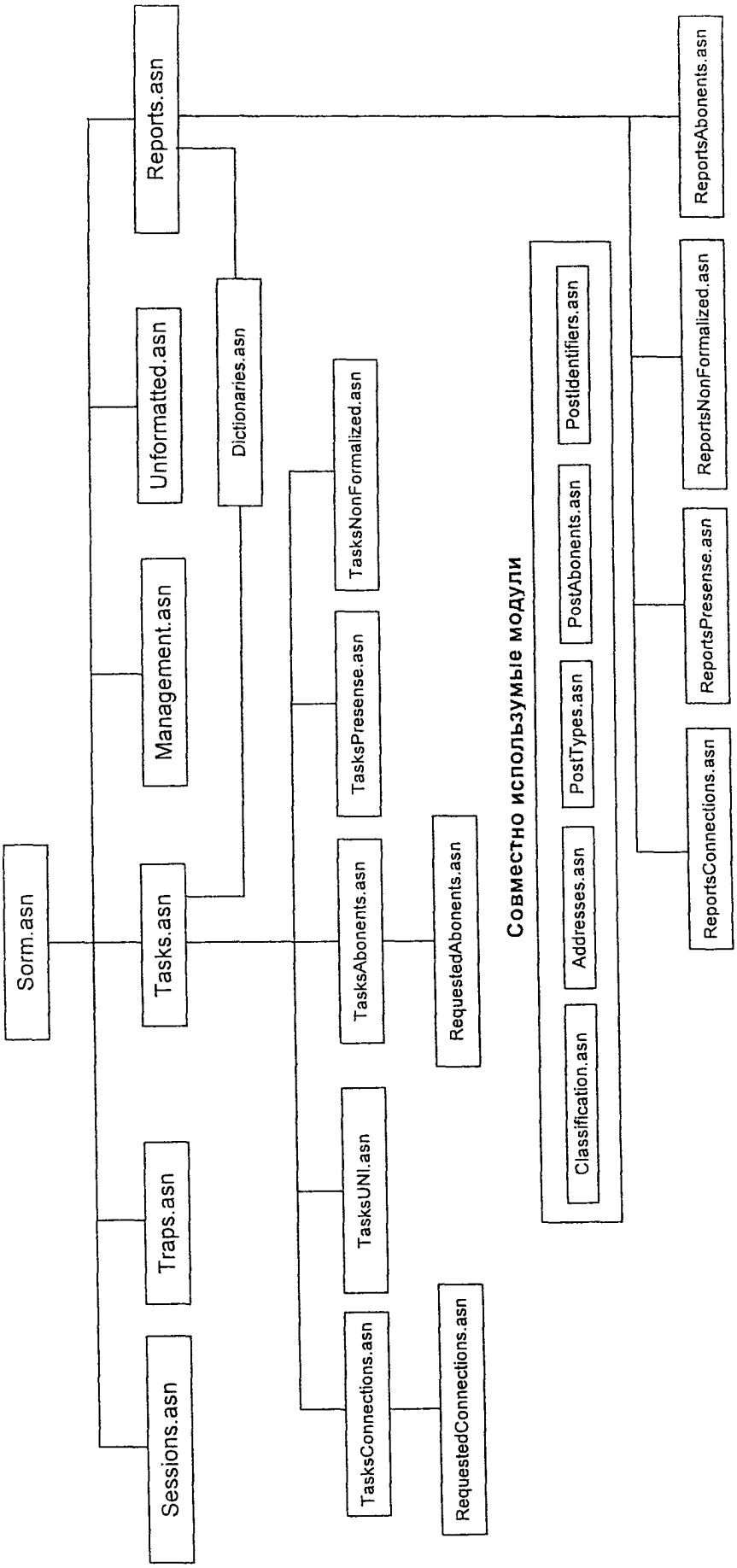

Рисунок 4. Структура ASN.1-модулей интерфейса взаимодействия ПУ и ТС ОРМ.

Приложение № 8 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

ASN.1-спецификация протокола взаимодействия ПУ и ТС ОРМ

```
Addresses.asn

Addresses DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS AddressType,
        ReportedAddresses,
        ReportedAddress,
        RequestedAddress;

AddressType ::= ENUMERATED {
        physical (0),     --- физический
        legal (1),        --- юридический
        delivery (2),     --- доставки счета
        receive (3),      --- приезда курьера
        common (4),       --- общий
       other (5)        --- другой
}
ReportedAddresses ::= SEQUENCE OF ReportedAddress

ReportedAddress ::= SEQUENCE {
  address-info AddressInfoReport,                    --- адрес
  title      AddressType OPTIONAL                        --- тип адреса
}
AddressInfoReport ::= CHOICE {
  struct-info[0]   AddressStructInfoReport,                  ---
структурированный адрес
```

```
 unstruct-info[1] UTF8String(SIZE (1 .. 1024))                    ---
неструктурированный адрес
}
AddressStructInfoReport ::= SEQUENCE {
 zip [0]      UTF8String (SIZE (1 .. 32))  OPTIONAL,                  --- почтовый
индекс, zip-код
 country [1]   UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- страна
 region [2]    UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- область
 zone [3]      UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- район,
муниципальный округ
 city [4]      UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- город,
поселок, деревня
 street [5]    UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- улица
 building [6]  UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- дом,
строение
 build-sect [7] UTF8String (SIZE (1 .. 128)) OPTIONAL,                --- корпус
 apartment [8]  UTF8String (SIZE (1 .. 128)) OPTIONAL                 --- квартира,
офис
}
RequestedAddress ::= SEQUENCE {
 address-info RequestedAddressInfo,                                --- адрес
 title      AddressType  OPTIONAL                                  --- тип адреса
}
RequestedAddressInfo ::= CHOICE {
 struct-info[0]   RequestedAddressStructInfo,                      ---
структурированный адрес
 unstruct-info[1] UTF8String(SIZE (1 .. 1024))                     ---
неструктурированный адрес
}
RequestedAddressStructInfo ::= SEQUENCE {
 country [0]   UTF8String (SIZE (1 .. 128)),                       --- страна
 region [1]    UTF8String (SIZE (1 .. 128)) OPTIONAL,                  --- область
 zone [2]      UTF8String (SIZE (1 .. 128)) OPTIONAL,                  --- район,
муниципальный округ
 city [3]      UTF8String (SIZE (1 .. 128)) OPTIONAL,                  --- город,
поселок, деревня
 street [4]    UTF8String (SIZE (1 .. 128)) OPTIONAL,                  --- улица
 building [5]  UTF8String (SIZE (1 .. 128)) OPTIONAL,                  --- дом,
строение
 build-sect [6] UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- корпус
 apartment [7]  UTF8String (SIZE (1 .. 128)) OPTIONAL,                 --- квартира,
офис
 zip [8]       UTF8String (SIZE (1 .. 32)) OPTIONAL                  --- почтовый
индекс, zip-код
```

```
}
END
```

Classification.asn

```
Classification DEFINITIONS IMPLICIT TAGS ::=
BEGIN
EXPORTS TAGGED,
        sorm-message-session,
        sorm-message-trap,
        sorm-message-task,
        sorm-message-report,
        sorm-message-management,
        sorm-message-unformatted,
        sorm-post-abonent-report,

        sorm-post-office,
        sorm-post-office-identifier,

        sorm-report-post-connection,
        sorm-request-post-connection,
        sorm-post-abonent-mailing,

        sorm-requested-post-abonent-person-mailing,
        sorm-requested-post-abonent-organization-mailing,
        sorm-reported-post-abonent-person-mailing,
        sorm-reported-post-abonent-organization-mailing,

        sorm-post-abonent,
        sorm-post-abonent-person,
        sorm-post-abonent-organization,

        sorm-post-type,
        sorm-post-type-package,
        sorm-post-type-money-order,
        sorm-post-type-secogram,
        sorm-post-type-postal-wrapper,
          sorm-post-type-postcard,
          sorm-post-type-ems,
          sorm-post-type-small-packet,
        sorm-post-type-bag-international,

        sorm-request-dictionaries,
        sorm-report-dictionary-telcos,
```

```
        sorm-report-dictionary-post-offices,
          sorm-report-dictionary-doc-types,

        sorm-request-presense,
        sorm-report-presense-abonents,
        sorm-report-presense-connections,
        sorm-report-presense-dictionaries,

        sorm-request-abonent,
        sorm-request-contract,
        sorm-requested-abonent-organization,
        sorm-requested-abonent-person,

        sorm-report-abonent-abonent,
        sorm-report-abonent-person,
        sorm-report-abonent-organization
    ;
TAGGED ::= CLASS {
    &id ObjectDescriptor UNIQUE,
    &Data
}
WITH SYNTAX {
    OID &id
    DATA &Data
}
--- Классификация
OID ::= PrintableString

-- Подструктура сообщений
sorm-message-session OID ::= "280"
sorm-message-trap OID ::= "6"
sorm-message-task OID ::= "282"
sorm-message-report OID ::= "7"
sorm-message-management OID ::= "284"
sorm-message-unformatted OID ::= "285"

-- Параметры почтовых отправлений
sorm-report-post-connection OID ::= "1"

-- Идентификаторы
sorm-request-post-connection OID ::= "21"
sorm-post-abonent-mailing OID ::= "22"
sorm-requested-post-abonent-person-mailing OID ::= "23"
sorm-requested-post-abonent-organization-mailing OID ::= "24"
```

```
sorm-reported-post-abonent-person-mailing OID ::= "25"
sorm-reported-post-abonent-organization-mailing OID ::= "26"

-- Абоненты
sorm-request-abonent OID ::= "180"
sorm-request-contract OID ::= "181"
sorm-requested-abonent-organization OID ::= "182"
sorm-requested-abonent-person OID ::= "183"

sorm-report-abonent-abonent OID ::= "30"
sorm-report-abonent-person OID ::= "32"
sorm-report-abonent-organization OID ::= "33"

-- Пользователи услуг почтовой связи
sorm-post-abonent OID ::= "2"
sorm-post-abonent-report OID ::= "3"
sorm-post-abonent-person OID ::= "4"
sorm-post-abonent-organization OID ::= "5"

-- Почтовые идентификаторы
sorm-post-office OID ::= "8"
sorm-post-office-identifier OID ::= "9"

-- Виды почтовых отправлений
sorm-post-type OID ::= "60"
sorm-post-type-package OID ::= "61"
sorm-post-type-money-order OID ::= "62"
sorm-post-type-mail OID ::= "63"
sorm-post-type-secogram OID ::= "64"
sorm-post-type-postal-wrapper OID ::= "65"
sorm-post-type-postcard OID ::= "66"
sorm-post-type-ems OID ::= "67"
sorm-post-type-small-packet OID ::= "68"
sorm-post-type-bag-international OID ::= "69"

-- Справочники
sorm-request-dictionaries OID ::= "110"
sorm-report-dictionary-telcos OID ::= "111"
sorm-report-dictionary-money-types OID ::= "112"
sorm-report-dictionary-post-offices OID ::= "113"
sorm-report-dictionary-doc-types OID ::= "114"

-- Запрос о наличии данных
sorm-request-presense OID ::= "260"
```

```
sorm-report-presense-abonents OID ::= "120"
sorm-report-presense-connections OID ::= "121"
sorm-report-presense-dictionaries OID ::= "123"

END
```

Dictionaries.asn

```
Dictionaries DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS
  TelcoID,
  TelcoList,
  MoneyType,
  DictionaryTask,
  DictionaryReport;

IMPORTS DateAndTime
  FROM Sorm
      TAGGED,
      sorm-request-dictionaries,
      sorm-report-dictionary-telcos,
    sorm-report-dictionary-post-offices
  FROM Classification

      ReportedAddress
  FROM Addresses;

-- Идентификатор оператора связи или филиала
TelcoID ::= INTEGER (0 .. 65535)

--- список идентификаторов операторов связи или филиалов
TelcoList ::= SEQUENCE OF TelcoID

-- Запрос
DictionaryTask ::= SEQUENCE {
 id TAGGED.&id ({DictionaryTaskVariants}),
 data TAGGED.&Data ({DictionaryTaskVariants}{@id})
}

DictionaryTaskVariants TAGGED ::= { dictionaryTask }
```

```
dictionaryTask TAGGED ::= {
 OID { sorm-request-dictionaries }
 DATA ObjectDescriptor                                    -- тип запрашиваемого
справочника (идентификатор отчёта)
}

-- ObjectDescriptor принимает значение одно из:
-- sorm-report-dictionary-telcos
-- sorm-report-dictionary-post-offices
-- sorm-report-dictionary-doc-types

-- Отчёт
DictionaryReport ::= SEQUENCE {
 id      TAGGED.&id ({DictionaryRecordsVariants}),                    ---
идентификтор записи справочника
 data  TAGGED.&Data({DictionaryRecordsVariants}{@id})          --- данные
записи справочника
}

DictionaryRecordsVariants TAGGED ::= {
     telcosRecords                                          --- операторы связи,
обслуживаемые ТС ОРМ
   | postOfficeRecords                                  --- узлы (отделения) почтовой
связи
   | docTypesRecords                                    --- типы документов,
удостоверяющие личность
}

--- операторы связи, обслуживаемые ТС ОРМ
telcosRecords TAGGED ::= {
 OID { sorm-report-dictionary-telcos }
 DATA SEQUENCE OF TelcosRecord }

TelcosRecord ::= SEQUENCE {
 telco-id      TelcoID,                                  --- номер филиала или
оператора связи
 begin-time    DateAndTime,                                --- время начала
действия
 end-time      DateAndTime OPTIONAL,                                --- время
конца действия
 description   UTF8String (SIZE (1 .. 256))                  --- описание
(наименование) оператора связи или филиала
}
```

```
--- размер платежа
MoneyType ::= SEQUENCE {
   value    REAL,                    --- сумма платежа
   type-id  PrintableString (SIZE (1 .. 3))        --- код валюты по ISO 4217
}

--- справочник узлов (отделений) почтовой связи
postOfficeRecords TAGGED ::= {
 OID { sorm-report-dictionary-post-offices }
 DATA SEQUENCE OF PostOfficeRecord  }

PostOfficeRecord ::= SEQUENCE {
   telco-id        TelcoID,                              --- номер филиала или
оператора связи
   index-post      PrintableString (SIZE (6 .. 10)) OPTIONAL,      --- индекс
узла/отделения почтовой связи
   office-type ENUMERATED {                    --- тип узла/отделения
почтовой связи
      international-point (0),                 --- международное место почтового
обмена
      customs (1),                             --- место таможенного досмотра
      regional-centre (2),                     --- региональный центр
      simple (3)                               --- узел почтовой связи
   },
   begin-time         DateAndTime,                                  --- время начала
действия
   end-time [0]       DateAndTime OPTIONAL,                                  ---
время конца действия
   full-name [1]      UTF8String (SIZE (1 .. 128)) OPTIONAL, --- полное
наименование отделения почтовой связи
   office-address [2]  ReportedAddress OPTIONAL,          --- адрес
узла/отделения почтовой связи
   contact [3]        UTF8String (SIZE (1 .. 128)) OPTIONAL, --- контактные
телефоны
   public-point-data [4] UTF8String (SIZE (1 .. 128)) OPTIONAL  --- информация
о пункте коллективного доступа, расположенного в узле/отделении почтовой
связи
}

docTypesRecords TAGGED ::= {
 OID {sorm-report-dictionary-doc-types}
 DATA SEQUENCE OF DocTypesRecord
}
```

```
DocTypesRecord ::= SEQUENCE {
 telco-id    TelcoID,                                  --- идентификатор
оператора связи или филиала
 doc-type-id INTEGER (1 .. 65535),                     --- идентификатор
типа документа
 begin-time  DateAndTime,                              --- время начала
действия
 end-time    DateAndTime OPTIONAL,                     --- время конца
действия
 description UTF8String (SIZE (1 .. 256))              --- описание
(наименование)
}
END
```

Management.asn

```
Management DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS managementMessage;

IMPORTS TAGGED,
        sorm-message-management
        FROM Classification;

managementMessage TAGGED ::= {
   OID { sorm-message-management }
   DATA CHOICE {
     request [0] ManagementRequest,
     response [1] ManagementResponse
   }
}

--- тип сообщения "команда управления ТС ОРМ"
ManagementRequest ::= CHOICE {
 get-structure [0]     GetStructureRequest,            --- запрос на получение
структуры ТС ОРМ - КТС и модулей СПО
 get-module-config [1] GetModuleConfigRequest,         --- запрос на
получение конфигурации КТС/модуля СПО
 set-module-config [2] SetModuleConfigRequest,         --- запрос на
изменение конфигурации КТС/модуля СПО
 check-module [3]      CheckModuleRequest,             --- запрос на получение
состояния модуля
```

```
 get-module-types [4]  GetModuleTypesRequest                    --- запрос на
получение типов модулей КТС и СПО
}

--- запрос на получение структуры ТС ОРМ - КТС и модулей СПО
GetStructureRequest ::= NULL

--- запрос на получение конфигурации КТС/модуля СПО
GetModuleConfigRequest::= CHOICE {
 hw-modules-list [0] RequestedHardwareModules,                --- перечень
идентификаторов узлов КТС ТС ОРМ
 sw-modules-list [1] RequestedSoftwareModules                 --- перечень
идентификаторов модулей СПО ТС ОРМ
}

RequestedHardwareModules ::= SEQUENCE OF ModuleId            --- перечень
идентификаторов узлов КТС ТС ОРМ
RequestedSoftwareModules ::= SEQUENCE OF ModuleId            --- перечень
идентификаторов модулей СПО ТС ОРМ

--- запрос на изменение конфигурации КТС/модуля СПО
SetModuleConfigRequest::= SEQUENCE {
 module-id    ModuleId,                                  --- идентификатор
конфигурируемого модуля
 module-config ConfiguratedModule                        --- устанавливаемая в
модуль конфигурация
}

ConfiguratedModule ::= CHOICE {
 sw-module [0] SormSoftwareModule,                       --- для узла КТС
 hw-module [1] SormHardwareModule                        --- для узла СПО
}

--- запрос на получение состояния модуля
CheckModuleRequest::= RequestedModulesList

RequestedModulesList ::= CHOICE {
 sw-modules [0] RequestedHardwareModules,                --- идентификаторы
узлов КТС, для которых запрашивается состояние
 hw-modules [1] RequestedSoftwareModules                 --- идентификаторы
модулей ТС ОРМ, для которых запрашивается состояние
}

--- запрос на получение типов модулей КТС и СПО
```

```
GetModuleTypesRequest ::= NULL

--- уникальный идентификатор КТС/модуля СПО ТС ОРМ
ModuleId ::= OCTET STRING (SIZE (8))

--- параметр модуля
ModuleParameter ::= SEQUENCE {
 parameter-name  UTF8String (SIZE (1 .. 256)),                    --- наименование
параметра
 read-only      BOOLEAN,                                  --- контролируемый или
измеряемый параметр
 parameter-value ParameterValue                           --- значение параметра
}

--- варианты значений параметров
ParameterValue ::= CHOICE {
 string [0]  UTF8String (SIZE (1 .. 256)),
 integer [1] INTEGER (0 .. 999999999),
 boolean [2] BOOLEAN
}

ModuleParameters ::= SEQUENCE OF ModuleParameter

SormSoftwareModule ::= SEQUENCE {
 module-id         ModuleId,                               --- уникальный
идентификатор данного модуля
 hardware-module-id ModuleId,                                  --- идентификатор КТС,
на котором работат данный блок модуля СПО
 block-name        INTEGER (0 .. 1024),                    --- номер блока СПО модуля
 module-name       UTF8String (SIZE (1 .. 512)),                --- наименование
модуля
 module-type         INTEGER (1 .. 512),                   --- идентификатор типа
модуля
 module-parameters  ModuleParameters,                             --- список
параметров модуля
 sub-modules-list   SubmodulesList OPTIONAL                       --- субмодули
}
SubmodulesList ::= SEQUENCE OF SormSoftwareModule

SormHardwareModule ::= SEQUENCE {
 module-id         ModuleId,                               --- уникальный
идентификатор данного модуля
 block-name        INTEGER (0 .. 1024),                    --- номер блока КТС
```

```
 module-name      UTF8String (SIZE (1 .. 512)),              --- наименование
модуля
 module-parameters HwParameterGroups                         --- значение
группы параметров КТС
}

HwParameterGroup ::= SEQUENCE {
 group-name       UTF8String (SIZE (1 .. 512)),        --- наименование группы
параметров для КТС
 module-parameters ModuleParameters                          --- перечень параметров
для КТС
}

HwParameterGroups ::= SEQUENCE OF HwParameterGroup

SormSoftwareModules ::= SEQUENCE OF SormSoftwareModule

SormHardwareModules ::= SEQUENCE OF SormHardwareModule

--- тип сообщения "ответ на команду управления ТС ОРМ"
ManagementResponse ::= CHOICE {
 get-structure [0]    GetStructureResponse,                  --- ответ на запрос
получения структуры ТС ОРМ - КТС и модулей СПО
 get-module-config [1] GetModuleConfigResponse,              --- ответ на запрос
получения конфигурации КТС/модуля СПО
 set-module-config [2] SetModuleConfigResponse,              --- ответ на запрос
изменения конфигурации КТС/модуля СПО
 check-module [3]     CheckModuleResponse,                   --- ответ на запрос
получения состояния модуля
 get-module-types [4]  GetModuleTypesResponse                --- ответ на запрос
получения типов модулей КТС и СПО
}

--- ответ на запрос получения структуры ТС ОРМ - КТС и модулей СПО
GetStructureResponse ::= SEQUENCE {
 sw-modules SormHardwareModules,                             --- перечень всех узлов
КТС
 hw-modules SormSoftwareModules                              --- перечень всех
модулей СПО ТС ОРМ
}

--- ответ на запрос получения конфигурации КТС/модуля СПО
GetModuleConfigResponse ::= SEQUENCE {
```

```
 sw-modules SormHardwareModules,                    --- конфигурации
запрошенных узлов КТС
 hw-modules SormSoftwareModules                     --- конфигурации
запрошенных модулей СПО ТС ОРМ
}

--- отчет на запрос изменения конфигурации КТС/модуля СПО
SetModuleConfigResponse::= ConfiguratedModule              --- установленная
в модуль конфигурация

--- ответ на запрос получения состояния модуля
CheckModuleResponse ::= CHOICE {
 hw-modules [0] SormHardwareModules,                       --- текущее
состояние запрошенных модулей СПО ТС ОРМ
 sw-modules [1] SormSoftwareModules                 --- текущее состояние
запрошенных узлов КТС
}

--- ответ на запрос получения типов модулей КТС и СПО
GetModuleTypesResponse ::= SEQUENCE OF ModuleType

ModuleType ::= SEQUENCE {
 module-type     INTEGER (1 .. 512),                --- идентификатор типа
модуля
 type-description UTF8String ( SIZE (1 .. 128))     --- расшифровка типа модуля
}
END
```

PostAbonents.asn

```
PostAbonents DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS PostContractAbonent,
        RequestedPostMailingAbonent,
       ReportedPostMailingAbonent,
       PassportInfoReport,
       PersonNameInfo;

IMPORTS TAGGED,
         sorm-post-abonent,
         sorm-post-abonent-person,
         sorm-post-abonent-organization,
```

```
        sorm-post-abonent-mailing,
        sorm-requested-post-abonent-person-mailing,
        sorm-requested-post-abonent-organization-mailing,
        sorm-reported-post-abonent-person-mailing,
        sorm-reported-post-abonent-organization-mailing

    FROM Classification
        DateAndTime
    FROM Sorm
        ReportedAddress
    FROM Addresses
        TelcoID
    FROM Dictionaries
        ;

PostContractAbonent ::= SEQUENCE {
 id     TAGGED.&id ({AbonentsContractData}),
 data   TAGGED.&Data({AbonentsContractData}{@id})
}

AbonentsContractData TAGGED ::= {
        postContractAbonent
}

postContractAbonent TAGGED ::= {
 OID sorm-post-abonent
 DATA PostAbonentContractRecord
}

PostAbonentContractRecord ::= SEQUENCE {
 telco-id        TelcoID,                                --- идентификатор оператора
связи или филиала
 abonent         AbonentContractData,                    --- информация об
абоненте (клиенте оператора связи)
 contract-data   PostContract OPTIONAL                   --- информация о
заключенном договоре (в случае юридических лиц, предоставляется только для
принадлежности)
}

PostContract ::= SEQUENCE {
 begin-date    DateAndTime,                              --- дата и время заключения
договора
 contract      UTF8String (SIZE (1 .. 64)),              --- номер договора
```

```
 end-date [0]  DateAndTime OPTIONAL,                    --- дата и время
расторжения договора

 bank [1]        UTF8String (SIZE(1 .. 256)) OPTIONAL,  --- банк абонента
(используемый при расчетах с оператором связи)
 bank-account [2]  UTF8String (SIZE(1 .. 512)) OPTIONAL,    --- счет абонента в
банке (используемый при расчетах с оператором связи)
 corr-account [3]  UTF8String (SIZE(1 .. 512)) OPTIONAL,    ---
корреспондентский счет
 kpp [4]         UTF8String (SIZE(9 .. 64)) OPTIONAL,    --- КПП
 bik [5]         UTF8String (SIZE(9 .. 64)) OPTIONAL,    --- БИК

 contacts [6]     SEQUENCE OF ContactInfo OPTIONAL,      --- контактные
лица
 addresses [7]    SEQUENCE OF PostContractAddress OPTIONAL  --- адреса по
договору
}

PostContractAddress ::= SEQUENCE {
 address ReportedAddress,                          --- адрес, указанный в контракте
 index-post PrintableString (SIZE (1 .. 10)) OPTIONAL       --- почтовый индекс
адреса
}

--- контактная информация по договору
ContactInfo ::= SEQUENCE {
 contact       PersonNameInfo,                          --- ФИО
отправителя/контактное лицо
 phone-list [0]  SEQUENCE OF UTF8String (SIZE (1 .. 32)) OPTIONAL,   ---
список контактных телефонов
 email-list [1]  SEQUENCE OF UTF8String (SIZE (1 .. 128)) OPTIONAL   ---
список е-мейл адресов
}

AbonentContractData ::= SEQUENCE {
 id     TAGGED.&id ({AbonentsContractDataVariants}),
 data   TAGGED.&Data({AbonentsContractDataVariants}{@id})
}

AbonentsContractDataVariants TAGGED ::= {
 abonentContractPerson |                                 --- информация об
абоненте - физическом лице
 abonentContractOrganization                              ---
информация об абоненте - юридическом лице
```

```
}

--- Пользователь услуг почтовой связи - физическое лицо
abonentContractPerson TAGGED ::= {
 OID { sorm-post-abonent-person }
 DATA AbonentContractPerson
}

AbonentContractPerson ::= SEQUENCE {
 name-info       PersonNameInfo,                    --- ФИО
 birth-date      GeneralizedTime OPTIONAL,          --- дата рождения
 passport-info   PassportInfoReport OPTIONAL,       ---
паспортные данные
 phone-fax       UTF8String (SIZE (1 .. 128)) OPTIONAL    --- контактные
телефоны, факс
}

PersonNameInfo ::= CHOICE {
 struct-name[0]  PersonStructNameInfoReport,        ---
структурированное ФИО
 unstruct-name[1] UTF8String(SIZE(1 .. 1024))       ---
неструктурированное ФИО
}

PersonStructNameInfoReport ::= SEQUENCE {
 given-name  UTF8String (SIZE (1 .. 128)),          --- имя
 initial     UTF8String (SIZE (1 .. 128)),          --- отчество (при
наличии)
 family-name UTF8String (SIZE (1 .. 128))           --- фамилия
}


PassportInfoReport::= SEQUENCE {
 ident-card-info IdentCardInfoReport,               --- описание
удостоверения личности
 doc-type-id INTEGER (0 .. 65535) OPTIONAL          ---
идентификатор типа документа, удостоверяющего личность
}

IdentCardInfoReport ::=      CHOICE {
 struct-info [0]   IdentCardStructInfoReport,       ---
структурированная информация
 unstruct-info [1]  UTF8String (SIZE (1 .. 512))    ---
неструктурированная информация
```

```
}

IdentCardStructInfoReport ::= SEQUENCE {
 ident-card-serial     UTF8String (SIZE (1..16)),                    --- серия
удостоверения личности
 ident-card-number     NumericString (SIZE (1..16)),                 --- номер
удостоверения личности
 ident-card-description UTF8String (SIZE (1 .. 256))                 --- когда и кем
выдано
}

abonentContractOrganization TAGGED ::= {
 OID { sorm-post-abonent-organization }
 DATA AbonentContractOrganization
}

AbonentContractOrganization ::= SEQUENCE {
 full-name        UTF8String (SIZE (1 .. 256)),                      --- полное
наименование
 inn [0]          NumericString (SIZE ( 10 .. 12 )) OPTIONAL,        --- ИНН
 ogrn [1]         UTF8String (SIZE(1 .. 32)) OPTIONAL,               --- ОГРН
 okpo [2]         UTF8String (SIZE(1 .. 32)) OPTIONAL,               --- ОКПО
 index-post [3]   PrintableString (SIZE (1 .. 10)) OPTIONAL          --- почтовый
индекс
}

--- поисковые критерии в информации об абоненте отправителе/получателе
почтового отправления
RequestedPostMailingAbonent ::= SEQUENCE {
 id     TAGGED.&id ({RequestedAbonentsMailingData}),
 data   TAGGED.&Data({RequestedAbonentsMailingData}{@id})
}

RequestedAbonentsMailingData TAGGED ::= {
 requestedMailingPerson |                                            --- информация об
абоненте - физическом лице
 requestedMailingOrganization                                        ---
информация об абоненте - юридическом лице
}

requestedMailingPerson TAGGED ::= {
 OID { sorm-requested-post-abonent-person-mailing }
 DATA RequestedAbonentMailingPerson
}
```

```
requestedMailingOrganization TAGGED ::= {
 OID { sorm-requested-post-abonent-organization-mailing }
 DATA RequestedAbonentMailingOrganization
}

RequestedAbonentMailingPerson ::= SEQUENCE {
 name-info [0]     PersonNameInfo OPTIONAL,                    --- ФИО
 birth-date [1]    GeneralizedTime OPTIONAL,                   --- дата
рождения
 passport-info [2]  PassportInfoReport OPTIONAL,               --- паспортные
данные
 phone-fax [3]     UTF8String (SIZE (1 .. 128)) OPTIONAL,      ---
контактные телефоны, факс
 address [4]          RequestedAddress OPTIONAL                --- контактный
адрес
}

RequestedAbonentMailingOrganization ::= SEQUENCE {
 full-name [0]    UTF8String (SIZE (1 .. 256)) OPTIONAL,       --- полное
наименование
 contact [1]      UTF8String (SIZE (1 .. 256)) OPTIONAL,       --- ФИО
отправителя/контактное лицо
 phone-fax [2]    UTF8String (SIZE (1 .. 128)) OPTIONAL,       ---
контактные телефоны, факс
 index-post [3]   PrintableString (SIZE (1 .. 10)) OPTIONAL,   --- почтовый
индекс
 address [4]          RequestedAddress OPTIONAL                --- контактный
адрес
}

ReportedPostMailingAbonent ::= SEQUENCE {
 personality-info   ReportedPostMailingAbonentInfo,
 phone-fax [0]     UTF8String (SIZE (1 .. 128)) OPTIONAL,      ---
контактные телефоны, факс
 contract [1]      UTF8String (SIZE (1 .. 64)) OPTIONAL,       --- номер договора
 contact-address [2] ReportedAddress OPTIONAL                  --- адрес
}

--- информация об абоненте отправителе/получателе почтового отправления в
записи блока отчета
ReportedPostMailingAbonentInfo ::= SEQUENCE {
 id     TAGGED.&id ({ReportedAbonentsMailingData}),
 data  TAGGED.&Data({ReportedAbonentsMailingData}{@id})
```

```
}

ReportedAbonentsMailingData TAGGED ::= {
 reportedMailingPerson |                                    --- информация об
абоненте - физическом лице
 reportedMailingOrganization                                 ---
информация об абоненте - юридическом лице
}

reportedMailingPerson TAGGED ::= {
 OID { sorm-reported-post-abonent-person-mailing }
 DATA ReportedAbonentMailingPerson
}

reportedMailingOrganization TAGGED ::= {
 OID { sorm-reported-post-abonent-organization-mailing }
 DATA ReportedAbonentMailingOrganization
}

ReportedAbonentMailingPerson ::= SEQUENCE {
 name-info [0]      PersonNameInfo OPTIONAL,                  --- ФИО
 birth-date [1]     GeneralizedTime OPTIONAL,                 --- дата
рождения
 passport-info [2]  PassportInfoReport OPTIONAL               --- паспортные
данные
}

ReportedAbonentMailingOrganization ::= SEQUENCE {
 full-name [0]      UTF8String (SIZE (1 .. 256)) OPTIONAL,    --- полное
наименование
 contact [1]        UTF8String (SIZE (1 .. 256)) OPTIONAL,    --- ФИО
отправителя/контактное лицо
 index-post [3]     PrintableString (SIZE (1 .. 10)) OPTIONAL --- почтовый
индекс
}
END
```

PostIdentifiers.asn

```
PostIdentifiers DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS PostOffice;
```

```
IMPORTS TAGGED,
        sorm-post-office,
        sorm-post-office-identifier
        FROM Classification;



--- идентификаторы отделения почтовой связи
PostOffice ::= SEQUENCE {
 id TAGGED.&id ({PostOfficeVariants}),
 data TAGGED.&Data ({PostOfficeVariants}{@id})
}

--- варианты запрашиваемых идентификаторов
PostOfficeVariants TAGGED ::= {
 postOfficeIdentifier                    --- идентификатор почтового отделения
}

--- поля параметра запроса на физическое лицо
postOfficeIdentifier TAGGED ::= {
 OID { sorm-post-office-identifier }
 DATA PostOfficeIdentifier
}

PostOfficeIdentifier ::= SEQUENCE {
   index-post PrintableString (SIZE (1 .. 10)) OPTIONAL,      --- индекс отделения
почтовой связи
   full-name  UTF8String (SIZE (1 .. 128)) OPTIONAL   --- полное наименование
отделения почтовой связи
}
END
```

PostTypes.asn

```
PostTypes DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS PostType,
        PostInfo;

IMPORTS  DateAndTime
   FROM Sorm

   TAGGED,
```

```
  sorm-post-type,
  sorm-post-type-mail,
  sorm-post-type-package,
  sorm-post-type-money-order,
  sorm-post-type-secogram,
  sorm-post-type-postal-wrapper,
  sorm-post-type-postcard,
  sorm-post-type-ems,
  sorm-post-type-small-packet,
  sorm-post-type-bag-international

  FROM Classification

  PostOffice
  FROM PostIdentifiers

  MoneyType
  FROM Dictionaries
  ;

PostType ::= SEQUENCE {
 id TAGGED.&id ({ReportedPostVariants}),
 data TAGGED.&Data ({ReportedPostVariants}{@id})
}

ReportedPostVariants TAGGED ::= {
      postMail       --- письмо
      | postPackage    --- посылка
      | postMoneyOrder  --- денежный перевод
     | postSecogram    --- секограмма
     | postalWrapper   --- бандероль
      | postCard       --- почтовая карточка
     | postEMS        --- экспресс-отправление
     | postPacket     --- пакет
     | postBagInternational --- мешок, международное
}

MailingType ::= ENUMERATED {
      simple (0),                          -- простое почтовое отправление
      registered (1),                          -- заказное почтовое
отправление
      international (3),                   -- международное почтовое
отправление
```

```
        ordinary (4),                                   -- обыкновенное почтовое
отправление
        with-declared-value (5),                        -- почтовое отправление с
объявленной ценностью
        with-declared-value-and-cod (6)                 -- почтовое отправление
с объявленной ценностью и наложенным платежом
}

--- письмо
postMail TAGGED ::= {
 OID { sorm-post-type-mail }
 DATA PostMailContent
}

--- письмо
PostMailContent ::= SEQUENCE {
   type-mail        MailingType,                    --- тип письма
   pay-on-delivery [2] MoneyType OPTIONAL,          --- наложенный
платеж
   first-class [3]    BOOLEAN OPTIONAL              --- почтовое
отправление первым классом
}

-- посылка
postPackage TAGGED ::= {
 OID { sorm-post-type-package }
 DATA PostPackageContent
}

--- посылка
PostPackageContent ::= SEQUENCE {
   type-package      MailingType,                   --- тип посылки
   pay-on-delivery [2] MoneyType OPTIONAL           --- наложенный платеж
}

--- денежный перевод
postMoneyOrder TAGGED ::= {
 OID { sorm-post-type-money-order }
 DATA PostMoneyOrderContent
}

--- денежный перевод
PostMoneyOrderContent ::= SEQUENCE {
```

```
   amount              MoneyType,                              --- величина почтового
перевода
   inn [1]             NumericString (SIZE ( 10 .. 12 )) OPTIONAL,      --- ИНН
   correspondent-account [2] NumericString (SIZE ( 20 )) OPTIONAL,      --- кор.
счет
   current-account [3]     PrintableString (SIZE ( 20 .. 25 )) OPTIONAL,     --- рас.
счет
   name-bank [4]           UTF8String (SIZE ( 1 .. 256 )) OPTIONAL,          ---
наименование банка
   bic [5]             NumericString (SIZE ( 9 )) OPTIONAL,             --- БИК
   message [6]             UTF8String (SIZE ( 1 .. 70 )) OPTIONAL -- Сообщение
}

--- секограмма
postSecogram TAGGED ::= {
 OID { sorm-post-type-secogram }
 DATA PostSecogramContent
}

--- секограмма
PostSecogramContent ::= MailingType


--- бандероль
postalWrapper TAGGED ::= {
 OID { sorm-post-type-postal-wrapper }
 DATA PostalWrapperContent
}

--- бандероль
PostalWrapperContent ::= SEQUENCE {
   type-wrapper       MailingType,                --- тип бандероли
   pay-on-delivery [2] MoneyType OPTIONAL,              --- наложенный платеж
   first-class [3]    BOOLEAN OPTIONAL             --- почтовое отправление
первым классом
}

--- почтовая карточка
postCard TAGGED ::= {
 OID { sorm-post-type-postcard }
 DATA PostcardContent
}

--- почтовая карточка
```

```
PostcardContent ::= MailingType


--- экспресс-отправление
postEMS TAGGED ::= {
 OID { sorm-post-type-ems }
 DATA PostEMSContent
}

--- экспресс-отправление
PostEMSContent ::= SEQUENCE {
   type-ems          MailingType,                          --- тип экспресс-отправления
   pay-on-delivery [2] MoneyType OPTIONAL                  --- наложенный платеж
}

--- пакет
postPacket TAGGED ::= {
 OID { sorm-post-type-small-packet }
 DATA PostPacketContent
}

--- пакет
PostPacketContent ::= MailingType


--- мешок, международное
postBagInternational TAGGED ::= {
 OID { sorm-post-type-bag-international }
 DATA PostBagContent
}

--- мешок, международное
PostBagContent ::= MailingType

--- характеристики отправления
PostInfo ::= SEQUENCE {
   post-weight [0]      Weight  OPTIONAL,                  --- вес почтового
отправления
   post-cost [1]        MoneyType OPTIONAL,                --- объявленная
стоимость отправления
   post-description [2]   UTF8String (SIZE (1 .. 1024)) OPTIONAL --- завявленное
описание содержимого отправления
}
```

```
-- вес почтового отправления
Weight ::= SEQUENCE {
   weight [0]     INTEGER OPTIONAL,     --- грамм
   weight-text [1] UTF8String (SIZE (1 .. 16)) OPTIONAL   --- текстовое описание
веса
}
END
```

Reports.asn

```
Reports DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS reportMessage,
        Acknowledgement;

IMPORTS TAGGED,
        sorm-message-report
        FROM Classification

        Message, MessageID, DateAndTime
        FROM Sorm

        TaskID
        FROM Tasks

        DictionaryReport
        FROM Dictionaries

        PostingReport
        FROM ReportsConnections

        PresenseReport
        FROM ReportsPresense

        NonFormalizedReport
        FROM ReportsNonFormalized

        AbonentsReport
        FROM ReportsAbonents;

 reportMessage TAGGED ::= {
  OID { sorm-message-report }
  DATA CHOICE {
```

```
   report [0]              Report,                          --- тип сообщения "отчет"
   ack [1]                 Acknowledgement                  --- тип сообщения
"подтверждение"
  }
}

-- Блок данных сообщения типа "отчет"
Report ::= SEQUENCE {
  request-id MessageID,                                    --- идентификатор
запроса, запросивший отчёт
  task-id TaskID,                                          --- идентификатор задачи,
сгенерировавшей данный отчет
  total-blocks-number INTEGER (1 .. 1000000000000),        --- общее
количество блоков в отчете
  block-number INTEGER (1 .. 1000000000000),               ---
порядковый номер текущего блока
  report-block ReportDataBlock                             --- блок данных
отчета
}

-- Подтверждение приёма блока, передаётся с номером соощения
соответствующему номеру сообщения блока отчёта
Acknowledgement ::= SEQUENCE {
  successful BOOLEAN,                                      --- признак
успешного приёма блока
  broken-record INTEGER (10 .. 100000) OPTIONAL,           --- номер
записи в отчете, обработанной на ПУ с ошибкой
  error-description UTF8String (SIZE(1..1024)) OPTIONAL    ---
описание ошибки приёма блока в произвольной форме
}

ReportDataBlock ::= CHOICE {
  dictionary [0]     DictionaryReport,                     --- отчеты задач
пополнения справочников (нормативно-справочная информация)
  abonents [1]       AbonentsReport,                       --- отчеты задач
поисков по принадлежности абонентов
  connections [2]    PostingReport,                        --- отчеты задач поисков
по оказанным услугам почтовой связи
  presense [3]       PresenseReport,                       --- отчеты задач по
запросу наличия в ТС ОРМ информации
  nonFormalized [4]  NonFormalizedReport                   --- отчеты задач по
обработке неформализованных данных
}
END
```

ReportsAbonents.asn

```
ReportsAbonents DEFINITIONS IMPLICIT TAGS ::=
BEGIN
EXPORTS AbonentsReport;
IMPORTS TAGGED,
        sorm-post-abonent-report
      FROM Classification

        PostContractAbonent
        FROM PostAbonents;

AbonentsReport ::= SEQUENCE {
 id     TAGGED.&id ({AbonentsReportVariants}),
 data   TAGGED.&Data({AbonentsReportVariants}{@id})
}

AbonentsReportVariants TAGGED ::= {
 reportAbonent                                    --- информация об абоненте
}

reportAbonent TAGGED ::= {
 OID { sorm-post-abonent-report }
 DATA SEQUENCE OF PostContractAbonent
}
END
```

ReportsConnections.asn

```
ReportsConnections DEFINITIONS IMPLICIT TAGS ::=
BEGIN
EXPORTS PostingReport,
        PostRecord;
IMPORTS  DateAndTime
   FROM Sorm
        TelcoID
   FROM Dictionaries
         TAGGED,
         sorm-report-post-connection
   FROM Classification
        PostOffice
   FROM PostIdentifiers
        PostType,
```

```
    PostInfo
  FROM PostTypes
    ReportedPostMailingAbonent
  FROM PostAbonents
    ReportedAddress
  FROM Addresses
    ;

PostingReport ::= SEQUENCE OF PostRecord

PostRecord ::= SEQUENCE {
 id TAGGED.&id ({ReportedPostVariants}),
 data TAGGED.&Data ({ReportedPostVariants}{@id})
}

ReportedPostVariants TAGGED ::= { postRecord }

-- Детализированные записи об оказанных услугах почтовой связи
postRecord TAGGED ::= {
 OID { sorm-report-post-connection }
 DATA PostRecordContent
}

PostRecordContent ::= SEQUENCE {
  telco-id          TelcoID,          --- идентификатор оператора связи или
филиала
  post-id           UTF8String (SIZE (1 .. 32)), --- идентификатор почтового
отправления
  post-registration    DateAndTime,       --- дата регистрации почтового
отправления
  sender-region-id     UTF8String (1..32),  --- идентификатор региона
отправителя
  receiver-region-id   UTF8String (1..32),  --- идентификатор региона
получателя
  operation-type ENUMERATED {            --- атрибут операции по обработке
отправления
  international-import (0),          --- импорт в пункт международного обмена
  customs-delivery (1),              --- передано таможне
  international-office-receive (2),     --- прибыло в место международного обмена
  international-office-send (3),        --- покинуло место международного обмена
  post-office-receive (4),           --- прибыло в сортировочный центр
  post-office-send (5),              --- покинуло сортировочный центр
  post-office-processing (6)           --- обработано на сортировочном центре
  },
```

```
  operation-description  UTF8String (SIZE (1 .. 256)), --- описание операции

  sender-info          ReportedPostMailingAbonent,        --- информация об
отправителе
  reciever-info        ReportedPostMailingAbonent,        --- информация об
адресате почтового отправления

  receiver-post-office  PostOffice,          --- информация об узле назначения
почтовой связи (получателя)


  post-office-type ENUMERATED {              --- тип операции на узле почтовой
связи
      receive (0),         --- приема почтового отправления
      process (1),         --- обработки почтового отправления
      delivery (2),        --- выдачи почтового отправления
      delivery-failed (3)   --- неудачная попытка вручения
  },
  post-office          PostOffice,         --- информация об узле почтовой связи

  delivery-address [0]  ReportedAddress,           --- адрес доставки отправления
  time-of-send [1]      DateAndTime OPTIONAL,      --- дата и время
отправления
  time-of-receive [2]   DateAndTime OPTIONAL,      --- дата и время получения
  post-type [3]         PostType OPTIONAL,         --- информация о почтовом
отправлении
  post-info [4]         PostInfo OPTIONAL,         --- характеристики отправления
  total-parts [5]       INTEGER (1..1000) OPTIONAL,  --- общее количество
составных частей отправления
  part-number [6]       UTF8String (1..128) OPTIONAL,--- идентификатор
составной части отправления
  third-part-payment [7] ReportedPostMailingAbonent OPTIONAL ---
информация об оплачивающей третьей стороне
}
END
```

ReportsNonFormalized.asn

```
ReportsNonFormalized DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS NonFormalizedReport;

IMPORTS
```

```
        EntityId,
        NonFormalizedEntityAttributeData
        FROM TasksNonFormalized

        StandardInterval
        FROM ReportsPresense;

--- отчет задачи по обработке неформализованных данных
NonFormalizedReport ::= CHOICE {
 nonformalized-report [0]   NonFormalizedRecords,                    --- отчет по
задаче обработки неформализованных данных
 nonformalized-presense [1] NonFormalizedPresenseInfo                ---
отчет по наличию неформализованных данных заданного вида
}

NonFormalizedRecords ::= SEQUENCE OF NonFormalizedRecord

NonFormalizedPresenseInfo ::= SEQUENCE OF StandardInterval                --
- диапазоны времени за которые есть неформализованные данные

--- запись неформализованных данных
NonFormalizedRecord ::= SEQUENCE OF NonFormalizedEntityAttributeData

END
```

ReportsPresense.asn

```
ReportsPresense DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS PresenseReport,
        StandardInterval;

IMPORTS FindRange,
        DateAndTime
    FROM Sorm

        TelcoID
    FROM Dictionaries

        TAGGED,
        sorm-report-presense-abonents,
        sorm-report-presense-connections,
        sorm-report-presense-dictionaries
```

```
   FROM Classification;

---отчет по запросу наличия информации
PresenseReport ::= SEQUENCE {
 id TAGGED.&id ({ReportedPresensesVariants}),
 data TAGGED.&Data ({ReportedPresensesVariants}{@id})
}

ReportedPresensesVariants TAGGED ::= {
   abonentsPresence
 | connectionsPresence
 | dictionariesPresence
}

--- отчет по наличию информации по абонентам и их идентификаторам.
--- Для каждого стандарта может быть указано более одного, либо ни
--- одного интервала (фактические периоды наличия информации);
abonentsPresence TAGGED ::= {
  OID { sorm-report-presense-abonents }
  DATA SEQUENCE OF StandardInterval
}

--- отчет по наличию информации по оказанным услугам почтовой связи
--- может быть указано более одного, либо ни одного, интервала (фактические
периоды наличия информации);
connectionsPresence TAGGED ::= {
  OID { sorm-report-presense-connections }
  DATA SEQUENCE OF ConnectionsPresenseRecord
}

ConnectionsPresenseRecord ::= SEQUENCE OF StandardInterval

--- отчет о наличии информации справочников в ТС ОРМ
--- если какой-либо из справочников не публикуется ТС ОРМ, запись о нем
отсутствует
dictionariesPresence TAGGED ::= {
  OID { sorm-report-presense-dictionaries }
  DATA SEQUENCE OF DictionaryInfo
}

--- запись отчета о наличии информации справочников
DictionaryInfo ::= SEQUENCE {
 telco-id      TelcoID,                          --- идентификатор оператора связи
или филиала
```

```
 dict        ObjectDescriptor,                    --- тип справочника, по
которому есть информация
                                                  --- (идентификатор запроса
справочника) (Requested...)
 count       INTEGER (1 .. 4294967295),           --- количество записей в
справочнике
 change-dates  FindRange                          --- минимальное и
максимальное дата/время изменения записей в справочнике
}


-- ObjectDescriptor принимает значение одно из:
-- sorm-report-dictionary-telcos
-- sorm-report-dictionary-money-types
-- sorm-report-dictionary-post-offices
-- sorm-report-dictionary-doc-types

--- интервал времени, на котором имеются данные по абонентам, соединениям
StandardInterval ::= SEQUENCE {
 telco-id  TelcoID,                               --- идентификатор оператора
связи или филиала
 range    FindRange                               --- интервал времени, на
который имеются данные в ИС
}
END
```

RequestedAbonents.asn

```
RequestedAbonents DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS RequestedAbonent;

IMPORTS TAGGED,
        sorm-request-abonent,
        sorm-requested-abonent-organization,
        sorm-requested-abonent-person
        FROM Classification

        RequestedAddress
        FROM Addresses

        PersonNameInfo,
      PassportInfoReport
```

```
        FROM PostAbonents
        ;

RequestedAbonent ::= SEQUENCE {
  id TAGGED.&id ({RequestedAbonentVariants}),
  data TAGGED.&Data ({RequestedAbonentVariants}{@id})
}

--- варианты запрашиваемых идентификаторов
RequestedAbonentVariants TAGGED ::= {
  requestedPostContractData |
  abonentContractPerson |
  requestedAbonentContractOrganization
}

requestedPostContractData TAGGED ::= {
  OID { sorm-request-contract }
  DATA RequestedPostContract
}

RequestedPostContract ::= CHOICE {
  contract [0]     UTF8String (SIZE (1 .. 64)),          --- номер договора
  bank [1]         UTF8String (SIZE(1 .. 256)),          --- банк абонента
(используемый при расчетах с оператором связи)
  bank-account [2]  UTF8String (SIZE(1 .. 512)),          --- счет абонента в
банке (используемый при расчетах с оператором связи)
  corr-account [3]  UTF8String (SIZE(1 .. 512)),          --- корреспондентский
счет
  kpp [4]          UTF8String (SIZE(9 .. 64)),           --- КПП
  bik [5]          UTF8String (SIZE(9 .. 64)),           --- БИК

  contacts [6]      RequestedContactInfo,                --- контактное лицо
  addresses [7]     RequestedPostContractAddress         --- адрес по договору
}

RequestedPostContractAddress ::= CHOICE {
  address [0]   RequestedAddress,                    --- адрес, указанный в
контракте
  index-post [1] PrintableString (SIZE (1 .. 10))      --- почтовый индекс адреса
}

--- контактная информация по договору
RequestedContactInfo ::= CHOICE {
  contact [0]       PersonNameInfo,      --- ФИО отправителя/контактное лицо
```

```
 phone [1]      UTF8String (SIZE (1 .. 32)),   --- контактный телефон
 email [2]       UTF8String (SIZE (1 .. 128))   --- e-мейл адрес
}

--- Пользователь услуг почтовой связи - физическое лицо
abonentContractPerson TAGGED ::= {
 OID { sorm-requested-abonent-person }
 DATA RequestedAbonentContractPerson
}

RequestedAbonentContractPerson ::= CHOICE {
 name-info   [0]  PersonNameInfo,                          --- ФИО
 birth-date  [1]  GeneralizedTime ,                        --- дата рождения
 passport-info [2]  PassportInfoReport,                --- паспортные данные
 phone-fax   [3]  UTF8String (SIZE (1 .. 128))             --- контактные
телефоны, факс
}

requestedAbonentContractOrganization TAGGED ::= {
 OID { sorm-requested-abonent-organization }
 DATA RequestedAbonentContractOrganization
}

RequestedAbonentContractOrganization ::= CHOICE {
 full-name [0]    UTF8String (SIZE (1 .. 128)), --- полное наименование
 inn [1]         NumericString (SIZE ( 10 .. 12 )),      --- ИНН
 ogrn [2]        UTF8String (SIZE(1 .. 32)),       --- ОГРН
 okpo [3]         UTF8String (SIZE(1 .. 32)),     --- ОКПО
 index-post [4]   PrintableString (SIZE (1 .. 10))  --- почтовый индекс
}
END
```

RequestedConnections.asn

```
RequestedConnections DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS RequestedConnection;

IMPORTS TAGGED,
        sorm-request-post-connection
        FROM Classification

DateAndTime
```

```
  FROM Sorm
PostOffice
  FROM PostIdentifiers
      PostType,
    PostInfo
  FROM PostTypes
      RequestedPostMailingAbonent
  FROM PostAbonents
      RequestedAddress
  FROM Addresses
      ;

RequestedConnection ::= SEQUENCE {
 id TAGGED.&id ({RequestedConnectionVariants}),
 data TAGGED.&Data ({RequestedConnectionVariants}{@id})
}

--- варианты запрашиваемых параметров связей
RequestedConnectionVariants TAGGED ::= {
  requestedPostConnection      --- параметры записи о почтовом отправлении
}

requestedPostConnection TAGGED ::= {
 OID { sorm-request-post-connection }
 DATA CHOICE {
      post-id [0]          UTF8String (SIZE (1 .. 32)),  --- идентификатор почтового
отправления
      operation-type [1] ENUMERATED {              --- атрибут операции по
обработке отправления
            international-import (0),          --- импорт в пункт международного
обмена
            customs-delivery (1),              --- передано таможне
            international-office-receive (2),      --- прибыло в место
международного обмена
            international-office-send (3),        --- покинуло место международного
обмена
            post-office-receive (4),           --- прибыло в сортировочный центр
            post-office-send (5)               --- покинуло сортировочный центр
      },

   operation-description  UTF8String (SIZE (1 .. 256)), --- описание операции

   sender-info [3]        RequestedPostMailingAbonent,       --- информация об
отправителе
```

```
  receiver-info [4]      RequestedPostMailingAbonent,      --- информация об
адресате почтового отправления

  receiver-post-office [5] PostOffice,        --- информация об узле почтовой связи

  post-office-type [6] ENUMERATED {           --- тип операции на узле
почтовой связи
      receive (0),        --- приема почтового отправления
      process (1),        --- обработки почтового отправления
      delivery (2),       --- выдачи почтового отправления
      delivery-failed (3)  --- неудачная попытка вручения
  },
  post-office [7]           PostOffice,       --- информация об узле почтовой связи

  delivery-address [8]  RequestedAddress,    --- адрес доставки отправления
  post-type [9]       PostType,          --- информация о почтовом отправлении
  post-info [10]      PostInfo           --- характеристики отправления
 }
}
END
```

Sessions.asn

```
Sessions DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS sessionMessage;

IMPORTS TAGGED
      ,sorm-message-session
      FROM Classification;

sessionMessage TAGGED ::= {
 OID { sorm-message-session }
 DATA CHOICE {
  connect [0]           ConnectRequest,                  --- запрос на открытие
сессии
  connect-response [1]     ConnectResponse,              --- ответ на запрос
открытия сессии

  adjustment [2]        AdjustmentRequest,               --- согласование
поддерживаемых типов со стороны ПУ
  adjustment-response [3]  AdjustmentResponse,           --- ответ на запрос
согласования данных
```

```
  disconnect [4]          DisconnectRequest,                --- запрос на закрытие
сессии
  disconnect-response [5]   DisconnectResponse                --- ответ на запрос
закрытия сессии
 }
}

--- запрос создания сессии
ConnectRequest ::= SEQUENCE {
  session-timeout          INTEGER (60 .. 2592000),       --- максимальное время
неактивности
  max-data-length          INTEGER (10 .. 100000),        --- максимальная длина
блока отчета (в строках)
  data-packet-window-size    INTEGER (4 .. 256),             --- окно канала
передачи данных
                                                          --- максимальное число
блоков данных, которое может быть
                                                          --- отправлено без
подтверждения приема
  data-load-timeout        INTEGER (1 .. 60),              --- таймаут начала
передачи блоков отчетов
  request-response-timeout   INTEGER (1 .. 60),              --- таймаут ответа
на запрос
  data-packet-response-timeout INTEGER (1 .. 60)             --- таймаут
подтверждения приема блока данных отчета
}

-- ответ на запрос создания сессии
ConnectResponse ::= SEQUENCE {
  confirmed-data-packet-window-size INTEGER (4 .. 256),          ---
подтвержденное окно передачи данных
                                                          --- то окно, которое может
обеспечить ТС ОРМ
                                                          --- должно быть меньше или
равно окну, переданному в ConnectRequest
  confirmed-session-timeout          INTEGER (60 .. 2592000),  ---
подтвержденное максимальное время неактивности
                                                          --- должно быть больше или
равно значению, переданному в ConnectRequest
  confirmed-data-load-timeout        INTEGER (1 .. 60),        ---
подтвержденный таймаут начала передачи блоков отчетов
                                                          --- должен быть больше или
равен значению, переданному в ConnectRequest
```

```
   confirmed-request-response-timeout    INTEGER (1 .. 60),            ---
подтвержденный таймаут ответа на запрос
                                                  --- должен быть больше или
равен значению, переданному в ConnectRequest
   supports SEQUENCE OF ObjectDescriptor                  --- весь список
поддерживаемых ТС ОРМ типов запросов, типов отчётов
}

--- согласование поддерживаемых типов со стороны ПУ
AdjustmentRequest ::= SEQUENCE {
 supports SEQUENCE OF ObjectDescriptor                          --- список
поддерживаемых ПУ типов запросов, типов отчётов
                                                  --- данный список должен
быть меньшим либо равным списку в сообщении ConnectRequest
}

-- ответ на согласование списка поддерживаемых типов
AdjustmentResponse ::= NULL                          --- ответ на запрос
согласования данных

--- запрос завершения сессии
DisconnectRequest ::= NULL

--- ответ на запрос завершения сессии
DisconnectResponse ::= NULL

END
```

Sorm.asn

```
Sorm DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS DateAndTime,
        FindRange,
        MessageID,
        Message;

IMPORTS TAGGED FROM Classification
        sessionMessage FROM Sessions
        trapMessage FROM Traps
        taskMessage FROM Tasks
        reportMessage FROM Reports
        managementMessage FROM Management
```

```
        unformattedMessage FROM Unformatted;

Version ::= PrintableString
vers Version ::= "2.0.0.0"                                  --- текущая версия протокола

-- Оболочка сообщения СОРМ --
Message ::= SEQUENCE {
   version      Version DEFAULT vers,                          --- версия протокола
   message-id    MessageID,                                  --- номер запроса
   message-time  DateAndTime,                                   --- время и дата запроса
   operator-name PrintableString (SIZE (6 .. 128)) OPTIONAL,  --- наименование
оператора связи

   id           TAGGED.&id ({SormPDUs}),                      --- идентификтор блока
данных
   data         TAGGED.&Data({SormPDUs}{@id})                   --- данные блока
данных
}

-- Блок данных сообщения
SormPDUs TAGGED ::= {
    sessionMessage                                          --- сообщения организации
сессии
    trapMessage                                     --- сообщения сигналов
    taskMessage                                     --- сообщения работы с задачами
    reportMessage                                       --- сообщения работы с
отчётами
    managementMessage                                   --- сообщения канала
передачи мониторинга (КПМ)
    unformattedMessage                                  --- сообщения канала
передачи неформатированных данных (КПНФ)
}

--- Общие данные

-- Номер сообщения
MessageID ::= INTEGER (0 .. 4294967295)

-- Дата и время
DateAndTime ::= UTCTime

-- Диапазон поиска
FindRange ::= SEQUENCE {
```

```
   begin-find [0] DateAndTime OPTIONAL,                --- время и дата начала
поиска информации
   end-find [1]   DateAndTime OPTIONAL                 --- время и дата
окончания поиска информации
}
END
```

Tasks.asn

```
Tasks DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS taskMessage,
        TaskID,
       ObjectUNI,
       LogicalOperation,
        CreateTaskResponse;

IMPORTS TAGGED,
        sorm-message-task
    FROM Classification
        FindRange,
        MessageID
    FROM Sorm
        TelcoList,
        DictionaryTask
    FROM Dictionaries
        AbonentsTask
    FROM TasksAbonents
        ConnectionsTask
    FROM TasksConnections
        UNIControlTaskRequest,
        UNIControlTaskResponse
    FROM TasksUNI
        PresenseTask
    FROM TasksPresense
        NonFormalizedTaskRequest,
        NonFormalizedTaskResponse
    FROM
        TasksNonFormalized;

taskMessage TAGGED ::= {
  OID {sorm-message-task}
  DATA CHOICE {
```

```
   data-ready-request [0]          DataReadyRequest,                    --- запрос
готовности данных
   data-ready-response [1]         DataReadyResponse,                   --- ответ на
запрос готовности данных
   data-load-request [2]           DataLoadRequest,                     --- запрос
загрузки данных
   data-load-response [3]          DataLoadResponse,                    --- ответ на
запрос загрузки данных
   data-drop-request [4]           DataDropRequest,                     --- запрос
удаления данных
   data-drop-response [5]          DataDropResponse,                    --- ответ на
запрос удаления данных
   data-interrupt-request [6]      DataInterruptRequest,                --- запрос
прерывания загрузки данных
   data-interrupt-response [7]     DataInterruptResponse,               --- ответ на
запрос прерывания загрузки данных
   create-task-request [8]         CreateTaskRequest,                --- запрос на
создание задачи по обработке информации
   create-task-response [9]        CreateTaskResponse,                  --- ответ на
запрос создания задачи
   uni-task-request [10]           UNIControlTaskRequest,               --- запрос на
постановку/снятие объекта наблюдения на контроль
   uni-task-response [11]          UNIControlTaskResponse,              --- ответ на
запрос постановки/снятия объекта наблюдения с контроля
   non-formalized-task-request [12]  NonFormalizedTaskRequest,          --
- запрос на создание задачи по обработке неформализованных данных
   non-formalized-task-response [13] NonFormalizedTaskResponse          --
- ответ на запрос создания задачи по обработке неформализованных данных
  }
}

--- в этом запросе не параметров
DataReadyRequest ::= NULL

--- запрос загрузки данных конкретной задачи
DataLoadRequest ::= TaskID

--- запрос удаления данных конкретной задачи
DataDropRequest ::= TaskID

--- запрос прерывания загрузки данных
DataInterruptRequest ::=  TaskID

--- запрос на создание задачи поиска
```

```
CreateTaskRequest ::= SEQUENCE {
  telcos [0]       TelcoList OPTIONAL,                    --- список операторов
связи
  range [1]        FindRange OPTIONAL,                    --- временной
диапазон поиска
  report-limit [2]  INTEGER (1 .. 10000000) OPTIONAL,     --- ограничение на
максимальное количество возвращаемых записей
  task [3] CHOICE {
      dictionary [0]          DictionaryTask,             --- задачи
пополнения справочников (нормативно-справочная информация)
      abonents [1]            AbonentsTask,               --- задачи поисков
по принадлежности абонентов
      connections [2]         ConnectionsTask,            --- задачи поисков
по соединениям абонентов
      presense [3]        PresenseTask                    --- задачи предоставления
сведений о наличии данных
  }
}

-- последовательность записей о готовности данных задач
DataReadyResponse ::=  SEQUENCE OF DataReadyTaskRecord

DataReadyTaskRecord ::=   SEQUENCE {
  task-id TaskID,                                         --- идентификатор
задачи
  result  TaskResult                                      --- результат
выполнения задачи
}

TaskResult ::= SEQUENCE {
  result ENUMERATED {
      data-not-ready (0),                                 --- данные не готовы,
задача еще выполняется
      data-ready (1),                                     --- данные есть, задача
выполнена
      data-not-found (2),                                 --- данных нет, задача
выполнена
      error (3)                                       --- в процессе выполнения
задачи произошла ошибка
  },
  report-records-number [0] INTEGER (0 .. 999999999999) OPTIONAL,    --- для
выполненной задачи - количество
                                                          --- записей в отчете
```

```
  report-limit-exeeded  [1] BOOLEAN OPTIONAL,                    ---
количество записей превысило лимит,заданный при создании задачи
  error-description    [2] UTF8String (SIZE (1 .. 256)) OPTIONAL    --- краткое
описание произошедшей ошибки,
                                                                  --- если обнаружена
}

DataLoadResponse ::= SEQUENCE {
  task-id          TaskID,                                        --- идентификатор
задачи, сгенерировавшей данный отчет
  data-exists              BOOLEAN,                               --- признак
существования результатов исполнения задачи
                                                                  --- (есть данные или нет)
  data-blocks-number  INTEGER (0 .. 999999999999) OPTIONAL,      ---
количество блоков в отчете
  error-description      UTF8String (SIZE (1 .. 256)) OPTIONAL   ---
краткое описание ошибки, если обнаружена
}

DataDropResponse ::= SEQUENCE {
  task-id          TaskID,                                        --- идентификатор
задачи, данные которой будут удалены
  successful               BOOLEAN,                               --- признак
успешного выполнения запроса
  error-description      UTF8String (SIZE (1 .. 256)) OPTIONAL    ---
краткое описание ошибки, если обнаружена
}

DataInterruptResponse ::= SEQUENCE {
  request-id         MessageID,                                   --- идентификатор
прерванного запроса загрузки данных
  successful         BOOLEAN,                                     --- признак успешного
выполнения запроса
  data-blocks-available INTEGER (0 .. 999999999999) OPTIONAL,    ---
количество оставшихся непереданными блоков
  error-description    UTF8String (SIZE (1 .. 256)) OPTIONAL     --- краткое
описание ошибки, если обнаружена
}

CreateTaskResponse ::= SEQUENCE {
  task-id          TaskID OPTIONAL,                             --- идентификатор
задачи
  successful       BOOLEAN,                                     --- признак успешного
выполнения запроса
```

```
   error-description UTF8String (SIZE (1 .. 256)) OPTIONAL          --- краткое
описание ошибки, если обнаружена
}

-- идентификатор задачи
TaskID ::= INTEGER (0 .. 4294967295)

-- идентификатор объекта наблюдения
ObjectUNI ::= INTEGER (0 .. 4294967295)

LogicalOperation ::= ENUMERATED {
 operation-open-bracket (0), -- открывающая скобка - "("
 operation-close-bracket (1), -- закрывающая скобка - ")"
 operation-or (2),        -- логическое "или"
 operation-and (3),           -- логическое "и"
 operation-not (4)             -- логическое "не"
}
END
```

TasksAbonents.asn

```
TasksAbonents DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS AbonentsTask;

IMPORTS
   LogicalOperation FROM Tasks
   RequestedAbonent FROM RequestedAbonents;

AbonentsTask ::= RequestedIdentifiers          --- задача на поиск информации о
принадлежности абонентов оператора почтовой связи

RequestedIdentifiers ::= SEQUENCE OF RequestedIdentifierParameters     --
последовательность из идентификаторов и логических операций

RequestedIdentifierParameters ::= CHOICE {
 separator [0] LogicalOperation,                --- логическая операция или
скобка
 find-mask [1] RequestedAbonent                 --- параметр - идентификатор
}
END
```

TasksConnections.asn

```
TasksConnections DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS ConnectionsTask,
      RequestedConnectionIdentifiers;

IMPORTS
   LogicalOperation FROM Tasks
   RequestedConnection FROM RequestedConnections
   ;

ConnectionsTask ::= RequestedConnectionIdentifiers --- задача на поиск
телефонных соединений между абонентами

RequestedConnectionIdentifiers ::= SEQUENCE OF
RequestedConnectionParameter

RequestedConnectionParameter ::= CHOICE {
 separator [0] LogicalOperation,                    --- логический оператор
связки
 find-mask [1] RequestedConnection                  --- параметр
запроса
}
END
```

TasksNonFormalized.asn

```
TasksNonFormalized DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS
      EntityId,
      NonFormalizedTaskRequest,
      NonFormalizedTaskResponse,
      NonFormalizedEntityAttributeData;

IMPORTS TelcoList
      FROM Dictionaries
      FindRange,
      MessageID,
      DateAndTime
      FROM Sorm
      LogicalOperation,
```

```
        CreateTaskResponse
        FROM Tasks
    ;

-- TaskID,
      -- LogicalOperation;

NonFormalizedTaskRequest ::= CHOICE {
   get-entities [0]     GetEntities,                          --- тип сообщения "запрос
получения списка типов сущностей"
   get-attributes [1]    GetEntityAtttibutes,                  --- тип сообщения
"запрос получения списка атрибутов сущности"
   validate-task [2]     ValidateNonFormalizedTask,            --- тип сообщения
"задача поиска неформализованных данных"
   validate-presense [3] NonFormalizedPresenseTask          --- тип сообщения
"задача предоставления сведений о наличии неформализованных данных"
}

NonFormalizedTaskResponse ::= CHOICE {
   entities         [0] GetEntitiesResponse,                 --- ответ на запрос получения
списка типов сущностей
   entity-attributes [1] GetEntityAtttibutesResponse,        --- ответ на запрос
получения списка атрибутов сущности
   validate-task    [2] ValidateNonFormalizedTaskResponse,   --- ответ на запрос
задачи поиска неформализованных данных
   validate-presense [3] NonFormalizedPresenseTaskResponse   --- ответ на запрос
задач предоставления сведений о наличии неформализованных данных
}

--- тип сообщения "запрос получения списка типов сущностей"
GetEntities ::= NULL

--- тип сообщения "запрос получения списка атрибутов сущности"
GetEntityAtttibutes ::= EntityId

--- тип сообщения "задача поиск неформализованных данных"
ValidateNonFormalizedTask ::= SEQUENCE {
 entity-id     EntityId,                                    --- сущность для поиска по
неформализованным данным
 parameters    NonFormalizedParameters,                     --- критерии поиска по
неформализованным данным
 range         FindRange OPTIONAL,                          --- временной диапазон
поиска
```

```
 report-limit  INTEGER (1 .. 10000000)  OPTIONAL          --- ограничение на
максимальное количество возвращаемых записей
}

--- тип сообщения "задача предоставления сведений о наличии
неформализованных данных"
NonFormalizedPresenseTask ::= EntityId

NonFormalizedParameters ::= SEQUENCE OF NonFormalizedParameter

NonFormalizedParameter ::= CHOICE {
 separator [0] LogicalOperation,                    --- логическая операция
 find-mask [1] NonFormalizedEntityCondition         --- условие
}

NonFormalizedEntityCondition ::= SEQUENCE {
 attribute      NonFormalizedEntityAttribute,       --- атрибут сущности
 operation      MathOperation,                      --- операция
 attribute-value NonFormalizedEntityAttributeData   --- значение атрибута
}

NonFormalizedEntityAttribute ::= SEQUENCE {
 attribute-name UTF8String (SIZE (1 .. 256)),       --- текстовое
наименование атрибута сущности
 attribute-type AttributeType                       --- тип данных атрибута
}

NonFormalizedEntityAttributeData ::= CHOICE {
 datetime [0] DateAndTime,                          --- дата и время с
точностью до секунд
 integer [1]  INTEGER (0 .. 4294967296),            --- целочисленный
 string  [2]  UTF8String ( SIZE (0 .. 1024)),       --- строковый
 boolean [3]  BOOLEAN,                              --- булевый
 float   [4]  REAL,                                 --- с плавающей запятой
 empty   [5]  NULL                                  --- пустое значение (null)
}

--- математические операции сравнения
MathOperation ::= ENUMERATED {
 equal (0),          --- равно
 less (1),           --- меньше
 greater (2),        --- больше
 not-equal (3),      --- не равно
 less-or-equal (4),  --- меньше или равно
```

```
 greater-or-equal (5)  --- больше или равно
}

GetEntitiesResponse ::= SEQUENCE OF NonFormalizedEntity

NonFormalizedEntity ::= SEQUENCE {
 entity-id   EntityId,                                    --- уникальный
идентификатор сущности
 entity-name UTF8String (SIZE (1 .. 256))                 --- текстовое
наименование сущности
}

GetEntityAtttibutesResponse ::=    SEQUENCE {
 entity-id        EntityId,                               --- уникальный
идентификатор сущности
 entity-attributes SEQUENCE OF NonFormalizedEntityAttribute      --- атрибуты
сущности
}

ValidateNonFormalizedTaskResponse ::= CreateTaskResponse
NonFormalizedPresenseTaskResponse ::= CreateTaskResponse
--- типы данных атрибутов
AttributeType ::= ENUMERATED {
 date-time (0),  --- дата и время с точностью до секунд
 integer (1),    --- целочисленный
 string (2),     --- строковый
 boolean (3),    --- булевый
 float (4)       --- с плавающей запятой
}

--- Идентификатор сущности
EntityId ::= INTEGER (0 .. 4294967296)

END
```

TasksPresense.asn

```
TasksPresense DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS PresenseTask;

IMPORTS
      TAGGED,
```

```
        sorm-request-presense
        FROM Classification;

PresenseTask ::= SEQUENCE {
 id TAGGED.&id ({PresenseListVariants}),
 data TAGGED.&Data ({PresenseListVariants}{@id})
}

PresenseListVariants TAGGED ::= { presenseInfo }

presenseInfo TAGGED ::= {
 OID { sorm-request-presense }
 DATA ENUMERATED {
  abonents (1),          --- запрос наличия информации по пользователям
услугами почтовой связи
  connections (2),       --- запрос наличия информации по записям об оказанных
услугам почтовой связи
  dictionaries (3)       --- запрос наличия справочников
 }
}
END
```

TasksUNI.asn

```
TasksUNI DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS
        UNIControlTaskRequest,
        UNIControlTaskResponse;

IMPORTS
        TAGGED
        FROM Classification

        RequestedConnection
        FROM RequestedConnections

       LogicalOperation,
        ObjectUNI
        FROM Tasks

        TelcoList
        FROM Dictionaries
```

```
        ;

UNIControlTaskRequest ::= CHOICE {
        create-uni [0] CreateUNIRequest,    --- запрос на создание объекта
наблюдения и постановки его на контроль
        drop-uni [1]  DropUNIRequest        --- запрос на снятие объекта
наблюдения с контроля и удаление объекта наблюдения
}

UNIControlTaskResponse ::= CHOICE {
        create-uni [0] CreateUNIResponse,
        drop-uni [1]  DropUNIResponse
}

CreateUNIRequest ::= SEQUENCE {
   uni-id           ObjectUNI,                  --- идентификатор объекта наблюдения,
переданный ПУ
   uni-criteria     SEQUENCE OF UNIParameter,       --- критерии отбора для
объекта наблюдения
   telcos           TelcoList OPTIONAL              --- список операторов связи
}

UNIParameter ::= CHOICE {
  separator [0] LogicalOperation,                          --- логический оператор
связки
  find-mask [1] UNIRequestedConnectionIdentifiers          --- параметр
запроса
}

UNIRequestedConnectionIdentifiers ::= SEQUENCE {
  uni-criteria      RequestedConnection,                --- критерий в записи об
отравлении для контроля
  mispelling-distance INTEGER (1..10) OPTIONAL          --- количество
ошибок ввода, заполняется для строковых критериев
}

CreateUNIResponse ::= SEQUENCE {
        uni-id          ObjectUNI OPTIONAL,             --- идентификатор
объекта наблюдения (переданный ПУ), по которму подтверждается команда
        uni-successful      BOOLEAN,                    --- признак
успешной постановки объекта наблюдения на контроль
      uni-error-description UTF8String (SIZE (1 .. 256)) OPTIONAL    --- краткое
описание ошибки, если обнаружена
}
```

```
DropUNIRequest ::= ObjectUNI                --- идентификатор объекта
наблюдения для снятия с контроля

DropUNIResponse ::= SEQUENCE {
        uni-dropped         ObjectUNI,
        uni-successful      BOOLEAN,                                --- признак
успешного снятия объекта наблюдения с контроля
      uni-error-description UTF8String (SIZE (1 .. 256)) OPTIONAL    --- краткое
описание ошибки, если обнаружена
}
END
```

Traps.asn

```
Traps DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS trapMessage;

IMPORTS
   TAGGED,
   sorm-message-trap
   FROM Classification
     MessageID
   FROM Sorm
     PostRecord
   FROM ReportsConnections
    ObjectUNI
   FROM Tasks
   ;
trapMessage TAGGED ::= {
   OID { sorm-message-trap }
   DATA CHOICE {
    trap [0]    Trap,                                  --- тип сообщения "сигнал"
    trap-uni [1] TrapUNI,                                  --- тип сообщения
"отобранная информация по объектам наблюдения"
    trap-ack [2] TrapAck                                   --- тип сообщения
"подтверждение сигнала"
   }
}

-- Блок данных сообщения типа "сигнал"
Trap ::= SEQUENCE {
```

```
 trap-type TrapType,                                    -- тип сообщения
 trap-message UTF8String (SIZE (1 .. 256)) OPTIONAL,    -- описание
сообщения
 reference-message MessageID OPTIONAL                   -- номер
сообщение к которому относится данный сигнал
                                                        -- (например номер
сообщения запросившего отчет при прерывании передачи)
}

TrapType ::= ENUMERATED {
 heartbeat (0),                                         -- тестовый пакет
 no-source-data (1),                                    -- нет данных/связи с ИС ОПС в
течение периода времение
 restart-software (2),                                  -- перезапуск ПО
 unauthorized-access (3),                               -- попытка
несанкционированного доступа
 critical-error (4),                                    -- критическая ошибка
ПО, потеря данных, дальнейшая работа невозможна
 major-error (5),                                       -- серьезная ошибка ПО,
потеря данных, но дальнейшая работа возможна
 minor-error (6)                                        -- незначительная ошибка
ПО, данные не потеряны, дальнейшая работа возможна
}
TrapUNI ::= SEQUENCE OF UNIData
UNIData ::= SEQUENCE {
   object-unis  SEQUENCE OF ObjectUNI,      --- идентификаторы объектов
наблюдения, по которым отобраны записи об отправлениях
   post-records PostRecord                  --- отобранные записи о почтовых
отправлениях по объекту наблюдения
}
-- Блок данных сообщения типа "подтверждение сигнала"
-- Номер сообщения TrapAck должен соответствовать номеру сообщения Trap
TrapAck ::= NULL

END
```

Unformatted.asn

```
Unformatted DEFINITIONS IMPLICIT TAGS ::=

BEGIN

EXPORTS unformattedMessage;
```

```
IMPORTS TAGGED,
   sorm-message-unformatted
   FROM Classification
   TelcoList,
   DictionaryReport
   FROM Dictionaries
   Acknowledgement
   FROM Reports
   PostingReport
   FROM ReportsConnections
   DateAndTime,
   MessageID
   FROM Sorm
   AbonentsReport
   FROM ReportsAbonents
;

unformattedMessage TAGGED ::= {
   OID { sorm-message-unformatted }
   DATA CHOICE {
        request [0]   RawRequest,
        response [1]  RawResponse,
        report [2]    RawReport,
        report-ack [3] RawAcknowledgement
   }
}

RawRequest ::= SEQUENCE {
 telcos   TelcoList,    --- список операторов связи
 raw-task  RawRequestTask --- запрос получения неформатированных данных
}

RawRequestTask ::= CHOICE {
 data-start-request [1]     DataStartRequest,      --- запрос на начало передачи
неформатированных данных
 data-stop-request [2]      DataStopRequest        --- запрос на остановку передачи
неформатированных данных
}

DataStartRequest ::=  NULL

DataStopRequest ::= NULL

RawResponse ::= CHOICE {
```

```
 data-start-response [1]      DataStartResponse,     --- ответ на запрос начала
передачи неформатированных данных
 data-stop-response [2]       DataStopResponse       --- ответ на запрос остановки
передачи неформатированных данных
}

DataStartResponse ::= BOOLEAN          --- признак успешности выполнения
команды
DataStopResponse ::= BOOLEAN           --- признак успешности выполнения
команды

RawReport ::= SEQUENCE {
 request-id         MessageID,                        --- идентификатор запроса
 stream-id          UTF8String (SIZE (1 .. 256)),     --- идентификатор потока в
сессии
 total-blocks-number INTEGER (0 .. 999999999999),     --- общее количество
блоков в отчете
 block-number       INTEGER (1 .. 1000000000000), --- порядковый номер
текущего блока
 report-block       RawDataBlock                      --- блок данных отчета
}

RawDataBlock ::= CHOICE {
 abonents [0]     AbonentsReport,    --- записи отчетов об абонентах
 reports [1]      PostingReport,     --- записи отчетов о почтовых отправлениях
 dictionaries [2] DictionaryReport   --- записи отчетов о справочниках
}

RawAcknowledgement ::= Acknowledgement
END.
```

Приложение № 9 к Требованиям к оборудованию и программному обеспечению операторов почтовой связи для обеспечения доступа к базам данных об оказанных услугах почтовой связи и пользователях услугами почтовой связи при проведении оперативно-розыскных мероприятий, утвержденным приказом Минкомсвязи России от 23.07.2015 № 279

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

АПК – аппаратно-программный комплекс.

БД – база данных.

Время выполнения задачи в ТС ОРМ – временной интервал между началом исполнения поисковой задачи в БД ТС ОРМ и временем завершения формирования результата в БД ТС ОРМ.

ИС – информационная система.

Источник информации – аппаратно-программные средства оператора связи, обеспечивающие ввод в ТС ОРМ информации о пользователях услуг почтовой связи и оказанных им услугах почтовой связи.

КТС – комплекс технических средств.

НСД – несанкционированный доступ.

ОПС – оператор почтовой связи.

ПУ – пункт управления.

ПО – программное обеспечение.

СПО – специальное программное обеспечение.

ТС ОРМ – технические средства оперативно-розыскных мероприятий.