

L\_ogbq\_kdb\_ lj\_[h\Zgby  
ij\_^ty\ey\_dubgnhjfZpbhgguf kbkl\_fZf kh^\_j`z  
^Zgguo Z[hg\_glh\ hi\_jZlhjZ k\yab b ij\_^hklZ\  
k\yab ijb \aZbfh^\_ckl\bb k miheghh q\_ggu  
hkms\_kl\eyxsbfbZlb\jzha ukdgnyx\_e\_vghklv

<\_jkb5

KH>?J@:GB?

1	H[sb_ lj_[h\Zgby ij_ ^ty\ey_fu_ d \j_f_gguf oZ [Zau ^Zgguo Z[hg_glh\ hi_jZihjZek\ygabo mkem]Zo k\yab	4
2	H[sb_ nmgdpbhgZevgu_ l_ogbq_kdb_	5
3	Lj_[h\Zgby ij_ ^ty\ey_fu_ d \j_f_gguf oZ bgnhjZpbhgguo kbkl_f b [ZaZf ^Zgguo	11
4	ljbeh`_gbLj_[h\Zgby d khklZ\m gZdZieb bgnhjZpbb h kh_ ^bg_gbyo Z[hg_glh\	14
5	ljbeh`_gbLj_[h\Zgby ij_ ^ty\ey_fu_ d bg \aZbfh^_ckl\by f_`^m IM b BK KHJF	20
6	ljbeh`_gbLj_[h\Zgby ij_ ^ty\ey_fu_ d nmgdpbhgbjh\Zgbx dZgZeZ di^	31
7	ljbeh`_gbLj_[h\Zgby ij_ ^ty\ey_fu_ d nmgdpbhgbjh\Zgbx dZgZeZ di^	35
8	ljbeh`_gbLj_[h\Zgby ij_ ^ty\ey_fu_ d nmgdpbhgbjh\Zgbx dZgZeZ di^	39
9	ljbeh`_gbLj_[h\Zgby ij_ ^ty\ey_fu_ d nmgdpbhgbjh\Zgbx dZgZeZ di^	42
10	ljbeh`_gbLj_[h\Zgby ij_ ^ty\ey_fu_ d nmgdpbhgbjh\Zgbx dZgZeZ di^	47
	ljbeh`_gbASN.1-ki_pbnbdZpby ijhlhdheZ \aZbfh^_ckl\by IM b BK KHJF	52

I?J?Q?GV L?JFBGH< B KHDJ:S?GBC

:ID		:iiZjZlgi h]jZffguc dhfie_dk
; >		;ZaZ ^Zgguo
<j_fy \uiheg_ aZ^Zqb \ BK		<j_f_gghc bgl_j\Ze f_`^m gZqZ ihbkdh\hc aZ^Zqb \ ;> BK KHJF aZ\_jr_gby nhjfbjh\Zgby j_ame KHJF
BK KHJF		BgnhjZpbhgg_u kbkl_fu kh^_j ^Zgguo hi_jZlhjh\ k\yab g_h[o h[_ki_q_gby ijh\_^_gby]Za_ujZl f_jhijbylbc gZ k_lyo we_dljhk\
Bklhqgbd bgnhjZpbb		:iiZjZlgi h]jZffgu_ kj_^kl\Z hi_j k\yab h[_ki_qb\Zxsb_ \\h^ \ B bgnhjZpbb h ihevah\Zl_eyo m ij_^hklZ\e_gguo bf mkem]Zo k\
DLK		Dhfie_dk l_kgba_kj_^kl\
GK>		G_kZgdpbhgbjh\Zgguc ^hklmi
IM		Imgdl mijZ\e_gby
IH		ljh]jZffgh_ h[_ki_q_gb_

1 H[sb\_ lj\_[h\Zgby d b gmgj fuZ pbb kl\_fZf kh^\_j`Zsbf  
^Zgguo Z[hg\_glh\ hi\_jZlhjZ k\yab b ij\_^hekmZg\_e\_g  
k\yab

1.1 BgnhjfZpbhggu\_ kbkl\_fu kh^\_j`Zsb\_ [Zau ^Zgg  
g\_h[oh^bfu\_ ^ey h[\_ki\_q\_gby ijh\\_^\_gby hi\_j  
f\_jhijbylbc gZ k\_lyo we\_dljh\Bkb KHEE\_\_  
ij\_^gZagZq\_gu ^ey gZdhie\_gby ojZg\_gby\_gb]jZ[  
miheghfhq\_gguf ]hkm^Zjkl\\_gguf hj]ZgZf hkr  
khhI\\_lkl\bb k N\_^\_jZevguf aZdhgh-fjha]g hci\_j  
^\_yl\_evghklb hijhZu k dgmx ^\_yl\_evghklv ^  
miheghfhq\_ggu\_ hj]Zgu bgnhjfZpbb h ihevah\Z  
ij\_^hklZ\ey\_bf mkem]Zo k\yab

1.2 GZklhysb\_ lj\_[h\Zgby jZkijhkljZgHyFkjiheZabK\_fu  
b beb ih^e`Zsb\_ mkIZgh\d\_ gZ k\_lyo hi  
hkms\_kl\eyxsbo ^\_yl\_evghklv \ h[eZklb hdZaZg  
k\yab

± mkem]b f\_klghc l\_e\_nhgghc k\yabkemaZ fbkd gk  
l\_e\_nhgghc k\yab k bkihevah\Zgb\_f IZdkh  
dhee\_dlb\gh]h ^hklmiZ

± mkem]b f\_`^m]hjh^ghc b f\_`^mgZjh^ghc l\_e\_nh

± mkem]b l\_e\_nhgghc k\yab \ \u^\_e\_gghc k\_lb k

± mkem]b \gmljbahgh\hc l\_e\_nhgghc k\yab

± mkem]b f\_klghc hcek\ylad k bkihevah\Zgb\_f IZd

± mkem]b f\_klghc l\_e\_nhgghc k\yab k bkihe  
dhee\_dlb\gh]h ^hklmiZ

± mkem]b l\_e\_]jZngghc k\yab

± mkem]b k\yab i\_jkhgZevgh]h jZ^bh\uah\Z

± mkem]b ih^\b`ghc jZ^bhk\yab \ k\_lb k\yab h[s\_

± mkem]b bi`ghc jZ^bhk\yab \ \u^\_e\_gghc k\_lb k\

± mkem]b ih^\b`ghc jZ^bhl\_e\_nhgghc k\yab \  
ihevah\Zgby

± mkem]b ih^\b`ghc kimlgbdh\hc jZ^bhk\yab

± mkem]b k\yab ih ij\_^hklZ\e\_gbx dZgZeh\ k\yab

± mkem]b k\yab \ k\_lb i\_j\_^Zqb ^Zgguj\_ ^Zqb b  
]hehkh\hc bgnhjfZpbb

± mkem]b k\yab ih i\_j\_^Zq\_ ]hehkh\hc bgnhjfZ  
^Zgguo

± l\_e\_fZlbq\_kdb\_ mkem]b k\yab

1.3 Mkeh\by ijbfg\_gby

1.3.1 MkIZgZ\eb\ZhfjZlhjZfb k\yab gZ k\_lyo k\yab  
^he`gu ij\_^klZ\eyl:VDkh]hcki\_qb\Zxsba kiph[hjZ

j\_]bkljZpbb h[jZ[hldb ojZg\_gby b i\_j\_^Zqb  
hj]ZgZf bgnhjZpbb h ihevah\ZI\_eyo mk  
ij\_^hkIZ\ey\_fuo bf mkem]Zo k\yab ihklmiZxs  
g\_kdhevdbo hi\_jZlhjh\ k\yab

1.3.2 BgnhjZpbyih~~h~~evah\ZI\_eyo mkemij\_ ~~h~~k\yab~~h~~e\_gguo  
ih~~h~~evah\ZI\_ex mkem]Zo k\yab kh^\_j`bl ^Zggu\_  
ih~~h~~evah\ZI\_ey ijb ihemq\_gbb mkem] k\yab h \  
qbke\_ ihiuldb kh\_^bg\_gbc dZd khklhy\  
g\_khklhy\rboky k mqZklb\_f ihevah\ZI\_ey  
Z^j\_kh\Zg\_gmo Z lZd`\_ h[ hlijZ\bl\_e\_ b ihemqZ  
l\_dklh\uo ]hekh\uo beb fmevlbf^bcguo  
ihiuldZo hlijZ\db khh[s\_gbc ^hkIZ\e\_gguo be  
kj\_^kl\Zo k\yab ^Zgghc k\_lb

2 H[sb\_ nmgdpbhgZevgu\_ l\_ogbq\_kdb\_ lj\_[h\Zgby

2.1 Ihklh\_gb\_ BK KHJF ^he`gh :h[\_ki\_qb\Zlv

2.1.1 hj]ZgbaZp~~h~~^ey gZdhie\_gby ojZg\_gby ~~g~~\_l fb\_g  
h[jZ[hldb bgnhjZpbb h ihevah\ZI\_eyo m  
ij\_^hkIZ\ey\_fuo bf mkem]Zo k\yab

2.1.2 ijb\_f b i\_j\bqgmh h[jZ[hldm bgnhjZpbb h ihe  
k\yab b ij\_ ~~h~~^ykh\Z~~o~~ bf mkem]Zo k\yab ba  
bklhqgb ~~h~~gnhjZpbb h^gh]h beb g\_kdhevdbo  
k\yab ^ey aZg\_k\_g~~h~~^yJF ;> BK

2.1.3 \hafh`ghkl~~h~~hemq\_gby bgnhjZp~~h~~bgZ^he`ghklb  
Z[hg\_glkdbo ghf\_jh\ b beb dh^h\ b^\_glbnbdZ  
hdhg\_qgu\_ ~~g~~^e\_fk\_lb k\_l\_c k\yab dhlh  
bkihevah\Zgu ^ey mklZgh\e\_gby kh\_^bg\_gby  
Z\lhfZlbq\_kdhf b beb Z\lhfZlbabh\Zgghf j\_`bf

2.1.4 ^h[Z\e\_gb;> BK KHJF bgnhjZpbb h ihevah\ZI\_ey  
k\yab ~~h~~iha^g\_\_ lj\_o ^g\_c i~~h~~kye\_dhagZ~~h~~gZl~~h~~gZaZgb  
m~~h~~ne] k\yab

2.1.5 ^h[Z\e\_gb;> BK KHJF bgnhjZpbb h[ mkem]Zo  
hdZau\Z\_fuo Z[hg\_glZf ihevah\ZI\_ ~~h~~^y^bngk\_g~~h~~  
ljZnbd\_g\_ aZ\bkbfb hl lh]h ijbkmlkl\m\_l beb  
bgnhjZpby h[ Z[hg\_glZo ihevah\ZI\_e~~h~~byabdh  
ij\_^hkIZ\eyxiky

2.1.6 h[gh\e\_gb;> BK KHJF bgnhjZpbb h ihevah\Z  
k\yab ~~h~~hdZaZgg~~h~~umok ~~h~~fnk\Zyca bieZI\_`Zo  
q g\_ j\_`\_ h^gh]hiyj~~h~~gZZ^p\Zlv fbgml ^ey l\_e\_  
kh\_^bg\_g~~h~~clyo ih^\~~h~~b`ghc l\_e\_nhgghc k\y  
q g\_j\_`\_ h^gh]h jZa~~h~~^ykhqZ~~h~~lg\_gbc kh\~~h~~jr\_  
Z[hg\_glZfb ihkj\_n~~h~~kd~~h~~klh\Zgghc l\_e\_nhggh  
k\yab ~~h~~\_lb i\_j\_^Zqb ~~h~~^Z~~h~~gguo \~~h~~b^h\ k\_l\_c k\

- 2.1.7 aZsbIm hl g\_kZgdpbhgbjh\Zggh]h ^hkHmIFZ d  
ihfhsvx ijh]jZffguo b l\_ogbq\_kdbZ KZ\_d^kl  
hj]ZgbahZgguo;f\_j
- 2.1.8 djm]ehkmlhgguc Z\lhfZlbabjh\Zgguc m^Ze\_gguo  
KHJFkh klhjhgu IM miheghfhq\_gguo hj]Zgh\  
2.1.9 ijb\_f blaM jZaebqguo \b^h\ aZijhkh\ h ihevah  
k\yab b hdZaZgguo bf mkem]Zo k\yab
- 2.1.10 i\_j\_^ZgnZ IMa BK HJFbgnhjfZpbb \\_kkllhbb k  
aZijhkZfb
- 2.1.11 j\_]bkljZpbx b ojZg\_gb\_ ihklmiZxsbo aZijhkh\  
miheghfhq\_ggufb ihevah\ZI\_eyfb

2.2 BK KHJF ^he`gZ y\eylvky aZ\\_jr\_gguf fh^m  
ij\_^hklZ\eyxsbf \hafh`ghklv hi\_jZlhjm k\yab \h^  
k\_l\_c b kj\_^klk k\yab gZ[hjhfh mkem] kh]eZkgh  
ij\_^hklZ\e\_gb\_ mkem] k\yab Fh^mevgh\_ il  
h[\_ki\_qb\Zlv \hafh`ghklv jZkrbj\_gby aZf\_gu kh  
ijh]jZffguo kj\_^kl\ BK KHJF ^ey memqr\_gby w  
oZjZdl\_jbklbd BK KHJFm\hebgj\_gby dhebq\_kl\Z  
ij\_^hklZ\ey\_fuo mkem] k\yab dhebq\_kl\Z Z[h  
h[t\_fh\ h[jZ[Zlu\Z\_fhc bgnhjfZpbb jZkrbj\_gby r  
[\_a baf\_g\_gby bgnhjfZpbb ojZgys\_cky \ m`\_ k  
^Zgguo

2.3 BK KHJF ^he`gZ kh^\_fjhZbvh]hg]Z b mijZ\e\_gby  
KHJF kh klhjhgu IM

2.4 BK KHJF ^he`gZ \\_klb`m]glZAgdexqZxsbk\_[y  
q bgnhjfZpbx h k\_kkbyo  
q bgnhjfZpbx h aZijhkZo gZ ihemq\_gb\_ ^Zgguo  
q bgnhjfZpbx h[ hl\\_lZo gZ aZijhku h ihemq\_gbb  
q bgnhjfZpbx h[ hlq\_lZo  
q bgnhjfZpbx h l\_dms\_c dhgnb]mjZpbb DLK  
ijbdeZ^gh]h IH  
q bgnhjfZpbx h[ baf\_g\_gbyo \ dhgnb]mjZpbb D  
ijbdeZ^gh]h IH  
q bgnhjfZpbx h g\_iheZ^dZo \ DLK kbkl\_fgfh b ij  
q bgnhjfZpbx h ^hklmi\_ l\_ogbq\_kdh]h i\_ja]hgZez  
BK KHJF  
q bgnhjfZpbx h GK> b ihiuldZo GK>  
BK KHJF g\_ ^he`gZ nbdkbjh\Zlv \ kbkl\_fgguo \  
bgnhjfZpbb hlghkysboky d b^\_glbnbdZlhjZf  
ihbkdh\uo aZ^Zq IM

- 2.5 Доступ технического персонала оператора связи к ИС СОРМ должен осуществляться через модуль доступа, который обеспечивает запись команд управления ИС СОРМ и построение парольной системы.  
Технический персонал оператора связи должен иметь доступ только к системным журналам ИС СОРМ, системному ПО и КТС, входящим в ИС СОРМ для проведения ремонтных работ.
- 2.6 Проведение организационно-технических мероприятий должно обеспечивать защиту ИС СОРМ от несанкционированного доступа:
- производителя ИС СОРМ;
  - неавторизованных пользователей;
  - технического персонала оператора связи;
  - третьих лиц.
- 2.7 Защита информации должна осуществляться при:
- загрузке информации в ИС СОРМ;
  - получении ИС СОРМ запросов;
  - обработке информации в ИС СОРМ;
  - хранении запросов к ИС СОРМ и результатов их выполнения;
  - передаче на ПУ результатов обработки запросов.
- 2.8 К ИС СОРМ должны иметь доступ:
- программные средства источников информации о пользователях услуг связи и предоставляемых им услугах связи (далее – источник информации);
  - технический персонал оператора связи;
  - ПУ уполномоченных органов.
- 2.9 Источник информации должен иметь возможность осуществлять ввод информации в ИС СОРМ в согласованном с поставщиком ИС СОРМ формате в автоматическом и/или автоматизированном режиме.
- 2.10 Для технического персонала оператора связи ИС СОРМ должна предоставлять следующие функции:
- доступ к журналу ошибочных блоков переданных отчетов и возможность редактирования ошибочных записей соответствующих отчетов;
  - доступ к аппаратным и программным компонентам ИС СОРМ для проведения регламентных и ремонтных работ.
- 2.11 Технический персонал оператора связи, обслуживающий ИС СОРМ, должен иметь доступ только через специальные технические средства защиты.

2.12 ИС СОПМ должна обеспечивать сбор и запись в БД ИС СОПМ информации как о состоявшихся соединениях абонентов, так и о несостоявшихся.

2.13 ИС СОПМ должна обеспечивать сбор и запись в БД ИС СОПМ информации о следующих видах соединений, совершенных абонентами посредством сети передачи данных:

- подключения/отключения абонента к сети передачи данных (AAA);
- HTTP-обращения к информационному ресурсу сети связи (посещение Интернет-страниц сайтов, порталов и т.д.);
- передачи почтового e-mail сообщения (в т.ч. сообщения веб-почты);
- передачи мгновенных электронных сообщений между пользователями (Instant Messaging);
- голосовой связи посредством сети передачи данных;
- передачи файловых данных;
- терминального доступа к оборудованию для удаленного управления;
- прочих данных (отличных от перечисленных в п.1-п.7), принимаемых, получаемых пользователем при помощи закрытых протоколов обмена (прочие UDP-датаграммы, TCP-потoki, другие IP-пакеты).

Информация о соединениях (в соответствии с лицензиями на оказание услуг связи, перечисленных в п. 1.2 настоящих Требований) должна записываться в БД ИС СОПМ в соответствии с Приложение №1 к настоящим Требованиям.

2.14 ИС СОПМ должна обеспечивать сбор и запись в БД ИС СОПМ информации о следующих видах телефонных соединений абонентов:

- входящих/исходящих телефонных соединениях;
- входящих/исходящих коротких текстовых сообщений (SMS, MMS и т.д.);
- служебных запросов (USSD и т.д.);
- соединения, совершенные абонентом в гостевых сетях (роуминг);
- прочие виды соединений.

Для телефонных соединений абонентов (в соответствии с лицензиями на оказание услуг связи, перечисленных в п. 1.2 настоящих Требований) в БД ИС СОПМ должна записываться информация в соответствии с Приложение №1 к настоящим Требованиям. Телефонные номера абонентов РФ должны записываться в БД ИС СОПМ в международном формате. В случае сокрытия реального номера групповым в БД ИС СОПМ в качестве внутреннего номера абонента должен записываться реальный номер в международном



формате.

2.15 ИС СОРМ должна обеспечивать сбор и запись в БД ИС СОРМ информации об изменении местоположения абонентов как при предоставлении услуги связи так и в пассивном режиме (ожидании вызова). ИС СОРМ должна накапливать информацию об изменении местоположения абонентского устройства (в соответствии с Locations.asn Приложения №8) при переключении его обслуживания, выполняемого разными устройствами сети оператора между разными географическими зонами обслуживания, в т.ч. при включении-выключении абонентского устройства.

2.16 ИС СОРМ должна выполнять следующие действия по предварительной обработке вводимых от аппаратно-программных средств данных о совершенных абонентами соединениях в сети передачи данных перед их записью в БД ИС СОРМ, в т.ч.:

- определение абонента-корреспондента при обмене мгновенными электронными сообщениями (идентификатор), абонента-корреспондента при обмене IP/UDP/TCP трафиком (IP-адрес/порт);
- отслеживание всех совершенных абонентом соединений с абонентом-корреспондентом в течение пяти минут;
- формирование одной общей записи о соединении абонента в сети передачи данных с абонентом-корреспондентом с фиксацией идентификаторов абонента сети передачи данных, абонента-корреспондента;
- обработка аналогичным образом всех последующих совершенных соединений абонента в сети передачи данных.

Приведенные действия по предварительной обработке должны выполняться для перечисленных ниже видов соединений абонентов в сети передачи данных с заполнением в соответствии с Приложение №1 настоящих Требований записей о таких соединениях:

- передачи мгновенных электронных сообщений между пользователями (Instant Messaging);
- терминального доступа к оборудованию для удаленного управления;
- прочих данных, принимаемых, получаемых пользователем при помощи закрытых протоколов обмена (прочие IP/TCP-потoki).

Для HTTP-соединений абонентов в сети передачи данных с информационными ресурсами должны фиксироваться сведения в соответствии с Приложение №1 настоящих Требований. В качестве URL HTTP-ресурса должен заполняться адрес страницы, запрошенной пользователем.

- 2.17 К несостоявшимся соединениям относятся:
- следующие виды телефонных соединений абонентов (любые типы соединений – голосовые, SMS и т.д.):
    - недоступность вызываемого абонента;
    - сброс вызова вызываемым абонентом;
    - вызываемый абонент находился в состоянии разговора на момент поступления нового вызова;
    - сбой сети связи при направлении вызова;
  - соединения, совершенных абонентами посредством сети передачи данных, в случаях, когда отвечающий абонент не смог принять входящее TCP/UDP-соединение либо соединение более высокого уровня.
- 2.18 Если в связи со структурой организации сети связи (центр коммутации, централизованная автоматизированная система расчетов) оператор обслуживает одновременно несколько входящих в его структуру территориально-распределенных филиалов по субъектам РФ (отдельных юридических лиц), то ИС СОРМ может накапливать информацию об абонентах, соединениях, платежах для нескольких филиалов оператора связи.
- Если в связи со структурой организации сети связи (центр коммутации, централизованная автоматизированная система расчетов) оператор предоставляет услуги по тарификации, ведению баз данных об абонентах других юридических лиц, то ИС СОРМ может накапливать информацию об абонентах, соединениях, платежах обслуживаемых других юридических лиц.
- В таком случае ИС СОРМ должна обеспечить ПУ возможность получения списка обслуживаемых ИС СОРМ филиалов.
- При обращении ПУ с поисковыми задачами к ИС СОРМ ПУ должен иметь возможность явно задавать перечень филиалов (других юридических лиц), для которых должна выполняться поисковая задача.
- При явном указании списка филиалов (других юридических лиц) оператора в параметрах поисковой задачи, запрос в целом должен выполняться в соответствии с временными характеристиками, приведенными в Разделе 3 «Требования, предъявляемые к временным характеристикам ИС СОРМ и БД» настоящих Требований. Обновление в БД ИС СОРМ информации об абонентах, соединениях, платежах по каждому филиалу (другому юридическому лицу) должно выполняться в соответствии с п. 2.1.6 настоящих Требований.
- 2.19 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 по Приложение №7), при выполнении запросов

об абонентах ИС СОРМ должна осуществлять полнотекстовый поиск по такой неструктурированной информации, используя поисковые критерии задачи.

### **3 Требования, предъявляемые к временным характеристикам ИС СОРМ и БД.**

- 3.1 Время выполнения задачи в ИС СОРМ не превышает следующих значений:
- 1 сек. для данных о принадлежности идентификаторов абонентов сети оператора связи;
  - 1 сек. для данных об использовании карты экспресс оплаты, телефонной карты;
  - 3 сек. для данных об идентификаторах абонентов зарегистрированных на физическое или юридическое лицо;
  - 1 сек. для данных о пополнении баланса личного счета абонента.
- 3.2 Время выполнения задач поиска информации о связях абонентов обслуживаемых сетью ИС СОРМ не превышает значений, приведенных в Таблице 1\*.

При одновременном выполнении ИС СОРМ более 50 одновременных поисковых задач, обеспечение временных характеристик в соответствии с Таблицей 1 не является обязательным.

Требования к временным характеристикам поиска информации в ИС СОРМ разделяются по следующим классам:

- класс 1 (К1) включает критерии:
  - номер телефона абонента сети оператора;
  - идентификатор подвижной станции абонента(IMEI, ESN/MIN);
  - идентификатор подвижного абонента(IMSI);
  - номер телефонной (таксофонной) карты;
- класс 2 (К2) включает критерии:
  - идентификатор абонента сети передачи данных;
  - идентификатор пользователя в сети передачи данных;
  - прикладной идентификатор пользователя при использовании услуг сети передачи данных;
- класс 3 (К3) включает критерии:
  - номер базовой станции;
  - номер пучка;
  - номер коммутатора;
  - идентификатор узла связи;
  - идентификатор устройства сети передачи данных.

Таблица 1

№	Класс параметра запроса	Временной интервал				
		до суток	до 1 месяца	до 6 месяцев	До 1 года	до 3 лет
1	К1	< 3 сек.	< 5 сек.	< 15 сек.	< 25 сек.	< 50 сек.
2	К2	<1 мин	<10 мин	<30 мин	<1 часа	<3 часов
3	К3	< 7 мин				

\* Примечание. Для отсутствующих в таблице идентификаторов время выполнения задачи в ИС СОРМ может быть больше приведенного.

3.3 ИС СОРМ должна поддерживать возможность выполнения комбинированных запросов. Комбинированными запросами являются поисковые критерии согласно п.3.2, объединенные логическими операциями

ИС СОРМ должна 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 ИС СОРМ 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»:  $T(AB) = \max(T(A), T(B)) * C$  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.

#### **4 Требования к составу накапливаемой ИС СОРМ информации о соединениях абонентов**

- 4.1 Идентификатор абонента в сетях персонального радиовызова, фиксированной сети телефонной связи, мобильной сети связи состоит из:
- 4.1.1 ТФОП-идентификатора, включающего:
    - телефонный номер в международном формате;
    - внутренний номер если есть;
  - 4.1.2 GSM-идентификатора, включающего:
    - телефонный номер в международном формате;
    - идентификатор мобильного абонента;
    - идентификатор мобильной станции;
    - идентификатор SIM-карты абонента (ICC);
  - 4.1.3 CDMA-идентификатора, включающего:
    - телефонный номер в международном формате;
    - идентификатор мобильного абонента;
    - идентификатор мобильной станции;
    - идентификатор мобильного абонента (CDMA);
    - идентификатор SIM-карты абонента (ICC);
  - 4.1.4 идентификатор сети передачи данных, включающего:
    - идентификатор пользовательского оборудования;
    - имя пользователя – login;
    - IP адрес;
    - адрес электронной почты;
    - PIN;
    - телефонный номер абонента сети передачи голосовой информации посредством сети передачи данных;
    - пользовательский домен;
  - 4.1.5 идентификатора сети персонального радиовызова;
  - 4.1.6 идентификатора в сети передачи голосовой информации посредством сети передачи данных, включающего:
    - IP-адрес абонента;
    - общедоступное имя инициатора связи;
    - телефонный номер абонента сети передачи голосовой информации посредством сети передачи данных.
- 4.2 ИС СОРМ должна накапливать следующую информацию о соединениях абонентов в сети персонального радиовызова:
- дата и время соединения;
  - тип соединения;
  - идентификаторы абонента;

- объем переданных данных;
- причина завершения соединения;
- идентификатор оператора связи или филиала.

4.3 ИС СОПМ должна накапливать следующую информацию о соединениях абонентов в фиксированной сети телефонной связи:

- дата и время начала соединения;
- время соединения;
- тип соединения;
- тип вызывающего абонента;
- коммутатор, обслуживший соединение;
- тип вызываемого абонента;
- входящий пучок;
- исходящий пучок;
- код пограничного коммутатора;
- причина завершения соединения;
- идентификатор оператора связи или филиала;
- ДВО при соединении;
- номер телефонной карты;
- идентификаторы вызывающего абонента;
- набранный номер вызываемого абонента;
- идентификаторы вызываемого абонента;
- телефонный номер при переадресации.

4.4 ИС СОПМ должна накапливать следующую информацию о телефонных соединениях абонентов в мобильной сети связи:

- дата и время начала соединения;
- время соединения;
- тип соединения;
- ДВО при соединении;
- тип вызывающего абонента;
- коммутатор, обслуживший соединение;
- тип вызываемого абонента;
- входящий пучок;
- исходящий пучок;
- код пограничного коммутатора;
- код роумингового партнера;
- причина завершения соединения;
- идентификатор оператора связи или филиала;
- идентификаторы вызывающего абонента;
- местоположение вызывающего абонента на конец вызова;
- местоположение вызывающего абонента на начало вызова;
- идентификаторы вызываемого абонента;
- местоположение вызываемого абонента на начало вызова;
- местоположение вызываемого абонента на конец вызова;

- телефонный номер при переадресации.

4.5 ИС СОРМ должна накапливать следующую информацию о соединениях абонентов в сети передачи данных:

4.5.1 Информацию о подключении/отключении абонента к сети передачи данных (AAA), включающую:

- дата и время подключения/отключения к сети передачи данных;
- тип соединения (подключение к сети передачи данных, отключение от сети передачи данных);
- идентификатор сессии;
- выделенный динамический IP-адрес;
- имя пользователя (логин);
- пользовательский пароль;
- код протокола в соответствии с RFC1700 либо номер порта для TCP/UDP;
- вызывающий номер;
- вызываемый номер;
- IP-адрес/порт NAS-сервера;
- объем принятых данных, байт;
- объем переданных данных, байт;
- идентификатор оператора связи или филиала;
- идентификатор пользовательского оборудования;
- наименование GPRS/EDGE точки доступа (APN);
- IP-адрес GPRS/EDGE SGSN;
- IP-адрес GPRS/EDGE GGSN;
- код зоны обслуживания (SAC) GPRS/EDGE;
- базовая станция, через которую установлено соединение (передача данных посредством подвижной сети связи);
- базовая станция, через которую завершено соединение (передача данных посредством подвижной сети связи);
- номер телефонной карты;
- IMSI мобильного абонента;
- идентификатор мобильной станции абонента;
- номер модемного пула.

4.5.2 Общую информацию для всех записей об использовании абонентами услуг связи и ресурсов сети передачи данных (за исключением соединений подключения/отключения к сети связи):

- идентификатор оператора связи или филиала;
- дата и время начала соединения;
- дата и время завершения соединения;
- информация о клиенте (IP/порт);
- информация о сервере (IP/порт);



- код протокола в соответствии с RFC1700 либо номер порта для TCP/UDP;
  - идентификатор точки подключения к сети передачи данных, с которой получена запись о соединении.
- 4.5.3 Информацию о HTTP-соединениях с информационными ресурсами сети передачи данных (посещение Интернет-страниц), включающую:
- URL (адрес, наименование информационного ресурса);
  - объем принятых и переданных данных в соединении (включает как соединения управления так и передачи данных), байт;
  - причина завершения соединения.
- 4.5.4 Информацию о соединениях передачи почтовых e-mail сообщений, включающую:
- отправитель почтового сообщения;
  - список получателей почтового сообщения;
  - список получателей копии почтового сообщения;
  - адрес, куда следует отправлять ответ на письмо;
  - тема почтового сообщения;
  - размер почтового сообщения, включая прикрепленные файлы, байт;
  - наличие прикрепленных файлов в письме (да/нет);
  - список текстовых наименований почтовых серверов, через которые отправлено письмо (в т.ч. сервера веб-почты);
  - причина завершения соединения;
  - протокол, при помощи которого отправлено сообщение;
  - тип пользовательской операции.
- 4.5.5 Информацию о соединениях передачи мгновенных электронных сообщений между пользователями (Instant Messaging, в т.ч. SMS, MMS сообщения, отправленные посредством сети передачи данных), включающую:
- наименование учетной записи пользователя при подключении;
  - пользовательский пароль при подключении;
  - общедоступное имя отправителя;
  - пользовательский идентификатор отправителя (в т.ч. для веб-чата);
  - список получателей, включающий для каждого получателя сообщения: общедоступное имя получателя, пользовательский идентификатор получателя (в т.ч. для веб-чата);
  - размер данных сессии, байт;
  - причина завершения соединения;
  - протокол, при помощи которого отправлены сообщения.
- 4.5.6 Информацию о соединениях голосовой связи посредством сети передачи данных, включающую:
- идентификатор сессии/call-id;
  - идентификатор конференции;
  - длительность разговора, сек.;
  - общедоступное имя инициатора связи

- способ подключения;
- номер вызывающего абонента;
- номер вызываемого абонента;
- объем переданных данных (включает как соединения управления так и передачи данных), байт;
- объем принятых данных (включает как соединения управления так и передачи данных), байт;
- наличие переданной в соединении факсовой информации (Т.38);
- причина завершения соединения;
- входящий пучок;
- исходящий пучок;
- идентификаторы медиашлюзов, обслуживавших соединение;
- протокол передачи;
- ДВО при соединении.

4.5.7 Информацию о соединениях передачи файловых данных, включающую:

- название сервера;
- имя пользователя, наименование учетной записи;
- пользовательский пароль;
- режим работы файлового сервера;
- объем переданных данных (включает как соединения управления так и передачи данных), байт;
- объем принятых данных (включает как соединения управления так и передачи данных), байт;
- причина завершения соединения.

4.5.8 Информацию о соединениях терминального доступа к оборудованию для удаленного управления, включающую:

- объем переданных данных (включает как соединения управления так и передачи данных), байт;
- объем принятых данных (включает как соединения управления так и передачи данных), байт;
- протокол удаленного доступа к оборудованию.

4.5.9 Информацию о соединениях передачи прочих данных принимаемых, получаемых абонентом при помощи закрытых протоколов обмена включающую:

- объем переданных данных (включает как соединения управления так и передачи данных), байт;
- объем принятых данных (включает как соединения управления так и передачи данных), байт;
- протокол передачи данных.

4.5.10 Информацию о трансляции сетевых адресов пользователей, накапливаемую отдельными ИС СОРМ по согласованию с ПУ, включающую:

- дату и время трансляции;
- тип записи (начало/окончание трансляции);

- внутренний сетевой адрес и порт;
- внешний сетевой адрес и порт;
- сетевой адрес назначения и порт;
- тип трансляции сетевых адресов.

4.6 ИС СОРМ должна накапливать информацию о пользователях услугами связи в соответствии с Приложением №8 (ReportsAbonents.asn);

4.7 ИС СОРМ должна накапливать информацию о платежах, совершенных абонентами в соответствии с Приложением №8 (ReportsPayments.asn).

## 5 Требования, предъявляемые к интерфейсу взаимодействия между ПУ и ИС СОРМ

- 5.1 ИС СОРМ должны подключаться к ПУ в точках подключения выделенными каналами связи. Интерфейс в точке подключения должен соответствовать спецификации Ethernet 10/100/1000 Base-T либо 100 Base-FX, 1000 Base-LN по согласованию с уполномоченными органами.
- 5.2 В качестве протокола передачи данных должен использоваться протокол TCP/IP.
- 5.2.1 Сетевой интерфейс первого уровня с протоколом TCP/IP используется для первичного подключения аппаратуры передачи данных – модемов, маршрутизаторов и т.п.
- 5.2.2 Для каждой ИС СОРМ внутри сетевого интерфейса первого уровня рекомендуется создавать свою виртуальная сеть VPN (Virtual Private Network) для туннелирования всего рабочего TCP/IP трафика между ИС СОРМ и пунктом управления, которая должна соответствовать спецификации L2TP, IPSec VPN.
- 5.3 Пропускная способность каналов передачи данных между ИС СОРМ и ПУ в точках подключения соответствует данным, приведенным в Таблице 2:

Таблица 2

Емкость абонентской базы (тыс. абонентов), не более	1	10	100	1000	10000
Скорость передачи данных (кбит/сек), не менее	2048	4096	10000	10000	100000

- 5.4 Взаимодействие ИС СОРМ с оборудованием ПУ на транспортном уровне осуществляется по схеме «клиент – сервер»:
- в качестве «сервера» должна выступать ИС СОРМ;
  - в качестве «клиента» выступает оборудование ПУ.
- 5.4.1 Для взаимодействия ИС СОРМ и ПУ организованы четыре канала передачи данных:
- кпд 1 – канал управления;
  - кпд 2 – канал данных;
  - кпд 3 – канал мониторинга;
  - кпд 4 – канал неформатированных данных.

- 5.4.2 Канал управления (кпд 1) служит для передачи:
- от ПУ в ИС запросов (команд);
  - от ИС СОРМ на ПУ ответов и «сигналов».
- 5.4.3 Канал данных (кпд 2) служит для передачи:
- от ИС СОРМ на ПУ блоков данных отчетов генерируемых ИС в качестве ответов на запросы из ПУ, «сигнала» heartbeat;
  - от ПУ в ИС СОРМ подтверждений о принятии блоков данных отчетов.
- 5.4.4 Канал мониторинга (кпд 3) служит для передачи:
- от ПУ в ИС СОРМ запросов о текущей конфигурации оборудования, системного и прикладного ПО ИС СОРМ и запросов на модификацию конфигурации;
  - от ИС СОРМ на ПУ ответов на запросы ПУ, «сигналов».
- 5.4.5 Канал неформатированных данных (кпд 4) служит для передачи:
- от ПУ в ИС СОРМ команд на виды передаваемых неформатированных данных;
  - от ИС СОРМ на ПУ неформатированных данных, «сигналов».
- 5.4.6 кпд 1, кпд 2, кпд3, кпд4 представляют собой TCP-соединения, создаваемые для подключения на определяемые оператором связи TCP-порты оборудования ИС СОРМ. ИС СОРМ выполняет прослушивание данных портов для создания TCP-соединений с ПУ.
- 5.5 ПУ выполняет попытки установления подключения с ИС СОРМ в соответствии с задаваемым интервалом по предоставленным ИС СОРМ TCP-портам.
- 5.5.1 Установление ПУ соединений с ИС СОРМ по каналам кпд1, кпд2 осуществляется в следующей последовательности:
- ПУ устанавливает TCP-соединение с ИС СОРМ по порту канала кпд1;
  - выполняется процедура взаимной SSL/TLS аутентификации в соответствии с п. 5.19;
  - ПУ устанавливает TCP-соединение с ИС СОРМ по порту канала кпд2;
  - выполняется процедура взаимной SSL/TLS аутентификации в соответствии с п. 5.19;
  - после успешной аутентификации ПУ выполняет создание сессии с ИС СОРМ;
  - ПУ должен ожидать данных и сигналов по кпд2 только после того, как была создана сессия по каналу кпд1. При приеме данных и сигналов по кпд2 при отсутствии установленной сессии по кпд1 ПУ должен разрывать соединение по кпд2.

- 5.5.2 Установление ПУ соединений с ИС СОРМ по каналам кпд3 и кпд4 осуществляется независимо друг от друга и от наличия установленных соединений по каналам кпд1 и кпд2:
- ПУ устанавливает TCP-соединение с ИС СОРМ по TCP-порту канала кпд3/кпд4;
  - выполняется процедура взаимной SSL/TLS аутентификации в соответствии с п. 5.19;
  - после успешной аутентификации ПУ выполняет создание сессии с ИС СОРМ.
  - после успешной аутентификации ПУ посылает на ИС СОРМ команды в соответствии с Приложение №7.
- 5.6 ПУ разрывает соединения с ИС СОРМ, 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» ИС СОРМ разрывает соединения с ПУ по соответствующему каналу.
- 5.7 ИС СОРМ должна обеспечивать одновременное подключение нескольких ПУ. Максимальное количество одновременно подключенных ПУ для одной ИС СОРМ – 100. ИС СОРМ обслуживает подключенные ПУ независимо друг от друга.
- 5.8 В ИС СОРМ на одновременном выполнении могут находиться не менее 50 одновременно запущенных задач, осуществляющих подготовку данных.
- 5.9 Каждому идентифицированному на ИС СОРМ ПУ назначается приоритет выполнения задач. В случае поступления на ИС СОРМ 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. ИС СОРМ должна обеспечивать возможность конфигурирования приоритетов ПУ. Приоритеты ПУ на ИС СОРМ сообщаются оператору связи внесистемными средствами. ИС СОРМ должна обеспечивать одновременное выполнение задач по каждому идентифицированному ПУ:
- по соединениям абонентов – не менее 3;
  - по совершенным платежам – не менее 2;
  - по пополнению справочников – не менее 1;
  - по принадлежности абонентов – не менее 4.

- 5.10 Единицей обмена в кпд 1, кпд 2, кпд 3 и кпд4 является «Сообщение» (Message). Форматы «Сообщений» описаны в Приложение №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.
- 5.11 Интерфейс взаимодействия между ПУ и ИС СОРМ предусматривает наличие следующих видов «Сообщений»:
- «запросы» – передаются от ПУ в ИС СОРМ по кпд 1;
  - «ответы» – передаются из ИС СОРМ на ПУ по кпд 1;
  - «сигналы» – передаются из ИС СОРМ на ПУ по кпд 1 и кпд 2 (для кпд2 только «сигнал» Heartbeat);
  - «отчеты» - формируются ИС СОРМ в качестве ответов на запросы из ПУ, передаются на ПУ по кпд 2;
  - «подтверждения» о принятии «отчетов» – передаются из ПУ в ИС СОРМ по кпд 2;
  - «подтверждения» о принятии «сигналов» - передаются из ПУ в ИС СОРМ по кпд1 и кпд2 (для кпд2 только для «сигнала» Heartbeat).
- 5.12 Интерфейс взаимодействия между ПУ и ИС СОРМ по каналу кпд3 предусматривает наличие следующих видов «Сообщений»:
- запросы;
  - ответы;
  - сигнал Heartbeat;
  - подтверждения о принятии сигнала Heartbeat и ответов.
- 5.13 Интерфейс взаимодействия между ПУ и ИС СОРМ по каналу кпд4 предусматривает наличие следующих видов «Сообщений»:
- «запросы» – передаются от ПУ в ИС СОРМ;
  - «ответы» – передаются из ИС СОРМ на ПУ;
  - «сигналы» – передаются из ИС СОРМ на ПУ;
  - «отчеты» - формируются ИС СОРМ в качестве ответов на запросы из ПУ;
  - «подтверждения» о принятии «отчетов» – передаются из ПУ в ИС СОРМ;
  - «подтверждения» о принятии «сигналов» - передаются из ПУ в ИС СОРМ.
- 5.14 ИС СОРМ выполняет любые действия, связанные с выдачей информации о пользователях услуг связи и предоставленных им услугах связи, управлением и мониторингом КТС и ПО ИС СОРМ только по «запросам» ПУ.
- 5.14.1 ИС СОРМ 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 создание задачи по обработке неформализованных данных» (NonFormalizedTaskRequest).

- 5.14.2 ИС СОРМ должна обеспечивать одновременную передачу блоков данных отчетов для нескольких завершенных поисковых задач по каналу кпд2. ИС СОРМ должна обеспечивать возможность многократной передачи отчетов выполненных задач на ПУ.
- 5.14.3 По запросу ПУ «Запрос загрузки данных» канала кпд1 ИС СОРМ должна обеспечивать передачу блоков отчетов по каналу кпд2 по запрошенной задаче без внесения дополнительных временных задержек между операциями по получению записей результата задачи и преобразования их в блоки.
- 5.14.4 На каждый «запрос» по кпд 1 ИС СОРМ на ПУ посылается «ответ» о принятии к обработке этого запроса. ИС СОРМ обеспечивает 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 создания задачи по обработке неформализованных данных» (NonFormalizedTaskResponse).
- 5.14.5 По «Запросу на создание задачи по обработке информации» ИС СОРМ обеспечивает подготовку и выдачу информации из БД ИС СОРМ, для следующих групп задач:
- «Задачи пополнения справочников (нормативно-справочная информация)» (DictionaryTask);
  - «Задачи поисков по принадлежности абонентов» (AbonentsTask);
  - «Задачи поисков по соединениям абонентов» (ConnectionsTask);
  - «Задачи получения данных местоположения абонентов» (LocationTask);



- «Задачи поисков по совершенным платежам» (PaymentsTask);
- «Задачи поисков по усечению отобранных данных» (CombinationTask);
- «Задачи предоставления сведений о наличии данных» (PresenseTask).

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

- пучки соединительных линий;
- базовые станции;
- роуминговые партнеры;
- коммутаторы;
- IP шлюзы;
- типы вызовов;
- список ДВО;
- способы оплаты (пополнения баланса);
- причины завершения соединения;
- справочник с планами IP-адресации;
- план телефонной номерной емкости;
- типы документов, удостоверяющих личность;
- операторы связи, обслуживаемые ИС СОРМ;
- идентификаторы точек подключения к сети передачи данных, с которых получены записи о соединениях;
- специальные номера оператора связи;
- карта связей пучков соединительных линий;
- план нумерации идентификаторов мобильных телефонных абонентов.

В группу «Задачи поисков по принадлежности абонентов» входят:

- «Задача на поиск информации о принадлежности идентификаторов абонентов сети оператора связи» (ValidateAbonentsTask);
- «Задача на поиск информации об идентификаторах абонентов сети оператора связи зарегистрированных на физическое или юридическое лицо» (ValidateIdentifiersTask);
- «Задача на поиск информации о доступных абоненту видам услуг связи» (ValidateServicesTask).

По задачам поиска принадлежности идентификаторов абонента и об идентификаторах абонентов, зарегистрированных на физическое или юридическое лицо, ИС СОРМ вместе с информацией об абоненте и зарегистрированных идентификаторах должна предоставлять сведения о текущих подключенных абонентом услугах связи.

По «Задаче на поиск информации о доступных абоненту видах услуг связи» ИС СОРМ должна формировать отчет, содержащий

список с историей подключенных абонентом услуг связи за весь период накопления данных. Поисковым критерием задачи должен являться идентификатор абонента, не содержащий символов маскирования.

В группу «Задачи поисков по соединениям абонентов» входят:

- «Задача на поиск по соединениям абонентов» (ValidateCallsTask), в т.ч.:
  - «Задача на поиск соединений абонентов» (ValidateConnectionsTask);
  - «Задача на поиск соединений между абонентами сети передачи данных» (ValidateDataTask);
  - «Задача на поиск информации о появлении абонента в сети связи (выхода на связь)» (ValidateEntranceTask).

В случае, если в качестве параметров «задачи на поиск соединений абонентов» указаны только диапазон времени и (опционально) филиал оператора связи, ИС СОРМ должна сформировать результат выполнения поисковой задачи, содержащий все соединения всех абонентов за указанный диапазон времени.

В группу «Задачи поисков по совершенным платежам» входят:

- «Задача на поиск пополнения баланса через банковский перевод» (bankTransactionTask);
- «Задача на поиск пополнения баланса через карты экспресс-оплаты» (expressCardTask);
- «Задача на поиск пополнения баланса через терминалы моментальных платежей» (publicTerminalTask);
- «Задача на поиск пополнения баланса через центры обслуживания клиентов (ЦОК)» (serviceCenterTask);
- «Задача на поиск пополнения баланса посредством снятия денег со счета другого абонента» (crossAccountTask);
- «Задача на поиск пополнения баланса через телефонные карты» (telephoneCardTask);
- «Задача на поиск перевода средств со счета абонента для их снятия в отделении банка» (bankDivisionTransferTask);
- «Задача на поиск перевода средств со счета абонента на банковскую карту» (bankCardTransferTask);
- «Задача на поиск перевода средств со счета абонента на счет в банке» (bankAccountTransferTask);
- «Общая задача на поиск пополнения баланса личного счета абонента» (balanceFillupTask).

По «Задаче получения данных местоположения абонентов» ИС СОРМ за запрошенный период времени ИС СОРМ должна предоставлять накопленную в БД ИС СОРМ информацию об изменении

местоположения абонентского устройства (историю) по запрошенному идентификатору.

В группу «Задачи поисков по усечению отобранных данных» (CombinationTask) входят (в соответствии с CombinationTask.asp Приложения №8):

- «Задача на усечение выборки по критериям» (ValidateCombinationFilter).

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

- запрос наличия информации по абонентам и их идентификаторам;
- запрос наличия информации по соединениям;
- запрос наличия имеющейся информации по платежам;
- запрос наличия справочников;
- запрос наличия информации по местоположениям абонентов.

5.14.6 По «Запросу на создание задачи по обработке неформализованных данных» ИС СОРМ обеспечивает подготовку и выдачу информации из БД ИС СОРМ по следующим видам запросов:

- «запрос получения списка типов сущностей» неформализованных данных ИС СОРМ (GetEntities);
- «запрос получения списка атрибутов сущности» (GetEntityAtttributes);
- «задача поиска неформализованных данных» (ValidateNonFormalizedTask);
- «задача предоставления сведений о наличии неформализованных данных» (NonFormalizedPresenseTask).

5.14.7 ИС СОРМ по «задаче поиска неформализованных данных» может предоставлять доступ ПУ к системным журналам ИС СОРМ.

5.14.8 ИС СОРМ должна обеспечивать выполнение поисковых задач по строковым итериям, содержащими символы маскирования, включающие:

- «\*» – обозначает любую комбинацию символов;
- «?» – обозначает любой один возможный символ.

ИС СОРМ должна 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.

- 5.14.9 В случае возникновения в ИС СОРМ исключительных ситуаций на ПУ передаются следующие «Сообщения» типа «Сигналы», содержащие информацию об уровне важности исключительной ситуации, ее влиянии на сохранность данных и выполнение задач:
- «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 ИС СОРМ, ПУ должно передавать «подтверждения сигнала» об их приеме.

5.14.10 ИС СОРМ по «Запросу удаления данных» (DataDropRequest) должна:

- прерывать задачу, находящуюся на выполнении;
- удалять данные отчета по завершившейся задаче.

5.15 Требования к функционированию канала кпд1 приведены в Приложение №3.

5.16 Требования к функционированию канала кпд2 приведены в Приложение №4.

5.17 Требования к функционированию канала кпд3 приведены в Приложение №5.

5.18 Требования к функционированию канала кпд4 приведены в Приложение №6.

5.19 При установлении соединения ПУ и ИС СОРМ должны быть взаимно аутентифицированы. Аутентификация выполняется установлением SSL/TLS-соединения поверх установленного TCP-соединения между ПУ и ИС СОРМ. Для взаимной аутентификации ПУ и ИС СОРМ предварительно создаются X.509-сертификаты, которые сообщаются ПУ и ИС СОРМ. В случае невозможности аутентифицировать одну из сторон TCP-соединение разрывается. Созданный для ПУ сертификат используется для аутентификации только одного данного ПУ на одной ИС СОРМ по всем каналам передачи данных — кпд1, кпд2, кпд3, кпд4. ПУ и ИС СОРМ должны использовать TLS версии 1.0. Требования к сертификатам (длины ключей, прочие параметры) должны согласовываться для каждой пары ИС СОРМ и ПУ отдельно.

5.20 Открытие сессии осуществляется выполнением процедуры аутентификации согласно п. 5.19 перед началом выполнения всех запросов. Открытие сессии осуществляется посылкой по каналу управления от ПУ к ИС СОРМ «Запроса на открытие сессии» (ConnectRequest).

5.21 «Запрос на открытие сессии» устанавливает следующие параметры сессии:

- «максимальное время отсутствия активности сессии» (session-timeout) – интервал времени, по истечении которого сессия принудительно прерывается от ИС СОРМ;
- «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 ИС СОРМ и ПУ (Приложение №8) считается сетевой активностью.

5.21.1 ИС СОРМ при получении сообщения «Запрос на открытие сессии», анализирует параметр «размер окна для канала передачи данных», определяет максимально возможный размер окна, не превышающий полученного от ПУ. ИС СОРМ определяет 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 ИС СОРМ 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 ИС СОРМ.

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

- 5.23 ПУ посылает ИС СОРМ «запросы» асинхронно, независимо от получения от ИС СОРМ «ответа» о приеме предыдущего «запроса».
- 5.24 «Запрос на создание задачи по обработке информации» приводит к созданию в ИС СОРМ задачи по обработке данных в БД ИС СОРМ, которой присваивается номер (идентификатор) задачи, передаваемый в «Ответе на запрос создания задачи» (TaskResponse). Идентификаторы задач генерируются ИС СОРМ независимо от сессий и являются уникальными в данной ИС СОРМ. ИС СОРМ присваивает идентификаторы задачам и выполняет обработку задач независимо для различных ПУ.
- 5.25 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.
- 5.26 ИС СОРМ при получении «Запрос загрузки данных» по кпд 1 формирует сообщения типа - «отчет», состоящие из блоков данных обработанной задачи и передает их на ПУ по кпд2.
- 5.27 Количество строк в каждом блоке не превышает параметр «максимальный размер блока данных отчетов», заданный при открытии сессии.
- 5.28 В каждом последовательном блоке данных из серии указываются идентификатор задачи, сгенерировавшей данный отчет, общее количество блоков в отчете, порядковый номер каждого блока.
- 5.29 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, ИС СОРМ посылает 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 ИС СОРМ посылает на этот ПУ «сигнал» попытки несанкционированного доступа (unauthorized-access) и ожидает его подтверждения.
- 5.30 ИС СОРМ производит уничтожение данных, сформированных в результате выполнения задачи и самой выполненной задачи после поступления с ПУ запроса на удаление данных.
- 5.31 В случае возникновения в ИС СОРМ или каналах передачи данных исключительных ситуаций на ПУ передаются «Сообщения» типа «Сигнал».

6 Требования, предъявляемые к функционированию канала кпд1.

6.1.1 Диаграмма состояний переходов ИС СОРМ по кпд1 приведена на  
 6.1.2 Рисунок 1.

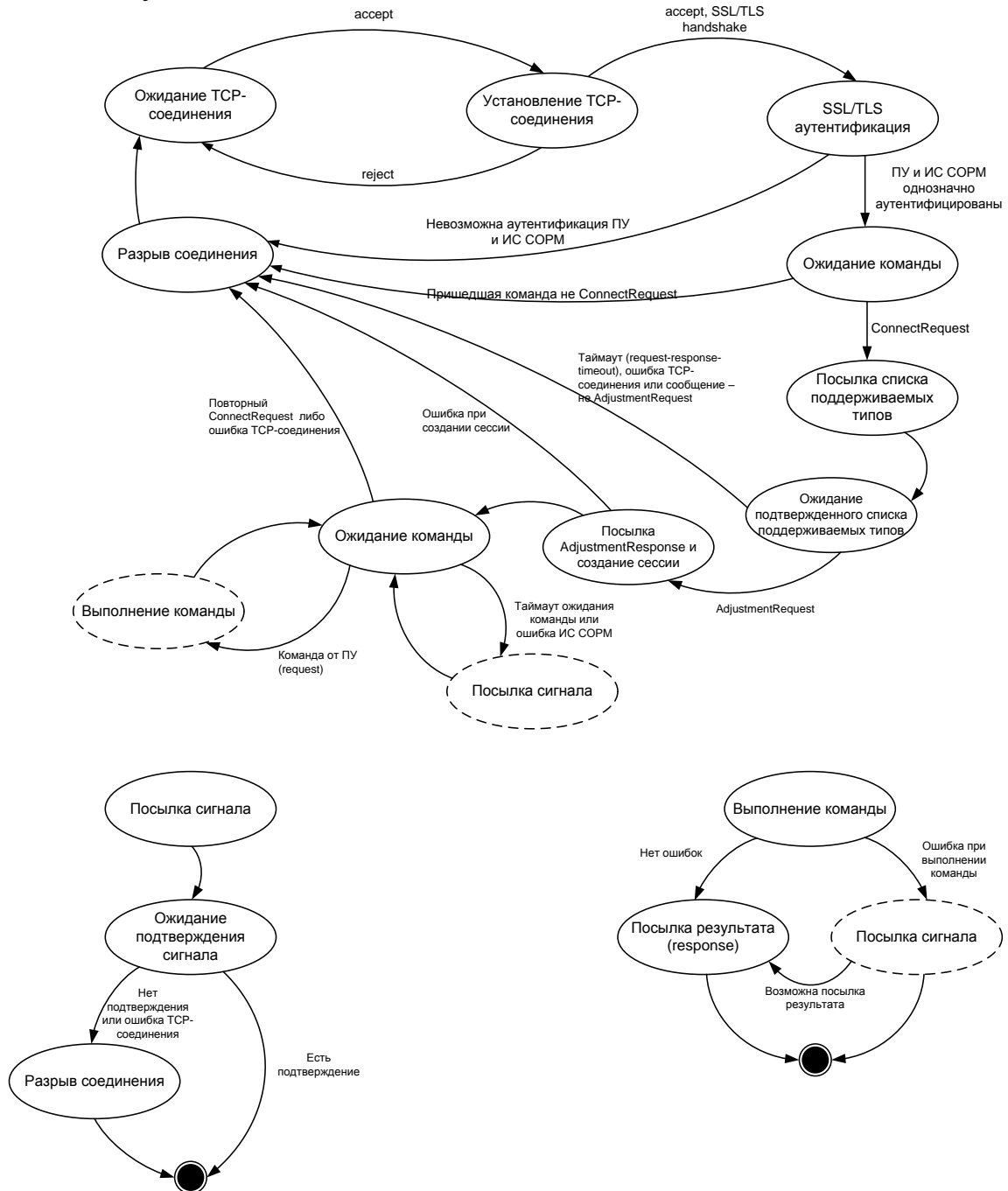


Рисунок 1. Диаграмма состояний перехода ИС СОРМ по кпд1.

6.1.2.1 ИС СОРМ по TCP-порту кпд1 ожидает входящих соединений. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

- 6.1.2.2 Если SSL/TLS-соединения по каналам управления и данных установлены, а сессия не открыта, ИС СОРМ должна реагировать только на сообщение «Запрос на открытие сессии». При попытке посылок каких-либо других сообщений со стороны ПУ ИС СОРМ должна разорвать TCP соединения по каналу управления и каналу данных и перевести канальный уровень подключения в исходное состояние.
- 6.1.2.3 При получении сообщение «Запрос на открытие сессии» ИС СОРМ создает список поддерживаемых ИС СОРМ типов запросов, отчетов и сигналов (в том числе и предыдущих версий) и отправляет его.
- 6.1.2.4 После отсылки списка поддерживаемых типов, ИС СОРМ ожидает от ПУ списка поддерживаемых им запросов, отчетов и сигналов. Список поддерживаемых ПУ типов должен являться подмножеством списка типов в ИС СОРМ.
- 6.1.2.5 При получении от ПУ списка поддерживаемых ПУ запросов ИС СОРМ посылает сообщение «Ответ на согласование списка поддерживаемых типов» и создает сессию.
- 6.1.2.6 После создания сессии кпд1 ИС СОРМ переводится 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.
- 6.1.2.7 В случае серьезного сбоя в ИС СОРМ, вызванного причинами, не предусмотренными режимом нормального функционирования системы, по каналу управления передается «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 ИС СОРМ требуется ее перезагрузка, то по каналу управления выдается прерывание «Перезапуск ПО». В этом случае ИС СОРМ и ПУ закрывают все открытые на текущий момент сессии.
- 6.1.2.8 В случае наличия признаков сбоя или ошибки выполнения конкретной задачи ИС СОРМ в режиме нормального функционирования, по каналу управления передается 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.
- 6.1.2.9 На каждый «сигнал», переданный ИС СОРМ, ПУ должно ответить сообщением «подтверждение сигнала» по кпд1. Отсутствие подтверждения в течение времени RequestResponseTimeout, которое



задается при открытии сессии, свидетельствует о прерывании соединения и должно вызывать действия, описанные в п. 5.22.

6.1.2.10 При отсутствии команд ПУ в течение «максимального времени неактивности» (session-timeout, задается в ConnectRequest), ИС СОРМ посылает на ПУ «сигнал» heartbeat и ожидает его подтверждения аналогично описанному в п. 6.1.2.9.

6.1.3 Диаграмма состояний переходов ПУ по кпд1 приведена на Рисунок 2.

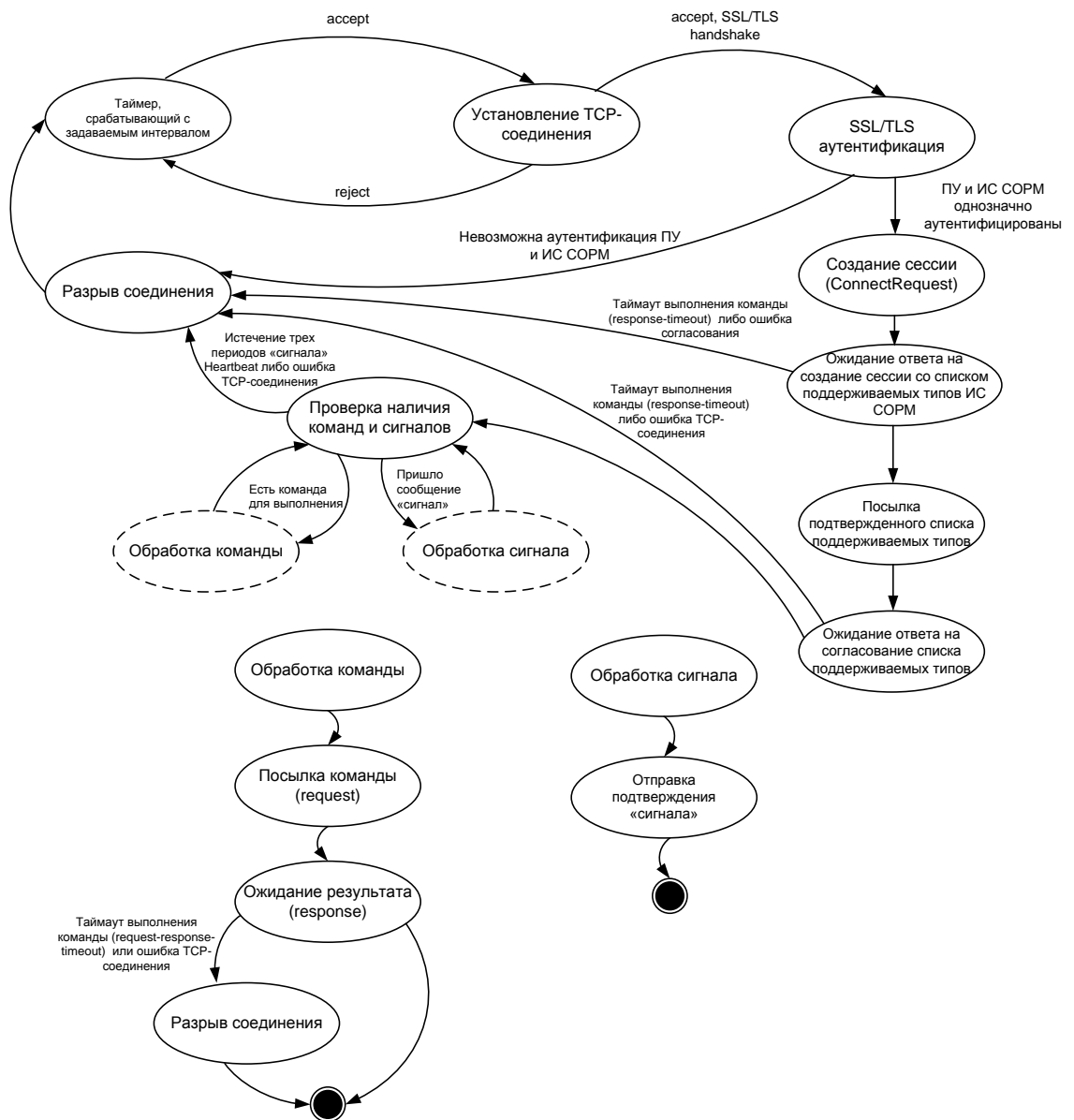


Рисунок 2. Диаграмма состояний перехода ПУ по кпд1.

6.1.3.1 ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ИС СОРМ по заданному порту кпд1. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

6.1.3.2 ПУ ожидает установления соединения по кпд2.

- 6.1.3.3 После установления TCP-соединения по кпд2 и прохождения по нему взаимной аутентификации ИС СОРМ и ПУ, ПУ по кпд1 посылает команду создания сессии (ConnectRequest).
- 6.1.3.4 ПУ ожидает сообщения «ответ» от ИС СОРМ на отправленную команду в течение времени «таймаут ответа на запрос» (request-response-timeout).
- 6.1.3.5 Если сообщение не получено в течение времени «таймаут ответа на запрос», ПУ разрывает соединения с ИС СОРМ по кпд1 и кпд2 и переводит их в изначальное состояние (п. 6.1.3.1). Время ожидания сообщения «ответ» не зависит от приема сообщений «сигналов», поступающих в этот интервал времени.
- 6.1.3.6 При получении сообщения «Ответ на запрос создания сессии» ПУ создает список поддерживаемых ПУ типов запросов, отчетов и сигналов и отправляет его ИС СОРМ. Список поддерживаемых ПУ типов должен быть подмножеством списка типов ИС СОРМ, полученных в п. 6.1.3.4.
- 6.1.3.7 После отправки сообщения «Согласование поддерживаемых типов со стороны ПУ» ПУ ждет ответа на сообщение от ИС СОРМ. Поведение ПУ при ожидании ответа аналогично описанному в п. 6.1.3.5.
- 6.1.3.8 Если во время ожидания сообщения «ответ» ИС СОРМ посылает на ПУ сообщение «сигнал» (в т.ч. heartbeat), то ПУ посылает «подтверждение» о принятии «сигнала» (в т.ч. heartbeat) и продолжает ожидать сообщения «ответ» на отосланную команду.
- 6.1.3.9 После создания сессии ПУ посылает поступающие команды на ИС СОРМ и ожидает сообщений «ответов» на них в соответствии с п. 6.1.3.4, 6.1.3.5, 6.1.3.8.
- 6.1.3.10 Если при ожидании поступления в ПУ команд ИС СОРМ не посылала «сигнал» heartbeat в течение трех интервалов «максимального времени неактивности» (session-timeout, задается в ConnectRequest) ПУ разрывает соединения с ИС СОРМ по кпд1 и кпд2 и переводит их в изначальное состояние (п. 6.1.3.1).
- 6.1.3.11 При поступлении команды «запрос на закрытие сессии» (DisconnectRequest), ПУ отсылает ее на ИС СОРМ, ожидает сообщения «ответ» в соответствии с п. 6.1.3.4, 6.1.3.5, 6.1.3.8, после чего разрывает соединение по кпд2 и кпд1, и переводит их в изначальное состояние (п. 6.1.3.1).

7 Требования, предъявляемые к функционированию канала кпд2.

7.1 ИС СОРМ обеспечивает подключение ПУ и обработку поступающих запросов по каналу кпд2 в соответствии с приведенными диаграммами состояний переходов.

7.1.1 Диаграмма состояний переходов ИС СОРМ по кпд2 приведена на Рисунок 3.

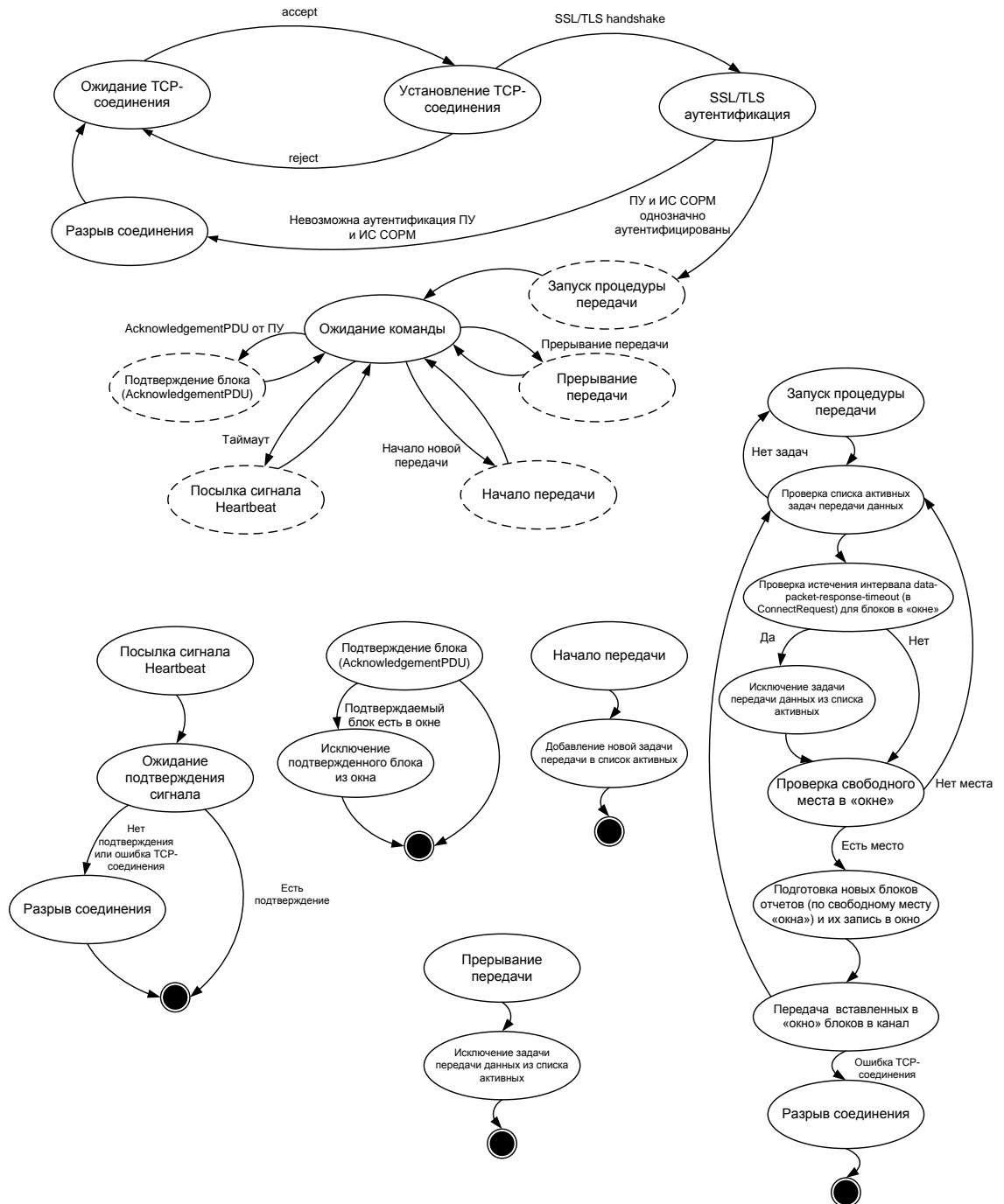


Рисунок 3. Диаграмма состояний перехода ИС СОРМ по кпд2.

- 7.1.1.1 ИС СОРМ по TCP-порту кпд2 ожидает входящих соединений. После установления соединения выполняется взаимная SSL/TLS-аутентификация.
- 7.1.1.2 Если на ИС СОРМ была передана «запрос загрузки данных» (DataLoadRequest), ИС СОРМ посылает «ответ на запрос загрузки данных» (DataLoadResponse) по каналу кпд1 и начинает передачу данных блоков отчетов по кпд2 при их наличии. ПУ может получить блоки отчетов по кпд2 до получения «ответа на запрос загрузки данных» (DataLoadResponse) по кпд1.
- 7.1.1.3 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 ИС СОРМ выполняет подготовку новых блоков отчетов по загружаемым задачам и посылает на их на ПУ. Количество подготовленных и переданных без получения «подтверждения» блоков не должно превышать размера окна канала передачи данных.
- 7.1.1.4 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.
- 7.1.1.5 При получении «подтверждения» блока «отчета» ИС СОРМ должна записать информацию об ошибочно принятом ПУ блоке и ошибочных записях в блоке в специальный журнал, передача последующих блоков по задаче на ПУ не прерывается. ИС СОРМ должна 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).
- 7.1.1.6 В случае разрыва TCP/IP соединения кпд2, при существующем соединении кпд1, по кпд1 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 кпд2 в соответствии с п. 5.19.

- 7.1.1.7 Передача блоков данных может быть прервана в случае получения ИС СОРМ «запроса прерывания загрузки данных».
- 7.1.1.8 Если по кпд2 не производится передача блоков отчетов в течение «максимального времени неактивности» (session-timeout при создании сессии ConnectRequest), ИС СОРМ посылает на ПУ «сигнал» heartbeat и ожидает его подтверждения аналогично описанному в п. 6.1.2.9.
- 7.1.2 Диаграмма состояний переходов ПУ по кпд2 приведена на Рисунок 4.

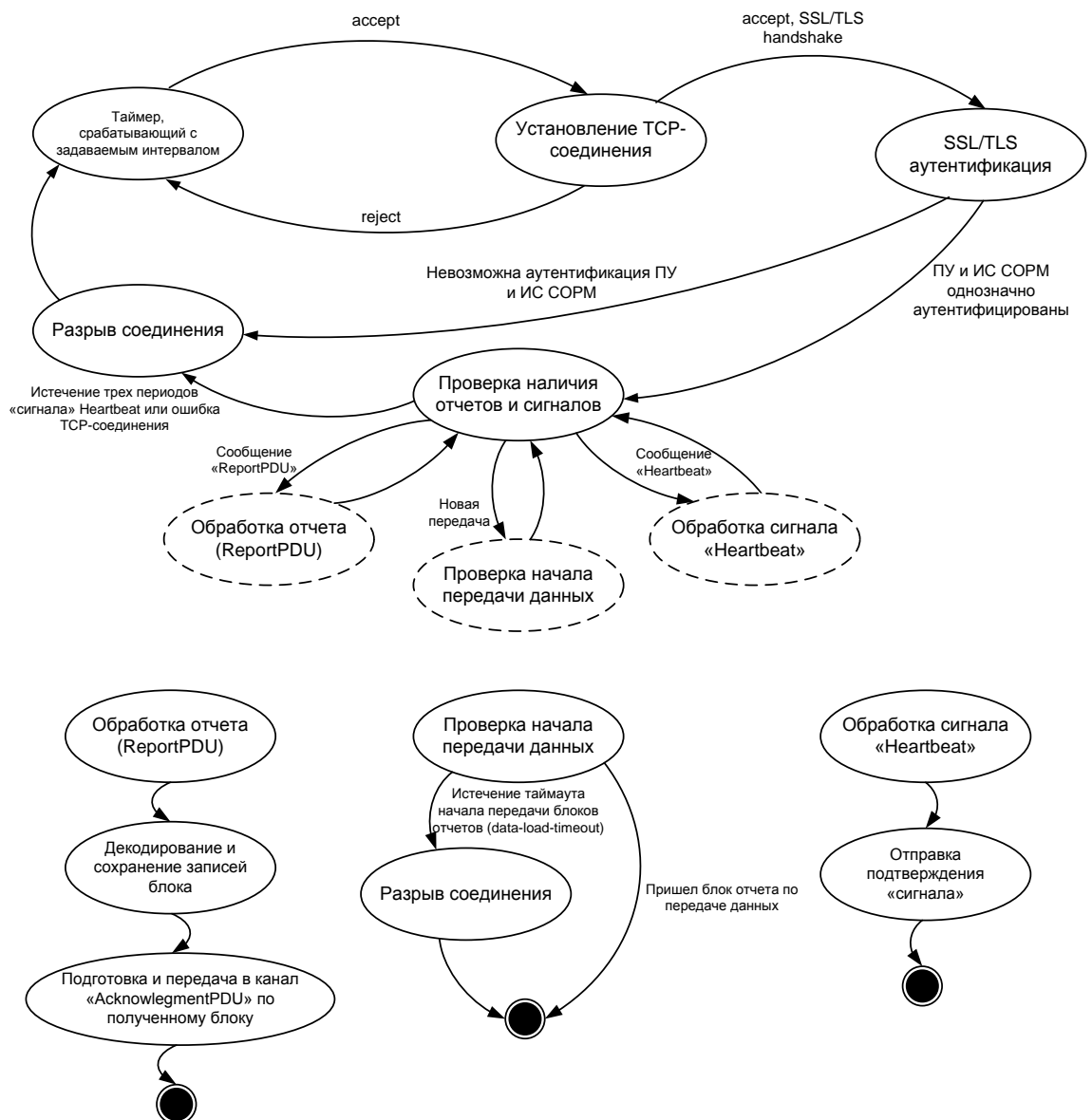


Рисунок 4. Диаграмма состояний перехода ПУ по кпд2.

- 7.1.2.1 ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ИС СОРМ по заданному порту кпд2. После

установления соединения выполняется взаимная SSL/TLS-аутентификация.

- 7.1.2.2 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 кпд1 и кпд2 и переводит соединения в изначальное состояние (п. 6.1.3.1, 7.1.2.1).
- 7.1.2.3 При поступлении блока отчета ПУ производит декодирование полученного блока и сохранение декодированных данных.
- 7.1.2.4 В ответ на переданный блок данных ПУ посылает сообщение «подтверждение» получения блока отчета. Количество последовательно переданных ИС СОРМ 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.
- 7.1.2.5 Если при ожидании поступления в ПУ блоков отчетов ИС СОРМ не посылала «сигнал» heartbeat в течение трех интервалов «максимального времени неактивности» (session-timeout, задается в ConnectRequest) ПУ разрывает соединения с ИС СОРМ по кпд1 и кпд2 и переводит их в изначальное состояние (п. 6.1.3.1, 7.1.2.1).

## 8 Требования, предъявляемые к функционированию канала кпдЗ.

- 8.1 ИС СОРМ обеспечивает подключение ПУ и обработку поступающих запросов по каналу кпдЗ (канал мониторинга).
- 8.1.1 ИС СОРМ обеспечивает прием и обработку следующих видов запросов от ПУ по каналу кпдЗ (канал мониторинга):
- запрос на получение структуры ИС СОРМ - КТС и списка модулей прикладного ПО ИС СОРМ («GetStructureRequest»);
  - запрос на получение конфигурации КТС/модуля ПО ИС СОРМ («GetModuleConfigRequest»);
  - запрос на изменение конфигурации КТС/модуля ПО ИС СОРМ («SetModuleConfigRequest»);
  - запрос на получение состояния модуля КТС/модуля ПО ИС СОРМ («CheckModuleRequest»).
- 8.1.2 На каждый запрос по кпдЗ ИС СОРМ посылает ответ на ПУ, содержащий результат обработки соответствующего запроса - «ManagementResponse».
- 8.1.3 Диаграммы состояний перехода ИС СОРМ и ПУ по кпдЗ соответствуют диаграммам для кпд1 (
- 8.1.4 Рисунок 1, Рисунок 2).
- 8.1.4.1 ИС СОРМ по TCP-порту кпдЗ ожидает входящих соединений. После установления соединения выполняется взаимная SSL/TLS-аутентификация.
- 8.1.4.2 Если SSL/TLS-соединение по кпдЗ установлено, а сессия не открыта, ИС СОРМ должна реагировать только на сообщение «Запрос на открытие сессии». Создание сессии аналогично описанному в пп. 6.1.2.3-6.1.2.5. При попытке посылок каких-либо других сообщений со стороны ПУ ИС СОРМ должна разорвать TCP соединение по кпдЗ и перевести канальный уровень подключения в исходное состояние.
- 8.1.4.3 После создания сессии ИС СОРМ переводится в режим ожидания команд от ПУ. Обработка поступающих команд и посылка сигналов производится аналогично описанному в пп. 6.1.2.6-6.1.2.10.
- 8.1.4.4 ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ИС СОРМ по заданному порту. После установления соединения выполняется взаимная SSL/TLS-аутентификация.
- 8.1.4.5 ПУ по кпдЗ посылает команду создания сессии (ConnectRequest).
- 8.1.4.6 Ожидание подтверждения создания сессии, согласование списка поддерживаемых типов, отправка команд, ожидание ответов и

обработка полученных от ИС СОРМ сигналов по кпдЗ производится ПУ аналогично описанному в пп. 6.1.3.4-6.1.3.10.

8.2 ИС СОРМ должна обеспечивать получение следующих видов информации о структуре и функционировании ИС СОРМ по запросу ПУ:

- о структуре и составе КТС ИС СОРМ, составе и состоянии интерфейсов взаимодействия КТС ИС СОРМ;
- об установленном на КТС ИС СОРМ общесистемном и специальном программном обеспечении ИС СОРМ, перечне и состоянии программных модулей в составе специального программного обеспечения ИС СОРМ;
- о точках подключения ИС СОРМ к сети оператора связи и интерфейсах ввода информации в ИС СОРМ.

8.2.1 В части структуры и состава КТС ИС СОРМ, состава и состоянии интерфейсов взаимодействия КТС ИС СОРМ, ИС СОРМ должна по запросу ПУ предоставлять следующую информацию:

- перечень коммутационного и серверного оборудования, средств хранения данных с его идентификацией;
- идентификацию интерфейсов подключения КТС ИС СОРМ 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.

8.2.2 В части точек подключения ИС СОРМ к сети оператора связи, интерфейсов ввода информации в ИС СОРМ, ИС СОРМ должна по запросу ПУ предоставлять текущую информацию на момент формирования запроса, содержащую:

- перечень точек подключения к сети связи и точек ввода информации в ИС СОРМ с их идентификацией;



- для каждой точки подключения предоставлять информацию:
  - вид поступающих в ИС СОРМ сведений (о соединениях абонентах, о платежах, об абонентах и т.д.);
  - 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 ИС СОРМ не поступала информация.

8.2.3 В части состава общесистемного и специального программного обеспечения ИС СОРМ, их текущего состояния, ИС СОРМ по запросу ПУ следующую информацию:

- перечень установленного общесистемного программного обеспечения с его идентификацией;
- предоставление для общесистемного программного обеспечения информации:
  - идентификатора КТС ИС СОРМ, на котором установлено;
  - наименование общесистемного программного обеспечения;
  - текущее состояние (штатное функционирование, сбой, не функционирует), текстовую расшифровку сбоя;
- перечень установленного специального программного обеспечения ИС СОРМ с его идентификацией;
- предоставление для специального программного обеспечения ИС СОРМ информации:
  - идентификатора КТС ИС СОРМ, на котором установлено;
  - назначение (определяется разработчиком ИС СОРМ);
  - текущее состояние (штатное функционирование, сбой, не функционирует), текстовую расшифровку сбоя;
  - список контролируемых параметров (определяется разработчиком ИС СОРМ).

8.2.4 ИС СОРМ должна предоставлять возможность изменения отдельных параметров функционирования КТС ИС СОРМ общесистемного и специального программного обеспечения по запросу ПУ посредством кпдЗ. Перечень доступных ПУ для изменения параметров определяется разработчиком ИС СОРМ.

## 9 Требования, предъявляемые к функционированию канала кпд4.

- 9.1 ИС СОРМ обеспечивает подключение ПУ и обработку поступающих запросов по каналу кпд4 (канал неформатированных данных).
- 9.1.1 ИС СОРМ обеспечивает прием и обработку следующих видов запросов от ПУ по каналу кпд4:
- «запрос проверки наличия вида неформатированных данных в ИС СОРМ» (DataTypesRequest);
  - «запрос на начало передачи неформатированных данных» (DataStartRequest);
  - «запрос на остановку передачи неформатированных данных» (DataStopRequest).
- 9.1.2 На каждый запрос по кпд4 ИС СОРМ посылает ответ на ПУ, содержащий результат обработки соответствующего запроса.
- 9.1.3 ИС СОРМ должна накапливать информацию с неформатированными данными в специальном буфере. Данные в буфере упорядочиваются согласно времени их поступления.

Для телефонных соединений абонентов (в соответствии с лицензиями на оказание услуг связи, перечисленных в п. 1.2 настоящих Требований) длительность временного хранения неформатированных данных в буфере – до 6 мес.

Для соединений, совершенных абонентами посредством сети передачи данных (в соответствии с лицензиями на оказание услуг связи, перечисленных в п. 1.2 настоящих Требований), длительность временного хранения неформатированных данных в буфере – до 10 суток. Данные о соединениях, совершенных абонентами посредством сети передачи данных, должны содержать все сетевые заголовки телекоммуникационных протоколов, использованных при обмене, не включая содержимое пакетов.

9.1.4 ИС СОРМ должна записывать в буфер следующую информацию.

В части телефонных соединений абонентов – файлы, формируемые коммутационным оборудованием, содержащие только записи о соединениях абонентов в не декодированном виде.

В части соединений, совершенных абонентами посредством сети передачи данных – файлы в формате TCPDUMP, содержащие исключительно заголовки пакетов обмена данными, включая прикладной уровень, метку времени формирования пакета, размер пакета.

9.1.5 ИС СОРМ должна записывать информацию в буфер циклически, т.е. при переполнении буфера старая информация в нем перезаписывается.

9.1.6 Диаграмма состояний переходов ИС СОРМ по кпд4 приведена на Рисунок 5.

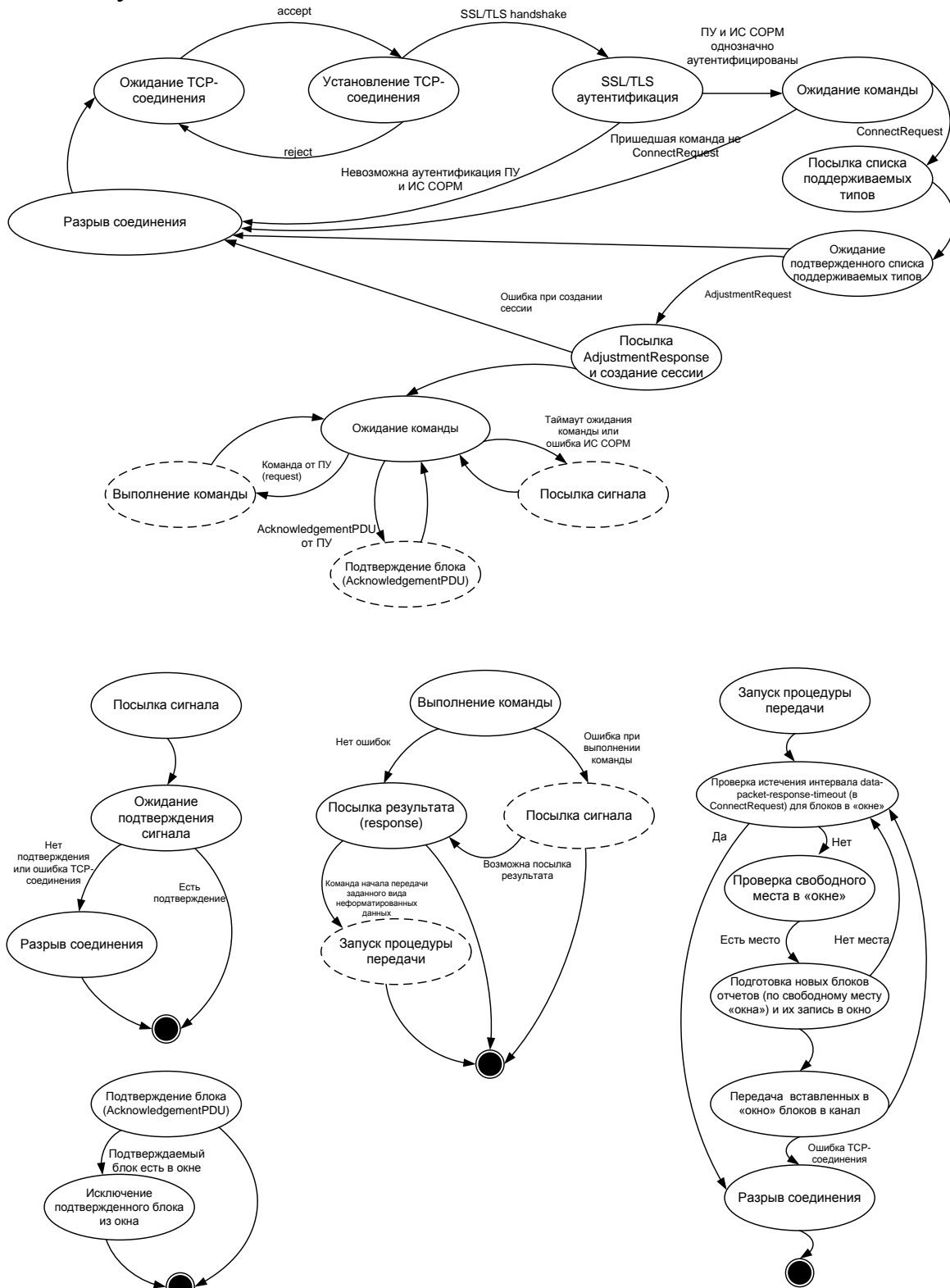


Рисунок 5. Диаграмма состояний перехода ИС СОРМ по кпд4.

- 9.1.6.1 ИС СОРМ ожидает и устанавливает соединение аналогично описанному в пп. 8.1.4.1, 8.1.4.2.
- 9.1.6.2 После создания сессии ИС СОРМ переводится в режим ожидания команд. Обработка команд и посылка «сигналов» осуществляется аналогично описанному в пп. 6.1.2.6 - 6.1.2.10, за исключением команд «запрос на начало, остановку передачи неформатированных данных» (DataStartRequest/ DataStopRequest).
- 9.1.6.3 При приеме команды «запрос на начало передачи неформатированных данных» ИС СОРМ, находясь в режиме ожидания команд, отправляет «ответ на запрос начала передачи неформатированных данных» и переводит канал кпд4 в режим передачи данных, при этом производится передача данных того типа, который указан в команде запроса.
- 9.1.6.4 При приеме команды «запрос на остановку передачи неформатированных данных» ИС СОРМ, находясь в режиме ожидания команд, посылает «ответ на запрос остановки передачи неформатированных данных» с отрицательным результатом.
- 9.1.6.5 В режиме передачи данных посылка блоков отчетов и их подтверждение производится аналогично описанному в пп. 7.1.1.3, 7.1.1.5, 7.1.1.8.
- 9.1.6.6 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, ИС СОРМ разрывает соединение по кпд4 и переводит кпд4 в изначальное состояние.
- 9.1.6.7 Передача неформатированных данных соответствующего типа производится из буфера кпд 4 (см. п. 9.1.3).
- 9.1.6.8 При приеме команды «запрос на остановку передачи неформатированных данных» ИС СОРМ, находясь в режиме передачи данных, посылает «ответ на запрос остановки передачи неформатированных данных», завершает посылку блоков данных и переводится в режим ожидания команд.
- 9.1.6.9 При приеме команды «запрос на начало передачи неформатированных данных» ИС СОРМ, находясь в режиме передачи данных, посылает «ответ на запрос начала передачи неформатированных данных» с отрицательным результатом, при этом передача неформатированных данных не прекращается.
- 9.1.6.10 Исходные данные о соединениях в виде неформатированных данных записываются в буфер независимо от текущего режима работы канала кпд4 ИС СОРМ.
- 9.1.6.11 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. 9.1.3.

### 9.1.7 Диаграмма состояний переходов ПУ по кпд4.

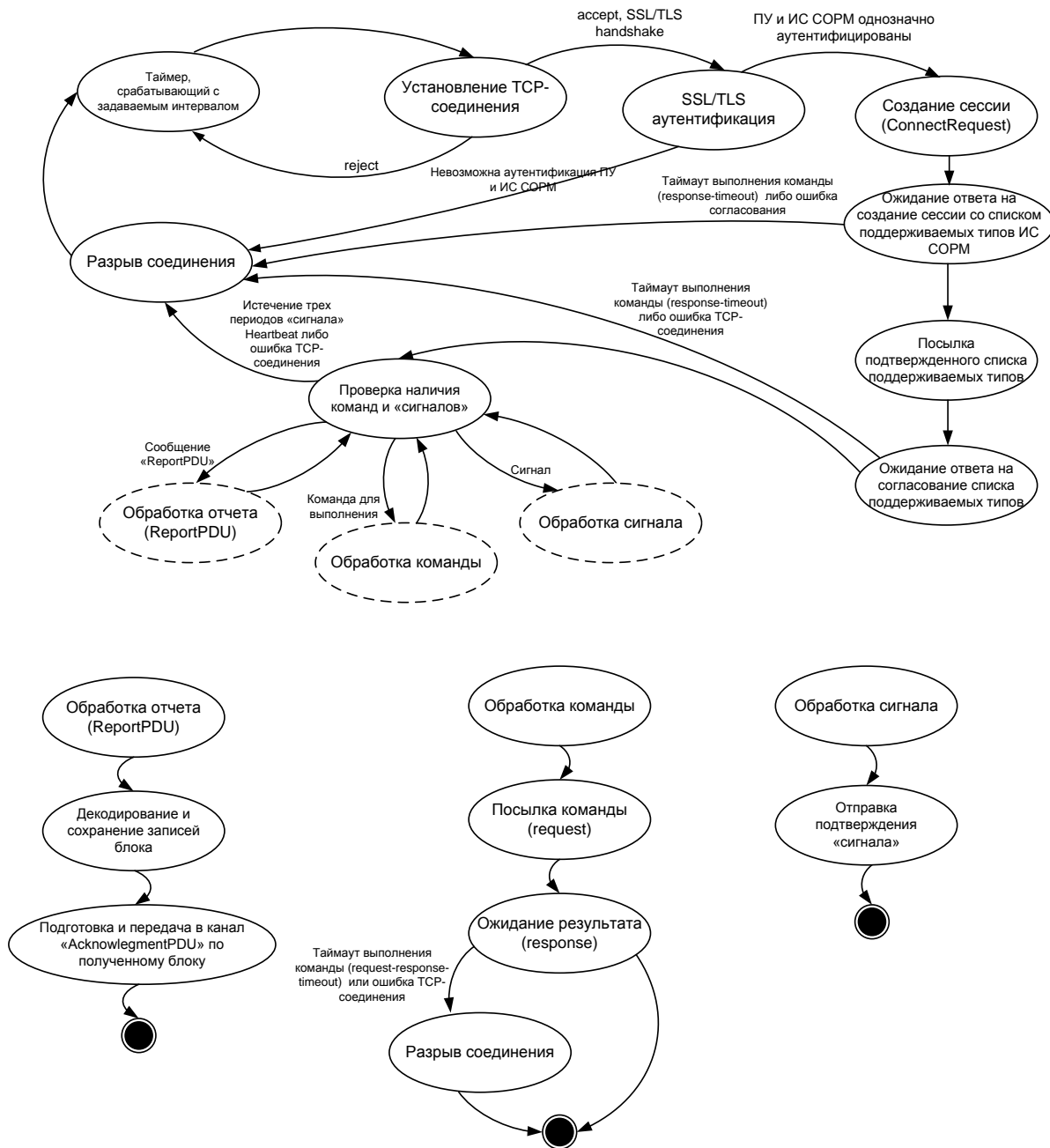


Рисунок 6. Диаграмма состояний перехода ПУ по кпд4.

9.1.7.1 ПУ с задаваемым интервалом выполняет попытки установления TCP-соединения с ИС СОРМ по заданному порту. После установления соединения выполняется взаимная SSL/TLS-аутентификация.

9.1.7.2 ПУ по кпд4 посылает команду создания сессии (ConnectRequest).

- 9.1.7.3 Ожидание подтверждения создания сессии, согласование списка поддерживаемых типов, отправка команд, ожидание ответов и обработка полученных от ИС СОРМ сигналов по кпд4 производится ПУ аналогично описанному в пп. 6.1.3.4 - 6.1.3.10, за исключением команд «запрос на начало, остановку передачи неформатированных данных» (DataStartRequest/ DataStopRequest).
- 9.1.7.4 При посылке команды «запрос на начало/остановку передачи неформатированных данных», ПУ ожидает результата (DataStartResponse) аналогично описанному в пп. 6.1.3.4 - 6.1.3.5, после чего переводит канал кпд4 в режим передачи данных.
- 9.1.7.5 В режиме передачи данных ПУ производит прием, декодирование и подтверждение приема данных аналогично описанному в пп. 7.1.2.3 - 7.1.2.5.
- 9.1.7.6 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.

## 10 Требования к форматам сообщений ИС СОРМ.

- 10.1 Структура разделения ASN.1 модулей протокола взаимодействия ПУ и ИС СОРМ приведена на Рисунок 10.
- 10.2 Модуль Classification.asn задает правила, в соответствии с которыми:
- выполняется расширение:
    - списка типов запросов к ИС СОРМ;
    - списка видов поисковых критериев к ИС СОРМ;
    - списка типов отчетов, формируемых ИС СОРМ;
  - выполняется ввод новых версий сообщений протокола.
- 10.3 Структура разделения ASN.1 «Сообщений» протокола приведена на Рисунок 7.

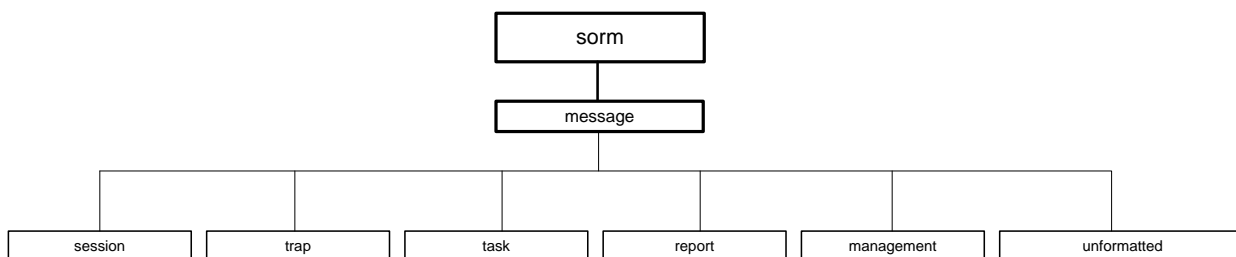


Рисунок 7. Структура видов сообщений протокола взаимодействия ПУ и ИС СОРМ.

- 10.4 В соответствии с Рисунок 8, Рисунок 9 в протокольных ASN.1 модулях выполняется подстановка соответствующих версий форматов поисковых критериев, отчетов, справочников, «сообщений» интерфейса взаимодействия ПУ и ИС СОРМ.
- 10.5 ASN.1–модуль «Classification.asn» содержит кодированные в иерархическом виде идентификаторы:
- видов «Сообщений» верхнего уровня интерфейса взаимодействия ПУ и ИС СОРМ, составляющих кпд1, кпд2, кпд3, кпд4;
  - видов поисковых критериев для формирования задач к ИС СОРМ;
  - видов форматов отчетов, формируемых ИС СОРМ.
- 10.6 Соответствующие идентификаторы используются в других ASN.1-модулях интерфейса взаимодействия ПУ и ИС СОРМ (Рисунок 10 с модулями), при этом идентификатор определяет конкретную версию и расширения формата соответствующего элемента (поисковых

критериев, отчетов, справочников «сообщений» – в соответствии с Рисунок 8, Рисунок 9).

- 10.7 Предоставленный ИС СОРМ при создании сессии перечень идентификаторов и согласованное из него ПУ подмножество в целом определяют конкретные возможности взаимодействия ПУ и ИС СОРМ в соответствии с выбранными идентификаторами.
- 10.8 Расширение интерфейса взаимодействия ПУ и ИС СОРМ 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 8, Рисунок 9 и стандартным кодированием ASN.1-модуля «Classification.asn».



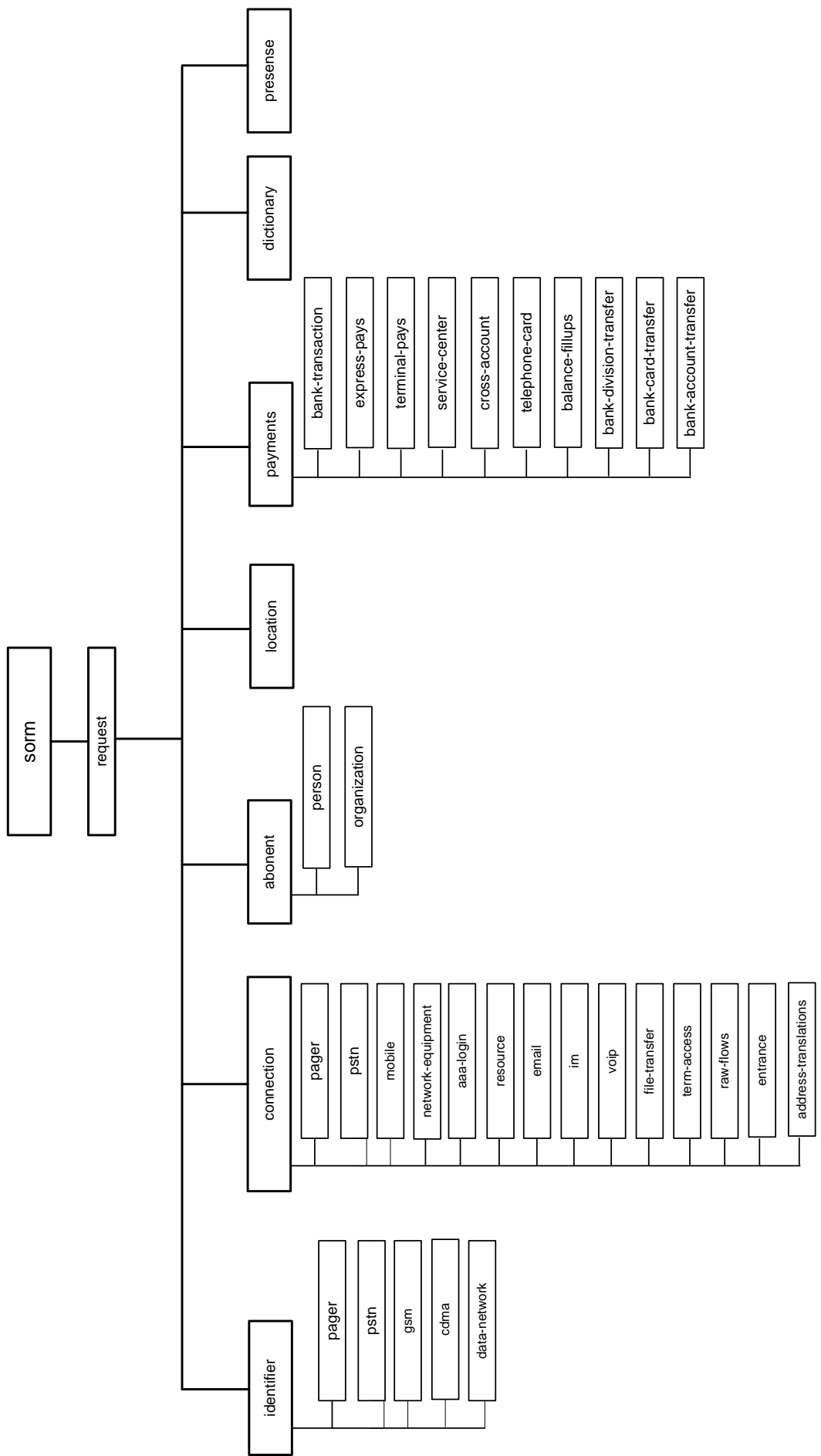


Рисунок 8. Структура разделения поисковых критериев кпд1.

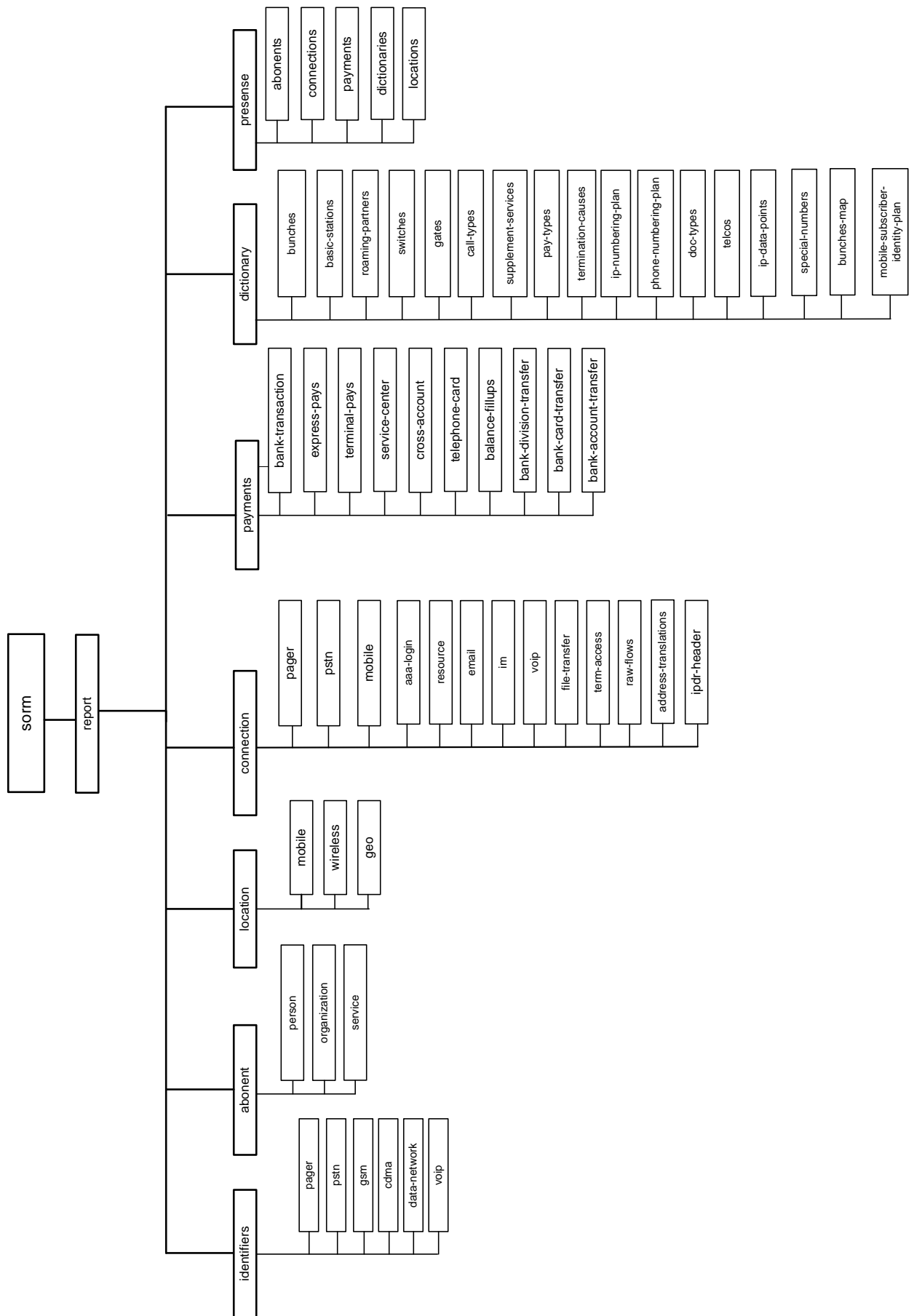


Рисунок 9. Структура разделения видов отчетов кпд2.

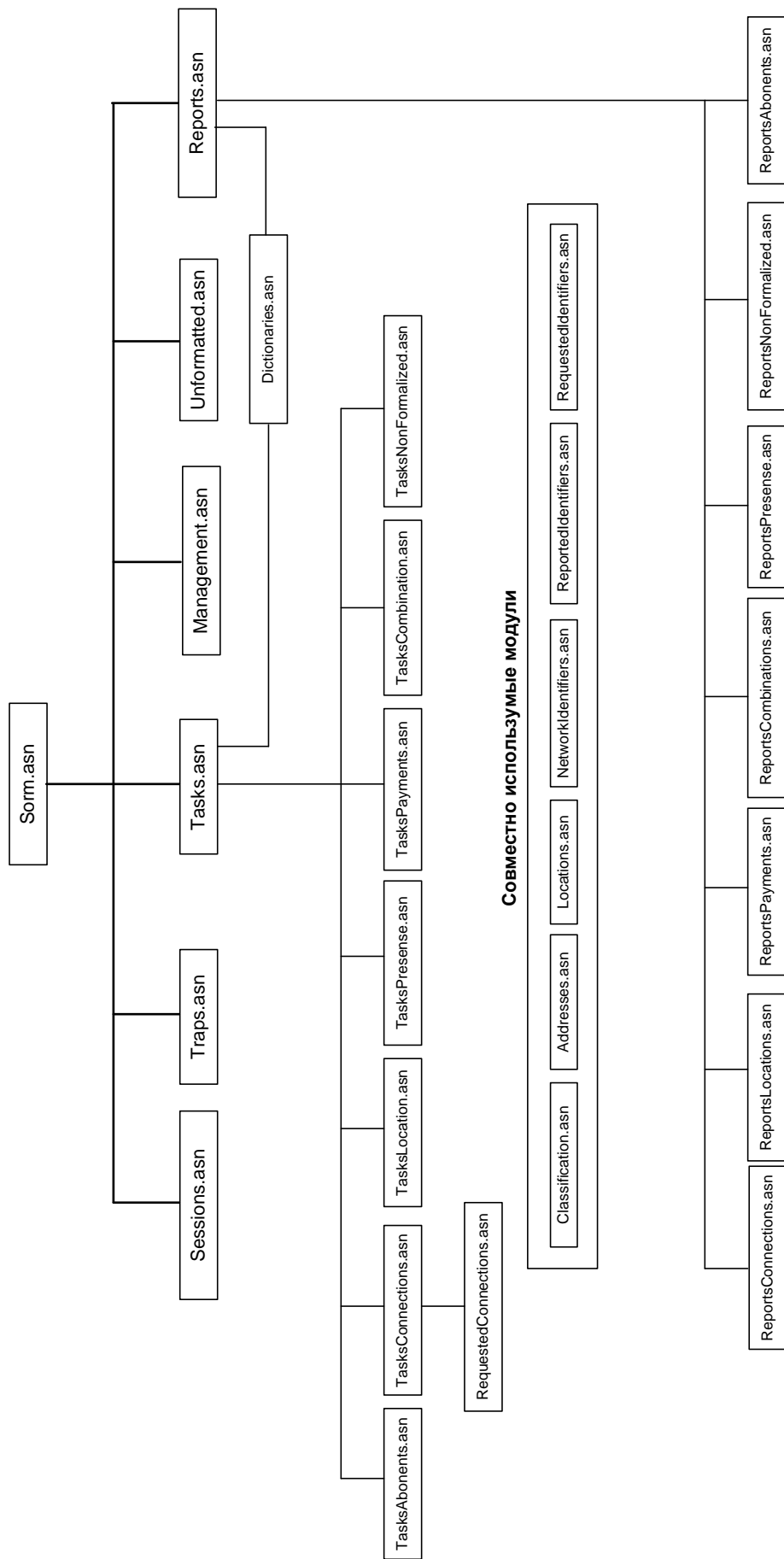


Рисунок 10. Структура ASN.1-модулей интерфейса взаимодействия ПУ и ИС СОРМ.

**ASN.1-спецификация протокола взаимодействия  
ПУ и ИС СОРМ**

```
1 Classification DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3 EXPORTS TAGGED,
4     sorm-message-session ,
5     sorm-message-trap ,
6     sorm-message-task ,
7     sorm-message-report ,
8     sorm-message-management ,
9     sorm-message-unformatted ,
10
11     sorm-request-identifier-pager ,
12     sorm-request-identifier-pstn ,
13     sorm-request-identifier-gsm ,
14     sorm-request-identifier-cdma ,
15     sorm-request-identifier-data-network ,
16
17     sorm-report-identifier-pager ,
18     sorm-report-identifier-pstn ,
19     sorm-report-identifier-gsm ,
20     sorm-report-identifier-cdma ,
21     sorm-report-identifier-data-network ,
22     sorm-report-identifier-voip ,
23
24     sorm-request-payment-bank-transaction ,
25     sorm-request-payment-express-pays ,
26     sorm-request-payment-terminal-pays ,
27     sorm-request-payment-service-center ,
28     sorm-request-payment-cross-account ,
29     sorm-request-payment-telephone-card ,
30     sorm-request-payment-balance-fillups ,
31     sorm-request-payment-bank-division-transfer ,
32     sorm-request-payment-bank-card-transfer ,
33     sorm-request-payment-bank-account-transfer ,
34
35
36     sorm-report-payment-bank-transaction ,
37     sorm-report-payment-express-pays ,
38     sorm-report-payment-terminal-pays ,
39     sorm-report-payment-service-center ,
40     sorm-report-payment-cross-account ,
41     sorm-report-payment-telephone-card ,
42     sorm-report-payment-balance-fillups ,
```

43 sorm-report-payment-bank-division-transfer ,  
44 sorm-report-payment-bank-card-transfer ,  
45 sorm-report-payment-bank-account-transfer ,  
46  
47  
48 sorm-request-connection-pager ,  
49 sorm-request-connection-pstn ,  
50 sorm-request-connection-mobile ,  
51 sorm-request-connection-aaa-login ,  
52 sorm-request-connection-resource ,  
53 sorm-request-connection-email ,  
54 sorm-request-connection-im ,  
55 sorm-request-connection-voip ,  
56 sorm-request-connection-file-transfer ,  
57 sorm-request-connection-term-access ,  
58 sorm-request-connection-raw-flows ,  
59 sorm-request-connection-entrance ,  
60 sorm-request-connection-address-translations ,  
61  
62 sorm-report-connection-pager ,  
63 sorm-report-connection-pstn ,  
64 sorm-report-connection-mobile ,  
65 sorm-report-connection-ipdr-header ,  
66 sorm-report-connection-aaa-login ,  
67 sorm-report-connection-resource ,  
68 sorm-report-connection-email ,  
69 sorm-report-connection-im ,  
70 sorm-report-connection-voip ,  
71 sorm-report-connection-file-transfer ,  
72 sorm-report-connection-term-access ,  
73 sorm-report-connection-raw-flows ,  
74 sorm-report-connection-address-translations ,  
75  
76 sorm-request-dictionaries ,  
77 sorm-report-dictionary-bunches ,  
78 sorm-report-dictionary-basic-stations ,  
79 sorm-report-dictionary-roaming-partners ,  
80 sorm-report-dictionary-switches ,  
81 sorm-report-dictionary-gates ,  
82 sorm-report-dictionary-call-types ,  
83 sorm-report-dictionary-supplement-services ,  
84 sorm-report-dictionary-pay-types ,  
85 sorm-report-dictionary-termination-causes ,  
86 sorm-report-dictionary-ip-numbering-plan ,

```

87 sorm-report-dictionary-phone-numbering-plan ,
88 sorm-report-dictionary-doc-types ,
89 sorm-report-dictionary-telcos ,
90 sorm-report-dictionary-ip-data-points ,
91 sorm-report-dictionary-special-numbers ,
92 sorm-report-dictionary-bunches-map ,
93
94
95 sorm-request-presense ,
96 sorm-report-presense-abonents ,
97 sorm-report-presense-connections ,
98 sorm-report-presense-payments ,
99 sorm-report-presense-dictionaries ,
100 sorm-report-presense-locations ,
101
102 sorm-request-abonent-person ,
103 sorm-request-abonent-organization ,
104
105 sorm-report-abonent-abonent ,
106 sorm-report-abonent-service ,
107 sorm-report-abonent-person ,
108 sorm-report-abonent-organization
109 ;
110
111
112 TAGGED ::= CLASS {
113     &id ObjectDescriptor UNIQUE,
114     &Data
115 }
116 WITH SYNTAX {
117     OID &id
118     DATA &Data
119 }
120
121 --- Классификация
122
123 OID ::= PrintableString
124
125 --- Подструктура сообщений
126 sorm-message-session OID ::= "280"
127 sorm-message-trap OID ::= "281"
128 sorm-message-task OID ::= "282"
129 sorm-message-report OID ::= "283"
130 sorm-message-management OID ::= "284"

```

```

131 sorm-message-unformatted OID ::= "285"
132
133
134 --- Идентификаторы
135 sorm-request-identifier-pager OID ::= "140"
136 sorm-request-identifier-pstn OID ::= "141"
137 sorm-request-identifier-gsm OID ::= "142"
138 sorm-request-identifier-cdma OID ::= "143"
139 sorm-request-identifier-data-network OID ::= "144"
140
141 sorm-report-identifier-pager OID ::= "1"
142 sorm-report-identifier-pstn OID ::= "2"
143 sorm-report-identifier-gsm OID ::= "3"
144 sorm-report-identifier-cdma OID ::= "4"
145 sorm-report-identifier-data-network OID ::= "5"
146 sorm-report-identifier-voip OID ::= "6"
147
148 --- Параметры соединений
149 sorm-request-connection-pager OID ::= "160"
150 sorm-request-connection-pstn OID ::= "161"
151 sorm-request-connection-mobile OID ::= "162"
152 sorm-request-connection-aaa-login OID ::= "164"
153 sorm-request-connection-resource OID ::= "165"
154 sorm-request-connection-email OID ::= "166"
155 sorm-request-connection-im OID ::= "167"
156 sorm-request-connection-voip OID ::= "168"
157 sorm-request-connection-file-transfer OID ::= "169"
158 sorm-request-connection-term-access OID ::= "170"
159 sorm-request-connection-raw-flows OID ::= "171"
160 sorm-request-connection-entrance OID ::= "172"
161 sorm-request-connection-address-translations OID ::= "173"
162
163 sorm-report-connection-pager OID ::= "20"
164 sorm-report-connection-pstn OID ::= "21"
165 sorm-report-connection-mobile OID ::= "22"
166 sorm-report-connection-ipdr-header OID ::= "23"
167 sorm-report-connection-aaa-login OID ::= "24"
168 sorm-report-connection-resource OID ::= "25"
169 sorm-report-connection-email OID ::= "26"
170 sorm-report-connection-im OID ::= "27"
171 sorm-report-connection-voip OID ::= "28"
172 sorm-report-connection-file-transfer OID ::= "29"
173 sorm-report-connection-term-access OID ::= "30"
174 sorm-report-connection-raw-flows OID ::= "31"

```



175 sorm-report-connection-address-translations OID ::= "32"  
176  
177 --- Абоненты  
178 sorm-request-abonent-person OID ::= "180"  
179 sorm-request-abonent-organization OID ::= "181"  
180  
181 sorm-report-abonent-abonent OID ::= "40"  
182 sorm-report-abonent-service OID ::= "41"  
183 sorm-report-abonent-person OID ::= "42"  
184 sorm-report-abonent-organization OID ::= "43"  
185  
186 --- Местоположение  
187 sorm-request-location OID ::= "200"  
188  
189 sorm-report-location-mobile OID ::= "60"  
190 sorm-report-location-wireless OID ::= "61"  
191 sorm-report-location-geo OID ::= "62"  
192  
193 --- Платежи  
194 sorm-request-payment-bank-transaction OID ::= "220"  
195 sorm-request-payment-express-pays OID ::= "221"  
196 sorm-request-payment-terminal-pays OID ::= "222"  
197 sorm-request-payment-service-center OID ::= "223"  
198 sorm-request-payment-cross-account OID ::= "224"  
199 sorm-request-payment-telephone-card OID ::= "225"  
200 sorm-request-payment-balance-fillups OID ::= "226"  
201 sorm-request-payment-bank-division-transfer OID ::= "227"  
202 sorm-request-payment-bank-card-transfer OID ::= "228"  
203 sorm-request-payment-bank-account-transfer OID ::= "229"  
204  
205  
206 sorm-report-payment-bank-transaction OID ::= "80"  
207 sorm-report-payment-express-pays OID ::= "81"  
208 sorm-report-payment-terminal-pays OID ::= "82"  
209 sorm-report-payment-service-center OID ::= "83"  
210 sorm-report-payment-cross-account OID ::= "84"  
211 sorm-report-payment-telephone-card OID ::= "85"  
212 sorm-report-payment-balance-fillups OID ::= "86"  
213 sorm-report-payment-bank-division-transfer OID ::= "87"  
214 sorm-report-payment-bank-card-transfer OID ::= "88"  
215 sorm-report-payment-bank-account-transfer OID ::= "89"  
216  
217 --- Справочники  
218

```

219 sorm-request-dictionaries OID ::= "240"
220
221 sorm-report-dictionary-bunches OID ::= "100"
222 sorm-report-dictionary-basic-stations OID ::= "101"
223 sorm-report-dictionary-roaming-partners OID ::= "102"
224 sorm-report-dictionary-switches OID ::= "103"
225 sorm-report-dictionary-gates OID ::= "104"
226 sorm-report-dictionary-call-types OID ::= "105"
227 sorm-report-dictionary-supplement-services OID ::= "106"
228 sorm-report-dictionary-pay-types OID ::= "107"
229 sorm-report-dictionary-termination-causes OID ::= "108"
230 sorm-report-dictionary-ip-numbering-plan OID ::= "109"
231 sorm-report-dictionary-phone-numbering-plan OID ::= "110"
232 sorm-report-dictionary-doc-types OID ::= "111"
233 sorm-report-dictionary-telcos OID ::= "112"
234 sorm-report-dictionary-ip-data-points OID ::= "113"
235 sorm-report-dictionary-special-numbers OID ::= "114"
236 sorm-report-dictionary-bunches-map OID ::= "115"
237 sorm-report-dictionary-mobile-subscriber-identity-plan OID ::= "116"
238
239 --- Запрос о наличии данных
240 sorm-request-presense OID ::= "260"
241
242 sorm-report-presense-abonents OID ::= "120"
243 sorm-report-presense-connections OID ::= "121"
244 sorm-report-presense-payments OID ::= "122"
245 sorm-report-presense-dictionaries OID ::= "123"
246 sorm-report-presense-locations OID ::= "124"
247
248
249 END

```

---

## Addresses.asn

```

1  Addresses DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS AddressType ,
5      ReportedAddresses ,
6      ReportedAddress ,
7      RequestedAddress ;
8
9
10 AddressType ::= ENUMERATED {

```

```

11 registered (0),
    физических лиц)
12 postal (1),
13 invoice (2),
    юридических лиц)
14 device-location (3),
    дополнительный адрес для абонентов
15 reserved (4)
16 }
17 }
18
19 ReportedAddresses ::= SEQUENCE OF ReportedAddress
20
21 ReportedAddress ::= SEQUENCE {
22   title      AddressType,
23   address-info AddressInfoReport
24 }
25
26 AddressInfoReport ::= CHOICE {
27   struct-info [1] AddressStructInfoReport,
28   unstruct-info [2] UTF8String (SIZE (1 .. 1024))
29 }
30
31 AddressStructInfoReport ::= SEQUENCE {
32   zip [0] UTF8String (SIZE (1 .. 32)) OPTIONAL,
33   country [1] UTF8String (SIZE (1 .. 128)) OPTIONAL,
34   region [2] UTF8String (SIZE (1 .. 128)) OPTIONAL,
35   zone [3] UTF8String (SIZE (1 .. 128)) OPTIONAL,
36   city [4] UTF8String (SIZE (1 .. 128)) OPTIONAL,
37   street [5] UTF8String (SIZE (1 .. 128)) OPTIONAL,
38   building [6] UTF8String (SIZE (1 .. 128)) OPTIONAL,
39   build-sect [7] UTF8String (SIZE (1 .. 128)) OPTIONAL,
40   apartment [8] UTF8String (SIZE (1 .. 128)) OPTIONAL
41 }
42
43
44
45 -- поля адресных данных
46 RequestedAddress ::= SEQUENCE {
47   zip [0] UTF8String (SIZE (1 .. 32)) OPTIONAL,
48   country [1] UTF8String (SIZE (1 .. 128)) OPTIONAL,
49   region [2] UTF8String (SIZE (1 .. 128)) OPTIONAL,
50   zone [3] UTF8String (SIZE (1 .. 128)) OPTIONAL,

```

— Адрес регистрации (обязателен для юридических и физических лиц)  
 — Почтовый адрес (дополнительный адрес для юридических лиц)  
 — Адрес доставки счета (дополнительный адрес для юридических лиц)  
 — Адрес установки устройства (телефонов) (дополнительный адрес для абонентов)  
 — телефонов общего пользования)  
 — Резерв  
 — тип адреса  
 — адрес  
 — структурированный адрес  
 — неструктурированный адрес  
 — почтовый индекс, zip-код  
 — страна  
 — область  
 — район, муниципальный округ  
 — город, поселок, деревня  
 — улица  
 — дом, строение  
 — корпус  
 — квартира, офис  
 — почтовый индекс, zip-код  
 — страна  
 — область  
 — район, муниципальный округ

```

51 city [4] UTF8String (SIZE (1 .. 128)) OPTIONAL,
52 street [5] UTF8String (SIZE (1 .. 128)) OPTIONAL,
53 building [6] UTF8String (SIZE (1 .. 128)) OPTIONAL,
54 build-sect [7] UTF8String (SIZE (1 .. 128)) OPTIONAL,
55 apartment [8] UTF8String (SIZE (1 .. 128)) OPTIONAL
56 }
57
58 END

```

— город, поселок, деревня, населенный пункт  
 — улица  
 — дом, строение  
 — корпус  
 — квартира, офис

## Dictionaries.asn

```

1 Dictionaries DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS
5   TelcoID,
6   TelcoList,
7
8   DictionaryTask,
9   DictionaryReport,
10
11   PhoneAbonentType;
12
13 IMPORTS DateAndTime
14 FROM Sorm
15 TAGGED,
16 sorm-request-dictionary,
17 sorm-report-dictionary-telcos,
18 sorm-report-dictionary-bunches,
19 sorm-report-dictionary-basic-stations,
20 sorm-report-dictionary-roaming-partners,
21 sorm-report-dictionary-switches,
22 sorm-report-dictionary-gates,
23 sorm-report-dictionary-call-types,
24 sorm-report-dictionary-supplement-services,
25 sorm-report-dictionary-pay-types,
26 sorm-report-dictionary-termination-causes,
27 sorm-report-dictionary-ip-numbering-plan,
28 sorm-report-dictionary-phone-numbering-plan,
29 sorm-report-dictionary-doc-types,
30 sorm-report-dictionary-ip-data-points,
31 sorm-report-dictionary-special-numbers,
32 sorm-report-dictionary-bunches-map,
33 sorm-report-dictionary-mobile-subscriber-identity-plan

```

```

34 FROM Classification
35
36 ReportedAddress
37 FROM Addresses
38
39 Bunch,
40 DataNetworkEquipment,
41 IPAddress,
42 IPPort,
43 IPMask,
44 NetworkPeerInfo,
45 NetworkType
46 FROM NetworkIdentifiers
47
48 GeoLocation FROM Locations;
49
50 -- Идентификатор оператора связи или филиала
51 TelcoID ::= INTEGER (0 .. 65535)
52
53 --- список идентификаторов операторов связи или филиалов
54 TelcoList ::= SEQUENCE OF TelcoID
55
56 -- Запрос
57 DictionaryTask ::= SEQUENCE {
58   id TAGGED.&id ({DictionaryTaskVariants}),
59   data TAGGED.&Data ({DictionaryTaskVariants}){@id}}
60 }
61
62 DictionaryTaskVariants TAGGED ::= { dictionaryTask }
63
64 dictionaryTask TAGGED ::= {
65   OID { sorm-request-dictionaries }
66   DATA ObjectDescriptor
67 }
68
69 -- ObjectDescriptor принимает значение одно из:
70 -- sorm-report-dictionary-telcos
71 -- sorm-report-dictionary-bunches
72 -- sorm-report-dictionary-basic-stations
73 -- sorm-report-dictionary-roaming-partners
74 -- sorm-report-dictionary-switches
75 -- sorm-report-dictionary-gates
76 -- sorm-report-dictionary-call-types
77 -- sorm-report-dictionary-supplement-services

```

— тип запрашиваемого справочника (идентификатор отчета)

```

78 -- sorm-report-dictionary-pay-types
79 -- sorm-report-dictionary-termination-causes
80 -- sorm-report-dictionary-ip-numbering-plan
81 -- sorm-report-dictionary-phone-numbering-plan
82 -- sorm-report-dictionary-doc-types
83 -- sorm-report-dictionary-ip-data-points
84 -- sorm-report-dictionary-special-numbers
85 -- sorm-report-dictionary-bunches-map,
86 -- sorm-report-dictionary-mobile-subscriber-identity-plan
87
88
89 -- Отчет
90 DictionaryReport ::= SEQUENCE {
91   id          TAGGED.&id ( { DictionaryRecordsVariants } ),
92   data       TAGGED.&Data( { DictionaryRecordsVariants } { @id } )
93 }
94
95 DictionaryRecordsVariants TAGGED ::= {
96   telcosRecords
97 | bunchesRecords
98 | basicStationsSectorRecords
99 | roamingPartnersRecords
100 | switchesRecords
101 | gatesRecords
102 | callTypesRecords
103 | supplementServicesRecords
104 | payTypesRecords
105 | terminationCausesRecords
106 | ipNumberingPlanRecords
107 | telephoneNumberingPlanRecords
108 | docTypesRecords
109 | ipDataPointsRecords
    данных, с которых получены записи о соединениях
110 | specialNumbersRecords
    сервисы)
111 | bunchesMapRecords
112 | mobileSubscriberIdentityPlanRecords
    абонентов
113 }
114
115 -- операторы связи, обслуживаемые ИС СОФМ
116 telcosRecords TAGGED ::= {
117   OID { sorm-report-dictionary-telcos }
118   DATA SEQUENCE OF TelcosRecord }

```

— идентификтор записи справочника  
— данные записи справочника

— операторы связи, обслуживаемые ИС СОФМ  
— пучки соединительных линий  
— базовые станции  
— роуминговые партнеры  
— коммутаторы  
— IP шлюзы  
— типы вызовов  
— список ДЮ  
— способы оплаты (пополнения баланса)  
— причины завершения соединения  
— IP-план адресации  
— план телефонной номерной емкости  
— типы документов, удостоверяющих личность  
— идентификаторы точек подключения к сети передачи  
— специальные номера оператора (SMS-центры, TMC,  
— карта связей пучков соединительных линий  
— план нумерации идентификаторов мобильных телефонных

```

119 TelcosRecord ::= SEQUENCE {
120   telco-id          TelcoID,
121   begin-time       DateAndTime,
122   end-time         DateAndTime OPTIONAL,
123   description     UTF8String (SIZE (1 .. 256)),
124   msc [0]         NumericString (SIZE(3)) OPTIONAL,
125   nmc [1]         NumericString (SIZE(3)) OPTIONAL
126 }
127
128
129 — пучки соединительных линий
130 bunchesRecords TAGGED ::= {
131   OID { sorm-report-dictionary-bunches }
132   DATA SEQUENCE OF BunchRecord
133 }
134
135 BunchRecord ::= SEQUENCE {
136   telco-id          TelcoID,
137   bunch-id         Bunch,
138   switch-id        UTF8String (SIZE (1 .. 128)),
139   bunch-type       ENUMERATED {
140     inbound (0),
141     outbound (1),
142     bidirectional (3)
143   },
144   begin-time       DateAndTime,
145   end-time         DateAndTime OPTIONAL,
146   description     UTF8String (SIZE(1 .. 256))
147 }
148
149
150 — базовые станции
151 basicStationsSectorRecords TAGGED ::= {
152   OID { sorm-report-dictionary-basic-stations }
153   DATA SEQUENCE OF BasicStationSectorRecord
154 }
155
156 BasicStationSectorRecord ::= SEQUENCE {
157   telco-id          TelcoID,
158   begin-time       DateAndTime,
159   end-time         DateAndTime,
160   address          UTF8String (SIZE (1 .. 256)),
161   расположение

```

— номер филиала или оператора связи  
 — время начала действия  
 — время конца действия  
 — описание (наименование) оператора связи или филиала  
 — код страны  
 — код оператора связи

— идентификатор оператора связи или филиала  
 — идентификатор пучка  
 — идентификатор коммутатора  
 — тип – входящий/исходящий

— время начала назначения пучка  
 — время конца назначения пучка  
 — расшифровка пучка

— идентификатор оператора связи или филиала  
 — время начала действия базовой станции  
 — время конца действия базовой станции  
 — произвольное текстовое описание адреса или места

```

162 sector-identifiers      BasicStationIdentifiers ,
163 antenna-configuration  BasicStationAntenna ,
164 station-type           BasicStationType ,
165
166 structured-address [0] ReportedAddress OPTIONAL,
167 location [1]          GeoLocation OPTIONAL
168 }
169
170 — идентификаторы сектора
171 BasicStationIdentifiers ::= CHOICE {
172 telephone [0] TelephoneIdentifiers ,
173 wireless [1] SEQUENCE OF WirelessIdentifiers
174 }
175
176 — идентификаторы сектора для телефонной сети
177 TelephoneIdentifiers ::= SEQUENCE {
178 lac INTEGER (0 .. 65535) ,
179 cell INTEGER (0 .. 4294967295) ,
180   базовой станции и сектор)
181   cell-sign UTF8String (SIZE (1 .. 18)) OPTIONAL
182 }
183
184 — идентификаторы сектора для сети передачи данных
185 WirelessIdentifiers ::= SEQUENCE {
186 cell UTF8String (SIZE (1 .. 64)) ,
187 ip-list IPList OPTIONAL,
188 mac OCTET STRING (SIZE (6)) OPTIONAL
189 }
190 IPList ::= SEQUENCE OF NetworkPeerInfo
191
192 — параметры антенной системы
193 BasicStationAntenna ::= CHOICE {
194 gsm-antenna [0] GsmAntenna ,
195 cdma-antenna [1] SEQUENCE OF CdmaAntenna ,
196 wireless-antenna [2] SEQUENCE OF WirelessAntenna
197 }
198
199 — параметры антенной системы GSM-сектора
200 GsmAntenna ::= SEQUENCE {
201 azimuth INTEGER (-1 .. 359) ,
202   если -1, то нет направленности
203 width INTEGER (0 .. 359) ,
204 horizon-angle INTEGER (0 .. 359) ,

```

— идентификаторы сектора  
— параметры антенной системы  
— тип сети базовой станции  
— адрес места установки базовой станции  
— географическое местоположение  
— идентификаторы сектора для телефонной сети  
— идентификаторы сектора для сети передачи данных  
— код зоны  
— идентификатор сектора базовой станции (идентификатор телефонной идентификатор соты)  
— идентификатор сектора для сети передачи данных  
— перечень назначенных сектору IP-адресов/портов  
— MAC-адрес сетевого оборудования сектора  
— параметры антенной системы GSM-сектора  
— параметры антенной системы CDMA-сектора  
— параметры антенной системы WiFi/WiMAX-сектора  
— азимут относительно направления на север, в градусах,  
— ширина раstra в градусах  
— угол наклона сектора к горизонту



```

204 power [0] INTEGER (0 .. 25000) OPTIONAL,
205 frequency [1] INTEGER (0 .. 10000000000) OPTIONAL,
206 vertical-lift [2] INTEGER (0 .. 100) OPTIONAL,
207 gain-factor [3] INTEGER (-100 .. 100) OPTIONAL,
208 polarization [4] INTEGER (-45 .. 45) OPTIONAL
209 }
210
211 CdmaAntenna ::= BroadbandWirelessParameters
212 WirelessAntenna ::= BroadbandWirelessParameters
213
214 BroadbandWirelessParameters ::= SEQUENCE {
215 azimuth INTEGER (-1 .. 359),
    если -1, то нет направленности
216 width INTEGER (0 .. 359),
217 horizon-angle INTEGER (0 .. 359),
218 power [0] INTEGER (0 .. 25000) OPTIONAL,
219 frequency-start [1] INTEGER (0 .. 10000000000) OPTIONAL,
220 frequency-stop [2] INTEGER (0 .. 10000000000) OPTIONAL,
221 leaf-level [3] INTEGER (-45 .. 45) OPTIONAL,
222 vertical-lift [4] INTEGER (0 .. 100) OPTIONAL,
223 gain-factor [5] INTEGER (-100 .. 100) OPTIONAL,
224 polarization [6] INTEGER (-45 .. 45) OPTIONAL
225 }
226
227 — виды базовых станций
228 BasicStationType ::= ENUMERATED {
229 gsm (0),
230 cdma (1),
231 umts (2),
232 wifi (3),
233 wimax (4)
234 }
235
236
237 roamingPartnersRecords TAGGED::= {
238 OID { sorm-report-dictionary-roaming-partners }
239 DATA SEQUENCE OF RoamingPartnerRecord
240 }
241
242 RoamingPartnerRecord ::= SEQUENCE {
243 telco-id TelcoID,
244 roaming-id INTEGER (0 .. 65535),
245 begin-time DateAndTime,
246 end-time DateAndTime OPTIONAL,

```

— мощность в ваттах (сектор)  
— частота излучения (сектор)  
— высота подвеса сектора  
— коэффициент усиления антенны (Дб)  
— поляризация антенной системы

— параметры антенной системы CDMA-сектора  
— параметры антенной системы WiFi/WMAX-сектора

— азимут относительно направления на север, в градусах,  
— ширина раstra в градусах  
— угол наклона сектора к горизонту  
— мощность в ваттах (сектор)  
— нижняя частота излучения диапазона (сектор)  
— верхняя частота излучения диапазона (сектор)  
— уровень боковых лепестков  
— высота подвеса сектора  
— коэффициент усиления антенны (Дб)  
— поляризация антенной системы

— идентификатор оператора связи или филиала  
— идентификатор роумингового партнра  
— время начала действия роуминга  
— время конца действия роуминга

```

247     description UTF8String (SIZE (1 .. 256))
248 }
249
250 switchesRecords TAGGED ::= {
251   OID { sorm-report-dictionary-switches }
252   DATA SEQUENCE OF SwitchesRecord
253 }
254
255 SwitchesRecord ::= SEQUENCE {
256   telco-id TelcoID,
257   switch-id UTF8String (SIZE (1 .. 128)),
258   begin-time DateAndTime,
259   end-time DateAndTime OPTIONAL,
260   description UTF8String (SIZE (1 .. 256)),
261   network-type NetworkType,
262   address ReportedAddress,
263   switch-sign NumericString (SIZE (1 .. 18)) OPTIONAL,
264   switch-type ENUMERATED {
265     internal(0), --- внутренний
266     border(1) --- пограничный
267 }
268 }
269
270 gatesRecords TAGGED ::= {
271   OID { sorm-report-dictionary-gates }
272   DATA SEQUENCE OF GatesRecord
273 }
274
275 GatesRecord ::= SEQUENCE {
276   telco-id TelcoID,
277   gate-id INTEGER (0 .. 65535),
278   ip-list IPList,
279   begin-time DateAndTime,
280   end-time DateAndTime OPTIONAL,
281   description UTF8String (SIZE (1 .. 256)),
282   address ReportedAddress,
283   gate-type ENUMERATED {
284     sgsn(0),
285     ggsn(1),
286     smsc(2),
287     gmsc(3),
288     hss(4),
289     pstn(5),
290     voip-gw(6),

```

--- описание

--- идентификатор оператора связи или филиала  
 --- идентификатор коммутатора  
 --- время начала действия коммутатора  
 --- время конца действия коммутатора  
 --- описание  
 --- тип сети связи  
 --- адрес места установки коммутатора  
 --- телефонный идентификатор коммутатора

--- идентификатор оператора связи или филиала  
 --- идентификатор шлюза  
 --- IP адрес шлюза  
 --- время начала действия шлюза  
 --- время конца действия шлюза  
 --- описание  
 --- адрес места установки шлюза  
 --- тип IP-шлюза

```

291         aaa (7)
292     }
293 }
294
295 callTypesRecords TAGGED ::= {
296     OID { sorm-report-dictionary-call-types }
297     DATA SEQUENCE OF CallsTypesRecord
298 }
299
300 CallsTypesRecord ::= SEQUENCE {
301     telco-id      TelcoID,
302     call-type-id  INTEGER (0 .. 65535),
303     begin-time   DateAndTime,
304     end-time     DateAndTime OPTIONAL,
305     description  UTF8String (SIZE (1 .. 256))
306 }
307
308
309 supplementServicesRecords TAGGED ::= {
310     OID { sorm-report-dictionary-supplement-services }
311     DATA SEQUENCE OF SupplementServicesRecord
312 }
313
314 SupplementServicesRecord ::= SEQUENCE {
315     telco-id      TelcoID,
316     service-id    INTEGER (1 .. 65535),
317     mnemonic      UTF8String (SIZE (1..10)),
318     begin-time   DateAndTime,
319     end-time     DateAndTime OPTIONAL,
320     description  UTF8String (SIZE (1 .. 256))
321 }
322
323
324 payTypesRecords TAGGED ::= {
325     OID { sorm-report-dictionary-pay-types }
326     DATA SEQUENCE OF PayTypesRecord
327 }
328
329 PayTypesRecord ::= SEQUENCE {
330     telco-id      TelcoID,
331     pay-type-id   INTEGER (0 .. 65535),
332     begin-time   DateAndTime,
333     end-time     DateAndTime OPTIONAL,
334     description  UTF8String (SIZE (1 .. 256))

```

— идентификатор оператора связи или филиала  
 — идентификатор типа вызова  
 — время начала действия  
 — время конца действия  
 — описание

— идентификатор оператора связи или филиала  
 — идентификатор сервиса  
 — мнемоническое обозначение  
 — время начала действия  
 — время конца действия  
 — описание

— идентификатор оператора связи или филиала  
 — идентификатор типа оплаты  
 — время начала действия  
 — время конца действия  
 — описание

```

335 }
336
337 terminationCausesRecords TAGGED ::= {
338   OID { sorm-report-dictionary-termination-causes }
339   DATA SEQUENCE OF TerminationCausesRecord
340 }
341
342 TerminationCausesRecord ::= SEQUENCE {
343   telco-id TelcoID,
344   termination-cause-id INTEGER (0 .. 16384),
345   begin-time DateAndTime,
346   end-time DateAndTime OPTIONAL,
347   description UTF8String (SIZE (1 .. 256)),
348   network-type NetworkType
349 }
350
351
352 ipNumberingPlanRecords TAGGED ::= {
353   OID { sorm-report-dictionary-ip-numbering-plan }
354   DATA SEQUENCE OF IpNumberingPlanRecord }
355
356 IpNumberingPlanRecord ::= SEQUENCE {
357   telco-id TelcoID,
358   description UTF8String (SIZE (1 .. 256)),
359   network-address IPAddress,
360   network-mask IPMask,
361   begin-time DateAndTime,
362   end-time DateAndTime OPTIONAL
363 }
364
365 telephoneNumberingPlanRecords TAGGED ::= {
366   OID { sorm-report-dictionary-phone-numbering-plan }
367   DATA SEQUENCE OF TelephoneNumberingPlanRecord
368 }
369
370 TelephoneNumberingPlanRecord ::= SEQUENCE {
371   telco-id TelcoID,
372   iso-3166-alpha-2 UTF8String (SIZE (2)),
373   iso-3166-alpha-3 UTF8String (SIZE (3)),
374   country-code UTF8String (SIZE (3)),
375   national-significant-number UTF8String (SIZE (14)),
376   код зоны
377   area-code-length INTEGER (0 .. 6),
378   min-subscr-nr-length INTEGER (1 .. 15),

```

```

--- идентификатор оператора связи или филиала
--- код причины
--- время начала действия
--- время конца действия
--- описание
--- тип сети связи

```

```

--- идентификатор оператора связи или филиала
--- описание назначения диапазона
--- подсеть
--- маска подсети
--- время начала действия
--- время конца действия

```

```

--- идентификатор оператора связи или филиала
--- 2-х символьная аббревиатура страны
--- 3-х символьная аббревиатура страны
--- международный код страны
--- номерной телефонный префикс оператора связи, включая
--- длина кода зоны в телефонном префиксе оператора связи
--- минимальная длина абонентского номера, символов (

```

```

378     national-significant-number + min-subscr)
max-subscr-nr-length INTEGER (1 .. 15),
national-significant-number + max-subscr)
379     utc-min INTEGER (-12 .. 12),
380     utc-max INTEGER (-12 .. 12),
381     destination UTF8String (SIZE (2 .. 255)),
382     operator-type-id NetworkType,
383
384     capacity-from NumericString (SIZE (1 .. 15)),
385     capacity-to NumericString (SIZE (1 .. 15)),
386     capacity-size INTEGER (1 .. 100000),
387
388     location UTF8String (SIZE (0 .. 255)),
389     registrar UTF8String (SIZE (0 .. 255)),
390     range-activation DateAndTime,
391     mobile-country-code NumericString (SIZE(3)),
392     mobile-network-code NumericString (SIZE(3)),
393
394     range-deactivation [0] DateAndTime OPTIONAL,
395     range-status [1] UTF8String (SIZE (2 .. 128)) OPTIONAL,
396     description [2] UTF8String (SIZE (2 .. 255)) OPTIONAL,
emкости
397     operating-company-number [3] UTF8String (SIZE (0 .. 4)) OPTIONAL
398 }
399
400 docTypesRecords TAGGED ::= {
401     OID {sorm-report-dictionary-doc-types}
402     DATA SEQUENCE OF DocTypesRecord
403 }
404
405 DocTypesRecord ::= SEQUENCE {
406     telco-id TelcoID,
407     doc-type-id INTEGER (0 .. 65535),
408     begin-time DateAndTime,
409     end-time DateAndTime OPTIONAL,
410     description UTF8String (SIZE (1 .. 256))
411 }
412
413
414
415 ipDataPointsRecords TAGGED ::= {
416     OID { sorm-report-dictionary-ip-data-points }
417     DATA SEQUENCE OF IpDataPointRecord
418 }

```

— максимальная длина абонентского номера, символов (
   
 — минимальный часовой пояс
   
 — максимальный часовой пояс
   
 — страна
   
 — тип сети связи оператора
   
 — нижняя граница диапазона выданных номеров (от)
   
 — верхняя граница диапазона выданных номеров (до)
   
 — количество выделенных номеров в диапазоне (емкость)
   
 — текстовое описание местоположения оператора связи
   
 — наименование оператора связи
   
 — дата и время начала действия номерной емкости
   
 — MCC
   
 — MNC
   
 — дата и время завершения действия номерной емкости
   
 — текущее состояние номерной емкости
   
 — расшифровка оказываемых услуг связи по номерной
   
 — международный идентификатор оператора связи
   
 — идентификатор оператора связи или филиала
   
 — идентификатор типа документа
   
 — время начала действия
   
 — время конца действия
   
 — описание (наименование)

```

419
420 IpDataPointRecord ::= SEQUENCE {
421     telco-id      TelcoID,
422     point-id     INTEGER (0 .. 1000),
423     description  UTF8String (SIZE (1 .. 256)),
424     begin-time   DateAndTime,
425     end-time     DateAndTime OPTIONAL
426 }
427
428 specialNumbersRecords TAGGED ::= {
429     OID { sorm-report-dictionary-special-numbers }
430     DATA SEQUENCE OF SpecialNumberRecord
431 }
432
433 SpecialNumberRecord ::= SEQUENCE {
434     telco-id      TelcoID,
435     directory-number UTF8String (SIZE (2 .. 32)),
436     description   UTF8String (SIZE (1 .. 256)),
437     номера
438     begin-time    DateAndTime,
439     end-time      DateAndTime OPTIONAL,
440     network-address IPAddress OPTIONAL
441 }
442
443 bunchesMapRecords TAGGED ::= {
444     OID { sorm-report-dictionary-bunches-map }
445     DATA SEQUENCE OF BunchesMapRecord
446 }
447
448 BunchesMapRecord ::= SEQUENCE {
449     a-bunch      BunchMapPoint,
450     b-bunch      BunchMapPoint,
451     begin-time   DateAndTime,
452     end-time     DateAndTime OPTIONAL
453 }
454
455 BunchMapPoint ::= SEQUENCE {
456     telco-id     TelcoID,
457     switch-id    UTF8String (SIZE (1 .. 128)),
458     bunch-id     Bunch
459 }
460
461

```

— идентификатор оператора связи или филиала  
 — идентификатор точки подключения  
 — описание (наименование) точки подключения  
 — время начала действия  
 — время конца действия

— идентификатор оператора связи или филиала  
 — специальный номер  
 — описание (наименование, назначение) специального  
 — время начала действия  
 — время конца действия  
 — адрес в сети передачи данных

— пара коммутатор/пучок в связи  
 — пара коммутатор/пучок в связи  
 — время начала действия связи  
 — время конца действия связи

— идентификатор оператора связи или филиала  
 — идентификатор коммутатора  
 — идентификатор пучка коммутатора

```

462 mobileSubscriberIdentityPlanRecords TAGGED ::= {
463   OID { sorm-report-dictionary-mobile-subscriber-identity-plan }
464   DATA SEQUENCE OF MobileSubscriberIdentityPlanRecord
465 }
466
467 MobileSubscriberIdentityPlanRecord ::= SEQUENCE {
468   telco-id
469     NumericString (SIZE(3)),
470     NumericString (SIZE(3)),
471     NumericString (SIZE(3 .. 10)),
472     capacity-from
473     NumericString (SIZE(0 .. 7)),
474     capacity-to
475     NumericString (SIZE(0 .. 7)),
476     description
477     INTEGER (1 .. 1000000),
478     region
479     UTF8String (SIZE(2 .. 255)),
480     city
481     UTF8String (SIZE(1 .. 128)),
482     range-activation
483     DateAndTime,
484     range-deactivation [0] DateAndTime OPTIONAL,
485     range-status [1]
486     UTF8String (SIZE(2 .. 128)) OPTIONAL,
487 }
488
489 -- Тип абонента
490 PhoneAbonentType ::= ENUMERATED {
491   local (0),
492   network (1),
493   roamer (2)
494 }
495
496 END

```

## Locations.asn

```

1 Locations DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS
5   Location,
6   GeoLocation;
7
8 Location ::= CHOICE {
9   mobile-location [0] MobileLocation,
10  wireless-location [1] WirelessLocation,

```

--- местоположение мобильного абонента  
 --- местоположение абонента мобильной сети передачи данных

```

11   geo-location [2]      GeoLocation
12 }
13
14 MobileLocation ::= SEQUENCE {
15   lac      INTEGER (0 .. 65535),
16   cell     INTEGER (0 .. 4294967295),
17   ta [0]   INTEGER (0 .. 63) OPTIONAL
18 }
19
20 WirelessLocation ::= SEQUENCE {
21   cell     UTF8String (SIZE (1 .. 64)),
22   mac      OCTET STRING (SIZE (6))
23 }
24
25 GeoLocation ::= SEQUENCE {
26   latitude-grade REAL,
27   longitude-grade REAL,
28   projection-type ENUMERATED {
29     wgs84 (0),
30     utm (1),
31     sgs85 (2)
32   }
33 }
34
35 END

```

--- географическое местоположение  
 --- код зоны  
 --- идентификатор базовой станции  
 --- Timing Advance (временная компенсация)  
 --- идентификатор сектора  
 --- MAC-адрес сетевого оборудования сектора  
 --- ширина  
 --- долгота  
 --- тип проекции координат

---

## Management.asn

```

1 Management DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS managementMessage;
5
6 IMPORTS TAGGED,
7   sorm-message-management
8   FROM Classification;
9
10 managementMessage TAGGED ::= {
11   OID { sorm-message-management }
12   DATA CHOICE {
13     request [0] ManagementRequest,
14     response [1] ManagementResponse
15   }
16 }

```



```

17
18
19 --- тип сообщения "команда управления ИС СОФМ"
20 ManagementRequest ::= CHOICE {
21   get-structure [0] GetStructureRequest,
22   get-module-config [1] GetModuleConfigRequest,
23   set-module-config [2] SetModuleConfigRequest,
24   check-module [3] CheckModuleRequest,
25   get-module-types [4] GetModuleTypesRequest
26 }
27
28
29
30 --- запрос на получение структуры ИС СОФМ – КТС и модулей СПО
31 GetStructureRequest ::= NULL
32
33 --- запрос на получение конфигурации КТС/модуля СПО
34 GetModuleConfigRequest ::= CHOICE {
35   hw-modules-list [0] RequestedHardwareModules,
36   sw-modules-list [1] RequestedSoftwareModules
37 }
38
39
40 RequestedHardwareModules ::= SEQUENCE OF ModuleId
41 RequestedSoftwareModules ::= SEQUENCE OF ModuleId
42
43
44
45 --- запрос на изменение конфигурации КТС/модуля СПО
46 SetModuleConfigRequest ::= SEQUENCE {
47   module-id ModuleId,
48   module-config ConfiguratedModule
49 }
50
51 ConfiguratedModule ::= CHOICE {
52   sw-module [0] SormSoftwareModule,
53   hw-module [1] SormHardwareModule
54 }
55
56
57 --- запрос на получение состояния модуля
58 CheckModuleRequest ::= RequestedModulesList
59
60 RequestedModulesList ::= CHOICE {

```

--- запрос на получение структуры ИС СОФМ – КТС и модулей СПО  
--- запрос на получение конфигурации КТС/модуля СПО  
--- запрос на изменение конфигурации КТС/модуля СПО  
--- запрос на получение состояния модуля  
--- запрос на получение типов модулей КТС и СПО

--- перечень идентификаторов узлов КТС ИС СОФМ  
--- перечень идентификаторов модулей СПО ИС СОФМ

--- перечень идентификаторов узлов КТС ИС СОФМ  
--- перечень идентификаторов модулей СПО ИС СОФМ

--- идентификатор конфигурируемого модуля  
--- устанавливаемая в модуль конфигурация

--- для узла КТС  
--- для узла СПО

```

61 sw-modules [0] RequestedHardwareModules,
62 hw-modules [1] RequestedSoftwareModules
63     состояние
64 }
65
66 — запрос на получение типов модулей КТС и СПО
67 GetModuleTypesRequest ::= NULL
68
69
70 — уникальный идентификатор КТС/модуля СПО ИС СОФМ
71 ModuleId ::= OBJECT STRING (SIZE (8))
72
73 — параметр модуля
74 ModuleParameter ::= SEQUENCE {
75     parameter-name UTF8String (SIZE (1 .. 256)),
76     read-only BOOLEAN,
77     parameter-value ParameterValue
78 }
79
80 — варианты значений параметров
81 ParameterValue ::= CHOICE {
82     string [0] UTF8String (SIZE (1 .. 256)),
83     integer [1] INTEGER (0 .. 999999999),
84     boolean [2] BOOLEAN
85 }
86
87 ModuleParameters ::= SEQUENCE OF ModuleParameter
88
89 SormSoftwareModule ::= SEQUENCE {
90     module-id ModuleId,
91     hardware-module-id ModuleId,
92     block-name INTEGER (0 .. 1024),
93     module-name UTF8String (SIZE (1 .. 512)),
94     module-type INTEGER (1 .. 512),
95     module-parameters ModuleParameters,
96     sub-modules-list SubmodulesList OPTIONAL
97 }
98
99 SubmodulesList ::= SEQUENCE OF SormSoftwareModule
100
101
102 SormHardwareModule ::= SEQUENCE {
103     module-id ModuleId,

```

--- идентификаторы узлов КТС, для которых запрашивается состояние  
 --- идентификаторы модулей ИС СОФМ, для которых запрашивается состояние  
 --- уникальный идентификатор данного модуля  
 --- идентификатор КТС, на котором работает данный блок модуля СПО  
 --- номер блока СПО модуля  
 --- наименование модуля  
 --- идентификатор типа модуля  
 --- список параметров модуля  
 --- субмодули  
 --- уникальный идентификатор данного модуля

```

104 block-name          INTEGER (0 .. 1024),
105 module-name         UTF8String (SIZE (1 .. 512)),
106 module-parameters  HwParameterGroups
107 }
108
109 HwParameterGroup ::= SEQUENCE {
110   group-name        UTF8String (SIZE (1 .. 512)),
111   module-parameters ModuleParameters
112 }
113
114 HwParameterGroups ::= SEQUENCE OF HwParameterGroup
115
116 SormSoftwareModules ::= SEQUENCE OF SormSoftwareModule
117
118 SormHardwareModules ::= SEQUENCE OF SormHardwareModule
119
120
121
122
123 --- тип сообщения "ответ на команду управления ИС СОФМ"
124 ManagementResponse ::= CHOICE {
125   get-structure [0]  GetStructureResponse,
126   get-module-config [1] GetModuleConfigResponse,
127   set-module-config [2] SetModuleConfigResponse,
128   check-module [3]  CheckModuleResponse,
129   get-module-types [4] GetModuleTypesResponse
130 }
131
132
133 --- ответ на запрос получения структуры ИС СОФМ – КТС и модулей СЮ
134 GetStructureResponse ::= SEQUENCE {
135   sw-modules SormHardwareModules,
136   hw-modules SormSoftwareModules
137 }
138
139
140 --- ответ на запрос получения конфигурации КТС/модуля СЮ
141 GetModuleConfigResponse ::= SEQUENCE {
142   sw-modules SormHardwareModules,
143   hw-modules SormSoftwareModules
144 }
145
146

```

--- номер блока КТС  
--- наименование модуля  
--- значение группы параметров КТС  
--- наименование группы параметров для КТС  
--- перечень параметров для КТС  
--- тип сообщения "ответ на команду управления ИС СОФМ"  
--- ответ на запрос получения структуры ИС СОФМ – КТС и модулей СЮ  
--- ответ на запрос получения конфигурации КТС/модуля СЮ  
--- ответ на запрос изменения конфигурации КТС/модуля СЮ  
--- ответ на запрос получения состояния модуля  
--- ответ на запрос получения типов модулей КТС и СЮ  
--- перечень всех узлов КТС  
--- перечень всех модулей СЮ ИС СОФМ  
--- конфигурации запрошенных узлов КТС  
--- конфигурации запрошенных модулей СЮ ИС СОФМ

```

147 --- отчет на запрос изменения конфигурации KTC/ модуля СПО
148 SetModuleConfigResponse ::= ConfiguredModule
149
150
151 --- установленная в модуль конфигурация
152
153 --- ответ на запрос получения состояния модуля
154 CheckModuleResponse ::= CHOICE {
155   hw-modules [0] SormHardwareModules,
156   sw-modules [1] SormSoftwareModules
157 }
158
159 --- ответ на запрос получения типов модулей KTC и СПО
160 GetModuleTypesResponse ::= SEQUENCE OF ModuleType
161
162 ModuleType ::= SEQUENCE {
163   module-type INTEGER (1 .. 512),
164   type-description UTF8String ( SIZE (1 .. 128))
165 }
166
167 --- идентификатор типа модуля
168 --- расшифровка типа модуля

```

## NetworkIdentifiers.asn

```

1 NetworkIdentifiers DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS Bunch, DataNetworkEquipment, IPAddress, IPPort, IPMask, NetworkPeerInfo, NetworkType, DataVoipNumber, IMProtocol,
   VoipProtocol;
5
6 --- идентификаторы пучка
7 Bunch ::= CHOICE {
8   gsm [0] INTEGER (0 .. 4294967295),
9   cdma-umts [1] DataNetworkEquipment
10 }
11
12 --- идентификатор оборудования сети передачи данных
13 DataNetworkEquipment ::= CHOICE {
14   mac [0] OCTET STRING (SIZE (6)),
15   atm [1] DataNetworkATM
16 }
17
18 --- ATM адрес (SDH/PDH сети)
19 DataNetworkATM ::= SEQUENCE {
20   vpi OCTET STRING (SIZE (1)),
21   vci OCTET STRING (SIZE (2)) OPTIONAL

```

```

22 }
23
24
25 --- тип сети связи (стандарт)
26 NetworkType ::= ENUMERATED {
27   not-specified (0),
28   mob-gsm(1),
29   mob-cdma(2),
30   fix-pstn(3),
31   data-ip(4),
32   data-srv(5),
33   data-ip-mob(6),
34   data-ip-wifi(7),
35   data-ip-max(8),
36   paging(9),
37   voip(10)
38   данных
39 }
40 --- IP-адрес
41 IPAddress ::= CHOICE {
42   ipv4 [0] IPV4Address,
43   ipv6 [1] IPV6Address
44 }
45
46 --- IPv4-адрес
47 IPV4Address ::= OCTET STRING (SIZE (4))
48
49
50 --- IPv6-адрес
51 IPV6Address ::= OCTET STRING (SIZE (16))
52
53
54 --- IP/UDP/TCP-порт
55 IPPort ::= OCTET STRING (SIZE (2)) --- порт
56
57
58 --- маска IP-подсети
59 IPMask ::= CHOICE {
60   ipv4-mask [0] IPV4Mask,
61   ipv6-mask [1] IPV6Mask
62 }
63
64

```

```

--- неконкретизированный стандарт
--- сеть мобильной связи стандарта GSM
--- сеть мобильной связи стандарта CDMA
--- TФoП-сеть
--- стационарные сети передачи данных
--- TМC-службы
--- мобильная сеть передачи данных
--- беспроводная сеть передачи данных стандарта WiFi
--- беспроводная сеть передачи данных стандарта WFMАX
--- персональный радиовывоз
--- сеть передачи голосовой информации посредством сети передачи

```

```

--- IPv4-адрес
--- IPv6-адрес

--- IP адрес (4 байта)

--- IP адрес (16 байт)

--- IPv4-маска
--- IPv6-маска

```

```

65 --- информация об участнике соединения передачи данных
66 NetworkPeerInfo ::= SEQUENCE {
67   ip-address IPAddress,
68   ip-port IPPort OPTIONAL
69 }
70
71 --- IPv4-маска
72 IPV4Mask ::= OCTET STRING (SIZE (4))
73
74 --- IPv6-маска
75 IPV6Mask ::= OCTET STRING (SIZE (16))
76
77
78 -- Номер телефона VoIP-абонента
79 DataVoipNumber ::= SEQUENCE {
80   original-number UTF8String (SIZE (1 .. 128)),
81   translated-number [0] UTF8String (SIZE (1 .. 32)) OPTIONAL,
82   e164-number [1] UTF8String (SIZE (1 .. 15)) OPTIONAL
83 }
84
85 VoipProtocol ::= ENUMERATED {
86   sip(0),
87   h323(1),
88   iax(2),
89   skype(100)
90 }
91 IMPProtocol ::= ENUMERATED {
92   icq(0),
93   aol(1),
94   msn(2),
95   yahoo(3),
96   web-mail(4),
97   skype(5),
98   irc(6),
99   jabber(7),
100  mra(8),
101  tencent(9),
102  mms(98),
103  sms(99),
104  vk(100),
105  facebook(101),
106  myspace(102),
107  twitter(103)
108 }

```

109  
110

**END**

## ReportedIdentifiers.asn

```
1  ReportedIdentifiers DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS
5    ReportedIdentifier
6  ;
7
8  IMPORTS TAGGED,
9    sorm-report-identifier -pager ,
10   sorm-report-identifier -pstn ,
11   sorm-report-identifier -gsm ,
12   sorm-report-identifier -cdma ,
13   sorm-report-identifier -data-network ,
14   sorm-report-identifier -voip
15  FROM Classification
16
17  IPAddress, DataNetworkEquipment
18  FROM NetworkIdentifiers;
19
20
21  ReportedIdentifier ::= SEQUENCE {
22    id TAGGED.&id ( {ReportedIdentifierVariants} ),
23    data TAGGED.&Data ( {ReportedIdentifierVariants} ){@id}
24  }
25
26  — варианты идентификаторов отччда
27  ReportedIdentifierVariants TAGGED ::= {
28    reportedPagerIdentifier
29    | reportedPstnIdentifier
30    | reportedGsmIdentifier
31    | reportedCdmaIdentifier
32    | reportedDataNetworkIdentifier
33    | reportedVoipIdentifier
34  }
35
36  — идентификатор сети персонального радиовызова
37  reportedPagerIdentifier TAGGED ::= {
38    OID { sorm-report-identifier -pager }
39    DATA ReportedPagerIdentifier
```

— идентификатор сети персонального радиовызова  
— идентификатор ТФОП  
— идентификатор GSM  
— идентификатор CDMA  
— идентификатор сети передачи данных

```

40 }
41
42 ReportedPagerIdentifier ::= NumericString (SIZE (2 .. 18))
43
44 — идентификатор телефонной сети общего пользования
45 reportedPstnIdentifier TAGGED ::= {
46   OID { sorm-report-identifier -pstn }
47   DATA ReportedPstnIdentifier
48 }
49
50 ReportedPstnIdentifier ::= SEQUENCE {
51   directory-number UTF8String (SIZE (2 .. 32)),
52   internal-number NumericString (SIZE (1 .. 32)) OPTIONAL
53 }
54
55
56 — идентификатор абонента GSM
57 reportedGsmIdentifier TAGGED ::= {
58   OID { sorm-report-identifier -gsm }
59   DATA ReportedGsmIdentifier
60 }
61
62 ReportedGsmIdentifier ::= SEQUENCE {
63   directory-number UTF8String (SIZE (2 .. 32)),
64   imsi NumericString (SIZE (10 .. 18)),
65   imei [0] NumericString (SIZE (10 .. 18)) OPTIONAL,
66   icc [1] NumericString (SIZE (19 .. 20)) OPTIONAL
67 }
68
69 — идентификатор абонента CDMA
70 reportedCdmaIdentifier TAGGED ::= {
71   OID { sorm-report-identifier -cdma }
72   DATA ReportedCdmaIdentifier
73 }
74
75 ReportedCdmaIdentifier ::= SEQUENCE {
76   directory-number UTF8String (SIZE (2 .. 32)),
77   imsi NumericString (SIZE (10 .. 18)),
78   esn [0] NumericString (SIZE (10 .. 18)) OPTIONAL,
79   min [1] NumericString (SIZE (10 .. 18)) OPTIONAL,
80   icc [2] NumericString (SIZE (19 .. 20)) OPTIONAL
81 }
82
83

```

— телефонный номер в международном формате  
— Дополнительный внутренний номер, если есть

— идентификатор абонента GSM  
— идентификатор мобильного абонента  
— идентификатор мобильной станции  
— идентификатор мобильного абонента (CDMA)  
— идентификатор SIM-карты абонента

— идентификатор абонента CDMA  
— идентификатор мобильного абонента  
— идентификатор мобильной станции  
— идентификатор мобильного абонента (CDMA)  
— идентификатор SIM-карты абонента



```

84 reportedDataNetworkIdentifier TAGGED ::= {
85   OID { sorm-report-identifier-data-network }
86   DATA ReportedDataNetworkIdentifier
87 }
88
89 ReportedDataNetworkIdentifier ::= SEQUENCE {
90   user-equipment [0] DataNetworkEquipment OPTIONAL,
91   login [1] UTF8String (SIZE (1 .. 128)) OPTIONAL,
92   ip-address [2] IPAddress OPTIONAL,
93   e-mail [3] UTF8String (SIZE (1 .. 128)) OPTIONAL,
94   pin [4] NumericString (SIZE (2 .. 20)) OPTIONAL,
95   phone-number [5] UTF8String (SIZE (2 .. 32)) OPTIONAL,
96   user-domain [6] UTF8String (SIZE (2 .. 128)) OPTIONAL,
97   reserved [7] UTF8String (SIZE (1 .. 128)) OPTIONAL
98 }
99
100 reportedVoipIdentifier TAGGED ::= {
101   OID { sorm-report-identifier-voip }
102   DATA ReportedVoipIdentifier
103 }
104
105 ReportedVoipIdentifier ::= SEQUENCE {
106   ip-address IPAddress,
107   originator-name UTF8String (SIZE (1 .. 32)),
108   calling-number UTF8String (SIZE (1 .. 32))
109 }
110
111
112 END

```

```

--- идентификатор пользователя оборудования
--- имя пользователя — login
--- IP адрес
--- адрес электронной почты
--- PIN
--- номер телефона
--- пользовательский домен
--- резерв

```

```

--- IP-адрес абонента
--- общедоступное имя инициатора связи
--- номер вызывающего абонента

```

## ReportsAbonents.asn

```

1 ReportsAbonents DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS AbonentsReport;
5
6 IMPORTS TAGGED,
7   sorm-report-abonent-abonent,
8   sorm-report-abonent-service,
9   sorm-report-abonent-person,
10  sorm-report-abonent-organization
11 FROM Classification
12

```

```

13      DateAndTime
14      FROM Sorm
15
16      TelcoID
17      FROM Dictionaries
18
19      ReportedIdentifier
20      FROM ReportedIdentifiers
21
22      NetworkType
23      FROM NetworkIdentifiers
24
25      AddressType,
26      ReportedAddresses,
27      ReportedAddress
28      FROM Addresses
29
30      Location
31      FROM Locations;
32
33      AbonentsReport ::= SEQUENCE {
34      id TAGGED.&id ({AbonentsReportVariants}),
35      data TAGGED.&Data({AbonentsReportVariants}@id)}
36      }
37
38      AbonentsReportVariants TAGGED ::= {
39      reportAbonent |
40      reportService
41      }
42
43      reportAbonent TAGGED ::= {
44      OID sorm-report-abonent-abonent
45      DATA SEQUENCE OF AbonentsRecord
46      }
47
48      AbonentsRecord ::= SEQUENCE {
49      telco-id TelcoID,
50      idents ReportedIdentifier,
51      contract-date DateAndTime,
52      contract UTF8String (SIZE (1 .. 64)),
53      actual-from DateAndTime,
54      actual-to DateAndTime,
55      актуальна информация
56      актуальна информация
57
58      --- идентификатор оператора связи или филиала
59      --- идентификаторы абонента
60      --- дата и время заключения договора
61      --- номер договора
62      --- дата и время начала интервала времени, на котором
63      --- дата и время окончания интервала времени, на котором
64
65      --- информация об абоненте
66      --- информация об услугах
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

55 abonent AbonentInfo,
56 status ActiveStatus,
57 attach [0] DateAndTime OPTIONAL,
подключение производилось на интервале актуальности)
58 detach [1] DateAndTime OPTIONAL,
отключение производилось на интервале актуальности)
59 last-location [2] Location OPTIONAL,
мобильных абонентов
60 services [3] ActiveServices OPTIONAL,
61 line-data [4] LineData OPTIONAL,
62 standard [5] Standard OPTIONAL,
63 addresses [6] ReportedAddresses OPTIONAL
64 }
65
66 ActiveServices ::= SEQUENCE OF AbonentService
67
68
69 ActiveStatus ::= ENUMERATED {
70 active (0),
71 not-active (1)
72 }
73
74 LineData ::= SEQUENCE {
75 object [0] UTF8String (SIZE (1 .. 128)) OPTIONAL,
76 cross [1] UTF8String (SIZE (1 .. 128)) OPTIONAL,
77 block [2] UTF8String (SIZE (1 .. 128)) OPTIONAL,
78 pair [3] UTF8String (SIZE (1 .. 128)) OPTIONAL,
79 reserved [4] UTF8String (SIZE (1 .. 128)) OPTIONAL
80 }
81
82
83 AbonentInfo ::= SEQUENCE {
84 id TAGGED.&id ({AbonentsInfoVariants}),
85 data TAGGED.&Data({AbonentsInfoVariants}){@id}
86 }
87
88 AbonentsInfoVariants TAGGED ::= {
89 abonentPerson |
90 abonentOrganization
91 }
92
93 — Физическое лицо
94 abonentPerson TAGGED ::= {
95 OID { sorm-report-abonent-person }

```

```

— информация об абоненте (клиенте оператора связи)
— текущий статус абонента (подключен/отключен)
— дата и время подключения основной услуги (если
— дата и время отключения основной услуги (если
— последнее зафиксированное местоположение для
— активированные услуги
— линейные данные (кросс, рамка, пара и т.д.)
— стандарт связи
— адреса абонента
— набор подключенных ДБО
— активный
— не активный
— описание объекта связи
— описание кросса
— описание блока
— описание пары
— резерв
— информация об абоненте — физическом лице
— информация об абоненте — юридическом лице

```

```

96  DATA AbonentPerson
97  }
98
99  AbonentPerson ::= SEQUENCE {
100  name-info      PersonNameInfoReport,
101  birth-date     GeneralizedTime OPTIONAL,
102  passport-info  PassportInfoReport,
103  bank [1]       UTF8String (SIZE(1 .. 256)) OPTIONAL,
104               связи)
105  bank-account [2] UTF8String (SIZE(1 .. 30)) OPTIONAL
106               оператором связи)
107 }
108 PersonNameInfoReport ::= CHOICE {
109  struct-name [0] PersonStructNameInfoReport,
110  unstruct-name [1] UTF8String (SIZE(1 .. 1024))
111 }
112 PersonStructNameInfoReport ::= SEQUENCE {
113  given-name   UTF8String (SIZE (1 .. 128)),
114  initial      UTF8String (SIZE (1 .. 128)),
115  family-name  UTF8String (SIZE (1 .. 128))
116 }
117
118 PassportInfoReport ::= SEQUENCE {
119  ident-card-info IdentCardInfoReport,
120  doc-type-id  INTEGER (0 .. 65535)
121               личность)
122 }
123 IdentCardInfoReport ::= CHOICE {
124  struct-info [0] IdentCardStructInfoReport,
125  unstruct-info [1] UTF8String (SIZE (1 .. 512))
126 }
127 IdentCardStructInfoReport ::= SEQUENCE {
128  ident-card-serial UTF8String (SIZE (1..16)),
129  ident-card-number NumericString (SIZE (1..16)),
130  ident-card-description UTF8String (SIZE (1 .. 256))
131 }
132
133
134
135 abonentOrganization TAGGED ::= {
136  OID { sorm-report-abonent-organization }

```

— ФИО

— дата рождения

— паспортные данные

— банк абонента (используемый при расчетах с оператором

связи)

— счет абонента в банке (используемый при расчетах с

оператором связи)

— структурированное ФИО

— неструктурированное ФИО

— имя

— отчество

— фамилия

— описание удостоверения личности

— идентификатор типа документа, удостоверяющего

— структурированная информация

— неструктурированная информация

— серия удостоверения личности

— номер удостоверения личности

— когда и кем выдано

```

137 DATA AbonentOrganization
138 }
139
140 AbonentOrganization ::= SEQUENCE {
141   full-name      UTF8String (SIZE (1 .. 128)),
142   inn            UTF8String (SIZE (1 .. 64)),
143   contact [0]   UTF8String (SIZE (1 .. 128)) OPTIONAL,
144   phone-fax [1] UTF8String (SIZE (1 .. 128)) OPTIONAL,
145   internal-users [2] InternalUsers OPTIONAL,
146
147
148
149   bank [3]      UTF8String (SIZE (1 .. 256)) OPTIONAL,
150
151   bank-account [4] UTF8String (SIZE (1 .. 30)) OPTIONAL
152   оператором связи)
153 }
154 InternalUsers ::= SEQUENCE OF InternalUsersRecord
155 абонента Ц
156
157 InternalUsersRecord ::= SEQUENCE {
158   user-name      UTF8String (SIZE (1 .. 64)),
159   internal-number NumericString (SIZE (1 .. 8))
160 }
161
162 Standard ::= NetworkType ---- перечень стандартов связи
163
164
165 reportService TAGGED ::= {
166   OID sorm-report-abonent-service
167   DATA SEQUENCE OF AbonentService
168 }
169
170 AbonentService ::= SEQUENCE {
171   telco-id      TelcoID,
172   service-id    INTEGER (0 .. 65535),
173   idsents       ReportedIdentifier OPTIONAL,
174   contract      UTF8String (SIZE (1 .. 64)),
175   begin-time    DateAndTime,

```

— полное наименование  
 — ИНН  
 — контактное лицо  
 — контактные телефоны, факс  
 — список внутренних пользователей, выдается только при  
 — заполнении аналогичного поля в запросе формирования  
 — задачи на поиск идентификаторов абонентов или при  
 — заполнении  
 — поля internal-number в запросе на принадлежность  
 — идентификаторов  
 — операторам связи  
 — банк абонента (используемый при расчетах с оператором  
 — счет абонента в банке (используемый при расчетах с  
 — набор записей, описывающих внутренних пользователей  
 — юридического лица  
 — строка – описатель внутреннего пользователя  
 — внутренний номер  
 — идентификатор оператора связи или филиала  
 — идентификатор услуги  
 — идентификаторы абонента  
 — номер договора  
 — дата и время начала оказания услуги

176 end-time DateAndTime, — дата и время окончания оказания услуги  
 177 parameter UTF8String (SIZE(1..256)) OPTIONAL — индивидуальные параметры настройки услуги абонента  
 178 } — номер на который осуществляется преадресация  
 179 }  
 180  
 181 **END**

## Reports.asn

```

1 Reports DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS reportMessage,
5 Acknowledgement;
6
7 IMPORTS TAGGED,
8 sorm-message-report
9 FROM Classification
10
11 Message, MessageID, DateAndTime
12 FROM Sorm
13
14 TaskID
15 FROM Tasks
16
17 DictionaryReport
18 FROM Dictionaries
19
20 ConnectionsReport
21 FROM ReportsConnections
22
23 LocationReport
24 FROM ReportsLocations
25
26 PaymentsReport
27 FROM ReportsPayments
28
29 CombinationReport
30 FROM ReportsCombinations
31
32 PresenseReport
33 FROM ReportsPresense
34
35 NonFormalizedReport

```

```

36 FROM ReportsNonFormalized
37
38 AbonentsReport
39 FROM ReportsAbonents;
40
41 reportMessage TAGGED ::= {
42   OID { sorm-message-report }
43   DATA CHOICE {
44     report [0] Report,
45     ack [1] Acknowledgement
46   }
47 }
48
49 -- Блок данных сообщения типа "отчет"
50 Report ::= SEQUENCE {
51   request-id MessageID,
52   task-id TaskID,
53   total-blocks-number INTEGER (1 .. 1000000000000),
54   block-number INTEGER (1 .. 1000000000000),
55   report-block ReportDataBlock
56 }
57
58 -- Подтверждение причма блока, передачда с номером сообщения соответствующему номеру сообщения блока отчда
59 Acknowledgement ::= SEQUENCE {
60   successful BOOLEAN,
61   broken-record INTEGER (10 .. 100000) OPTIONAL,
62   error-description UTF8String (SIZE(1..1024)) OPTIONAL
63 }
64
65 ReportDataBlock ::= CHOICE {
66   dictionary [0] DictionaryReport,
67   abonents [1] AbonentsReport,
68   connections [2] ConnectionsReport,
69   locations [3] LocationReport,
70   payments [4] PaymentsReport,
71   combination [5] CombinationReport,
72   presense [6] PresenseReport,
73   nonFormalized [7] NonFormalizedReport
74 }
75
76
77

```

--- тип сообщения "отчет"  
--- тип сообщения "подтверждение"  
--- идентификатор запроса, запрошивший отчет  
--- идентификатор задачи, сгенерировавшей данный отчет  
--- общее количество блоков в отчете  
--- порядковый номер текущего блока  
--- блок данных отчета  
--- признак успешного причма блока  
--- номер записи в отчете, обработанной на ПУ с ошибкой  
--- описание ошибки причма блока в произвольной форме  
--- отчеты задач пополнения справочников (нормативно-  
--- отчеты задач поисков по принадлежности абонентов  
--- отчеты задач поисков по соединениям абонентов  
--- отчет задачи получения данных местоположения  
--- отчеты задач поисков по совершенным платежам  
--- отчеты задач поисков по усечению отобранных данных  
--- отчеты задач по запросу наличия в ИС СОФМ информации  
--- отчеты задач по обработке неформализованных данных

78  
79

**END**

---

### ReportsCombinations.asn

```
1 ReportsCombinations DEFINITIONS IMPLICIT TAGS ::=  
2 BEGIN  
3  
4 EXPORTS CombinationReport;  
5  
6 IMPORTS CallsRecords  
7 FROM ReportsConnections;  
8  
9 — отчеты задач поисков по усечению отобранных данных  
10 CombinationReport ::= CallsRecords  
11  
12 END
```

---

### ReportsConnections.asn

```
1 ReportsConnections DEFINITIONS IMPLICIT TAGS ::=  
2 BEGIN  
3  
4 EXPORTS ConnectionsReport, CallsRecords;  
5  
6 IMPORTS DateAndTime  
7 FROM Sorm  
8  
9 Location  
10 FROM Locations  
11  
12 ReportedIdentifier  
13 FROM ReportedIdentifiers  
14  
15 Bunch,  
16 NetworkPeerInfo,  
17 DataNetworkEquipment,  
18 DataVoipNumber,  
19 IPAddress,  
20 VoipProtocol,  
21 IMProtocol  
22 FROM NetworkIdentifiers  
23  
24 PhoneAbonentType,
```



```

25 TelcoID
26 FROM Dictionaries
27
28 TAGGED,
29     sorm-report-connection-pager,
30     sorm-report-connection-pstn,
31     sorm-report-connection-mobile,
32     sorm-report-connection-ipdr-header,
33     sorm-report-connection-aaa-login,
34     sorm-report-connection-resource,
35     sorm-report-connection-email,
36     sorm-report-connection-im,
37     sorm-report-connection-voip,
38     sorm-report-connection-file-transfer,
39     sorm-report-connection-term-access,
40     sorm-report-connection-raw-flows,
41     sorm-report-connection-address-translations
42 FROM Classification;
43
44
45 ConnectionsReport ::= CallsRecords
46
47
48 CallsRecords ::= SEQUENCE {
49     id TAGGED.&id ({ReportedCallsVariants}),
50     data TAGGED.&Data ({ReportedCallsVariants}@id)}
51 }
52
53 ReportedCallsVariants TAGGED ::= {
54     pagerRecord
55     | pstnRecord
56     | mobileRecord
57     | dataAAARecord
58     | dataEmailRecord
59     | dataImRecord
60     | dataFileTransferRecord
61     | dataTermAccessRecord
62     | dataRawFlowsRecord
63     | dataResourceRecord
64     | dataVoipRecord
65     | dataAddressTranslationRecord
66 }
67
68

```

```

69 -- Детализированные записи отправки пейджинг-сообщений
70 pagerRecord TAGGED ::= {
71   OID { sorm-report-connection-pager }
72   DATA SEQUENCE OF PagerRecordContent
73 }
74
75 PagerRecordContent ::= SEQUENCE {
76   telco-id          TelcoID,
77   call-type-id     INTEGER (0 .. 65535),
78   connection-time  DateAndTime,
79   info             ReportedIdentifier,
80   in-bytes-count   INTEGER (0 .. 1024),
81   term-cause       INTEGER (0 .. 16384)
82 }
83
84 -- Детализированные записи звонков абонентов ТФОП, в т.ч. и неудавшиеся попытки соединений
85 pstnRecord TAGGED ::= {
86   OID { sorm-report-connection-pstn }
87   DATA SEQUENCE OF PstnRecordContent
88 }
89
90 PstnRecordContent ::= SEQUENCE {
91   telco-id          TelcoID,
92   begin-connection-time  DateAndTime,
93   duration          INTEGER (0 .. 86399),
94   call-type-id     INTEGER (0 .. 65535),
95   supplement-service-id  INTEGER (0 .. 65535),
96   in-abonent-type  PhoneAbonentType,
97   out-abonent-type PhoneAbonentType,
98   switch-id       UTF8String (SIZE (1 .. 128)),
99   inbound-bunch   INTEGER (0 .. 4294967295),
100  outbound-bunch  INTEGER (0 .. 4294967295),
101  term-cause       INTEGER (0 .. 16384),
102
103  phone-card-number [0]   NumericString (SIZE (1.. 20)) OPTIONAL,
104  in-info [1]             ReportedIdentifier OPTIONAL,
105  dialed-digits [2]       UTF8String (SIZE (2 .. 32)),
106  out-info [3]           ReportedIdentifier OPTIONAL,
107  forwarding-identifier [4] UTF8String (SIZE (2 .. 32)) OPTIONAL,
108  border-switch-id [5]   UTF8String (SIZE (1 .. 128)) OPTIONAL
109 }
110
111 -- Детализированные записи звонков мобильных абонентов, в т.ч. и неудавшиеся попытки соединений
112 -- Должны содержать также записи об SMS, в т.ч. и неудавшиеся попытки отправки

```

113 mobileRecord TAGGED ::= {  
114   OID { sorm-report-connection-mobile }  
115   DATA SEQUENCE OF MobileRecordContent  
116 }  
117  
118 MobileRecordContent ::= SEQUENCE {  
119   telco-id                    TelcoID,  
120   begin-connection-time     DateAndTime,  
121   duration                    INTEGER (0 .. 86399),  
122   call-type-id                INTEGER (0 .. 65535),  
123   supplement-service-id      INTEGER (0 .. 65535),  
124   in-abonent-type            PhoneAbonentType,  
125   out-abonent-type            PhoneAbonentType,  
126   switch-id                   UTF8String (SIZE (1 .. 128)),  
127   term-cause                  INTEGER (0 .. 16384),  
128  
129   inbound-bunch [0]  
130   outbound-bunch [1]  
131   in-info [2]  
132   in-end-location [3]  
133   in-begin-location [4]  
134   out-info [5]  
135   out-begin-location [6]  
136   out-end-location [7]  
137   forwarding-identifier [8] UTF8String (SIZE (2 .. 32)) OPTIONAL,  
138   roaming-partner-id [9]     INTEGER (0 .. 65535) OPTIONAL,  
139   border-switch-id [10]      UTF8String (SIZE (1 .. 128)) OPTIONAL  
140 }  
141  
142 — запись IPDR подключения/отключения абонента к сети связи  
143 dataAAARecord TAGGED ::= {  
144   OID { sorm-report-connection-aaa-login }  
145   DATA SEQUENCE OF DataAAARecordContent  
146 }  
147  
148 DataAAARecordContent ::= SEQUENCE {  
149   telco-id                    TelcoID,  
150   point-id                    INTEGER (0 .. 1000),  
151   aaa-connection-time        DateAndTime,

— идентификатор оператора связи или филиала  
— дата и время начала соединения  
— время соединения  
— тип соединения  
— ДБО при соединении  
— тип вызываемого абонента  
— код коммутатора обслуживающего соединения  
— причина завершения соединения  
— входящий пучок  
— исходящий пучок  
— идентификаторы вызываемого абонента  
— местоположение вызываемого абонента на конец  
— местоположение вызываемого абонента на начало  
— идентификаторы вызываемого абонента  
— местоположение вызываемого абонента на начало  
— местоположение вызываемого абонента на конец  
— телефонный номер при переадресации  
— код роумингового партнера  
— код пограничного коммутатора  
— идентификатор оператора связи или филиала  
— идентификатор точки подключения к сети  
— дата и время подключения/отключения к сети

```

152   передачи данных
153   aaa-login-type ENUMERATED {
154     connect (0),
155     disconnect (1)
156   },
157   aaa-session-id
158   aaa-allocated-ip
159   aaa-user-name
160   aaa-connect-type
161     номер порта для TCP/UDP
162   aaa-calling-number
163   aaa-called-number
164   aaa-has
165   aaa-in-bytes-count
166   aaa-out-bytes-count
167   aaa-user-password [0]
168   aaa-user-equipment [1]
169   aaa-apn [2]
170   )
171   IPAddress OPTIONAL,
172   IPAddress OPTIONAL,
173   IPAddress OPTIONAL,
174   INTEGER (0 .. 65535) OPTIONAL,
175   Location OPTIONAL,
176   соединение (передача данных посредством подвижной сети связи)
177   Location OPTIONAL,
178   Location OPTIONAL,
179   соединение (передача данных посредством подвижной сети связи)
180   NumericString (SIZE (20)) OPTIONAL,
181   NumericString (SIZE (10 .. 18)) OPTIONAL,
182   NumericString (SIZE (10 .. 18)) OPTIONAL,
183   NumericString (SIZE (10 .. 18)) OPTIONAL,
184   NumericString (SIZE (10 .. 18)) OPTIONAL,
185   NumericString (SIZE (2 .. 18)) OPTIONAL
186 }
187
188   запись IPDR передачи почтового сообщения
189   dataEmailRecord TAGGED ::= {
190     OID { sorm-report-connection-email }
191     DATA SEQUENCE OF DataEmailRecordContent
192   }
193
194   DataEmailRecordContent ::= CHOICE {
195     mail-aaa [0] DataEmailRecordContentAAA,
196     mail-ipdr [1] DataEmailRecordContentIPDR
197   }

```

— тип соединения  
— подключение к сети передачи данных  
— отключение от сети передачи данных

— идентификатор сессии  
— выделенный динамический IP-адрес  
— имя пользователя (логин)  
— код протокола в соответствии с RFC1700 либо

— вызывающий номер  
— вызываемый номер  
— IP-адрес/порт NAS-сервера  
— объем принятых данных  
— объем переданных данных  
— пользовательский пароль  
— идентификатор пользовательского оборудования  
— наименование точки доступа (Access Point Name)

— IP-адрес GPRS/EDGE SGSN  
— IP-адрес GPRS/EDGE GGSN  
— код зоны обслуживания (SAC) GPRS/EDGE  
— базовая станция, через которую установлено

— базовая станция, через которую завершено

— номер телефонной карты  
— IMSI мобильного абонента  
— идентификатор мобильной станции абонента  
— идентификатор мобильной станции абонента  
— номер модемного пула

```

191
192 DataEmailRecordContentIPDR ::= SEQUENCE {
193   mail-cdr-header DataNetworkCdrHeader,
194   mail-event      EmailEvent,
195   mail-sender     UTF8String (SIZE (1 .. 256)),
196   mail-receiver   EmailReceivers,
197   mail-cc         EmailReceivers,
198   mail-subject    UTF8String (SIZE (1 .. 256)),
199   mail-size       INTEGER (0 .. 4294967295),
200   attachments    INTEGER (0 ..1),
201   mail-servers   EmailServers,
202   mail-term-cause INTEGER (0 .. 16384),
203
204   mail-reply-to  UTF8String (SIZE (1 .. 256)) OPTIONAL,
205   mail-protocol  ENUMERATED {
206     smtp(0),
207     pop3(1),
208     imap(2),
209     web-mail(3)
210   } OPTIONAL
211 }
212
213 DataEmailRecordContentAAA ::= SEQUENCE {
214   mail-cdr-header DataNetworkCdrHeader,
215   mail-event      EmailEvent,
216   mail-aaa-info   IP-AAAInformation
217 }
218
219 EmailEvent ::= ENUMERATED
220 {
221   email-send(1),
222   email-receive(2),
223   email-download(3),
224   email-logon-attempt(4),
225   email-logon(5),
226   email-logon-failure(6),
227   email-logoff(7),
228   email-partial-download(8)
229 }

```

— заголовок IPDR-соединения  
— тип события  
— отправитель почтового сообщения  
— получатель почтового сообщения  
— получатель-копия почтового сообщения  
— тема почтового сообщения  
— размер почтового сообщения, включая  
— наличие прикрепленных файлов в письме (да/нет)  
— список текстовых наименований почтовых  
— причина завершения соединения  
— имя и адрес, куда следует адресовать  
— протокол, при помощи которого отправлено  
— заголовок IPDR-соединения  
— тип события  
— пользовательские реквизиты входа

```

230 EmailReceivers ::= SEQUENCE {
231   data SEQUENCE OF UTF8String (SIZE (1 .. 256))
232 }
233
234 EmailServers ::= SEQUENCE {
235   data SEQUENCE OF UTF8String (SIZE (1 .. 256))
236 }
237
238 — запись IPDR передачи мгновенных электронных сообщений между пользователями
239 dataImRecord TAGGED ::= {
240   OID { sorm-report-connection-im }
241   DATA SEQUENCE OF DataImRecordContent
242 }
243
244 DataImRecordContent ::= SEQUENCE {
245   im-cdr-header DataNetworkCdrHeader,
246   im-user-login UTF8String (SIZE (1 .. 20)),
247   im-user-password UTF8String (SIZE (1 .. 16)),
248   im-sender-screen-name UTF8String (SIZE (1 .. 32)),
249   im-sender-uid UTF8String (SIZE (1 .. 256)),
250   т.ч. для веб-чата
251   im-receivers ImReceivers,
252   im-size INTEGER (0 .. 4294967295),
253   im-term-cause INTEGER (0 .. 16384),
254   im-protocol [0] IMProtocol OPTIONAL
255 }
256
257 ImReceivers ::= SEQUENCE OF ImReceiver
258
259 ImReceiver ::= SEQUENCE {
260   im-receiver-screen-name UTF8String (SIZE (1 .. 32)),
261   im-receiver-uid UTF8String (SIZE (1 .. 256))
262   т.ч. для веб-чата
263 }
264
265 — запись IPDR передачи файловых данных
266 dataFileTransferRecord TAGGED ::= {
267   OID { sorm-report-connection-file-transfer }
268   DATA SEQUENCE OF DataFileTransferRecordContent
269 }
270
271 DataFileTransferRecordContent ::= SEQUENCE {

```

```

— заголовок IPDR-соединения
— учетная запись пользователя при подключении
— пользовательский пароль при подключении
— общедоступное имя отправителя
— пользовательский идентификатор отправителя (в
— список получателей сообщения
— размер данных сессии, байт
— причина завершения соединения
— общедоступное имя получателя
— пользовательский идентификатор получателя (в

```

```

272 file -cdr-header          DataNetworkCdrHeader,
273 file -server-name        UTF8String (SIZE (1 .. 256)),
274 file -user-name          UTF8String (SIZE (1 .. 64)),
275 file -user-password      UTF8String (SIZE (1 .. 256)),
276 file -server-type       BOOLEAN,
277 file -in-bytes-count     INTEGER (0 .. 18446744073709551615),
                             соединения управления так и передачи данных), байт
278 file -out-bytes-count   INTEGER (0 .. 18446744073709551615),
                             соединения управления так и передачи данных), байт
279 file -term-cause       INTEGER (0 .. 16384)
280 }
281
282 — запись IPDR терминального доступа к оборудованию
283 dataTermAccessRecord TAGGED ::= {
284   OID { sorm-report-connection-term-access }
285   DATA SEQUENCE OF DataTermAccessRecordContent
286 }
287
288 DataTermAccessRecordContent ::= SEQUENCE {
289   term-cdr-header        DataNetworkCdrHeader,
290   term-in-bytes-count   INTEGER (0 .. 18446744073709551615),
                             соединения управления так и передачи данных), байт
291   term-out-bytes-count  INTEGER (0 .. 18446744073709551615),
                             соединения управления так и передачи данных), байт
292   term-protocol         ENUMERATED {
293     telnet(0),
294     ssh(1),
295     scp(2)
296   } OPTIONAL
297 }
298
299 — запись IPDR прочих данных, принимаемых, получаемых пользователем при помощи закрытых протоколов обмена
300 dataRawFlowsRecord TAGGED ::= {
301   OID { sorm-report-connection-raw-flows }
302   DATA SEQUENCE OF DataRawFlowsRecordContent
303 }
304
305 DataRawFlowsRecordContent ::= SEQUENCE {
306   flow-cdr-header       DataNetworkCdrHeader,
307   flow-in-bytes-count   INTEGER (0 .. 18446744073709551615),
                             соединения управления так и передачи данных), байт
308   flow-out-bytes-count  INTEGER (0 .. 18446744073709551615),
                             соединения управления так и передачи данных), байт

```

— заголовок IPDR-соединения  
 — название сервера  
 — имя пользователя, наименование учетной записи  
 — пользовательский пароль  
 — режим работы файлового сервера (ACTIVE/  
 — объем переданных данных (включает как  
 — объем принятых данных (включает как  
 — причина завершения соединения

— заголовок IPDR-соединения  
 — объем переданных данных (включает как  
 — объем принятых данных (включает как  
 — протокол удаленного доступа к оборудованию

— заголовок IPDR-соединения  
 — объем переданных данных (включает как  
 — объем принятых данных (включает как

```

309 flow-protocol ENUMERATED {
310     ip(0),
311     udp(1),
312     tcp(2)
313 } OPTIONAL
314 }
315
316 — запись IPDR HTTP-обращения к информационному ресурсу сети связи (сайт, портал)
317 dataResourceRecord TAGGED ::= {
318     OID { sorm-report-connection-resource }
319     DATA SEQUENCE OF DataResourceRecordContent
320 }
321
322 DataResourceRecordContent ::= SEQUENCE {
323     res-cdr-header          DataNetworkCdrHeader,
324     res-url                 UTF8String (SIZE (1 .. 1024)),
325     res-bytes-count        INTEGER (0 .. 18446744073709551615),
326                             соединения (включает как соединения управления так и передачи данных), байт
327     res-term-cause         INTEGER (0 .. 16384),
328                             причина завершения соединения
329     res-aaa-info [0]       IP-AAAInformation OPTIONAL
330                             пользовательские реквизиты входа в
331                             информационный ресурс
332 }
333
334 — запись IPDR голосовой связи посредством сети передачи данных
335 dataVoipRecord TAGGED ::= {
336     OID { sorm-report-connection-voip }
337     DATA SEQUENCE OF DataVoipRecordContent
338 }
339
340 DataVoipRecordContent ::= SEQUENCE {
341     voip-cdr-header          DataNetworkCdrHeader,
342     voip-session-id          UTF8String (SIZE (1..64)),
343     voip-conference-id       UTF8String (SIZE (1..64)),
344     voip-duration            INTEGER (0 .. 86399),
345     voip-originator-name     UTF8String (SIZE (1 .. 32)),
346     voip-service-type        INTEGER (0 .. 255),
347     voip-calling-number      DataVoipNumber,
348     voip-called-number       DataVoipNumber,
349     voip-in-bytes-count      INTEGER (0 .. 18446744073709551615),
350                             соединения управления так и передачи данных), байт
351     voip-out-bytes-count     INTEGER (0 .. 18446744073709551615),
352                             соединения управления так и передачи данных), байт
353     voip-fax                 BOOLEAN,
354                             .38)
355 }
356
357 — протокол передачи данных
358 }
359
360 — заголовок IPDR-соединения
361 — Наименование информационного ресурса
362 — объем принятых и переданных данных в
363 — причина завершения соединения
364 — пользовательские реквизиты входа в
365
366 — заголовок CDR-соединения
367 — идентификатор сессии/call-id
368 — идентификатор конференции
369 — длительность разговора, сек.
370 — общедоступное имя инициатора связи
371 — способ подключения
372 — номер вызывающего абонента
373 — номер вызываемого абонента
374 — объем переданных данных (включает как
375 — объем принятых данных (включает как
376 — была попытка передачи факсовой информации (Т

```



```

348 voip-term-cause          INTEGER (0 .. 16384) ,
349
350 inbound-bunch [0]       Bunch OPTIONAL,
351 outbound-bunch [1]     Bunch OPTIONAL,
352 voip-gateways [2]     SEQUENCE OF IPAddress OPTIONAL,
                          соединение
353
354 voip-protocol [3]     VoipProtocol OPTIONAL,
355
356 supplement-service-id [4] INTEGER (0 .. 65535) OPTIONAL
357 }
358
359
360 --- запись IPDR трансляции сетевых адресов
361 dataAddressTranslationRecord TAGGED ::= {
362   OID { sorm-report-connection-address-translations }
363   DATA SEQUENCE OF DataAddressTranslationRecordContent
364 }
365
366 DataAddressTranslationRecordContent ::= SEQUENCE {
367   telco-id              TelcoID,
368   point-id              INTEGER (0 .. 1000) ,
                          которой получена запись о соединении
369   translation-time      DateAndTime,
370   record-type           ENUMERATED {
371     session-start (0) ,
372     session-end (1)
373   } ,
374
375   private-ip           NetworkPeerInfo,
376   public-ip            NetworkPeerInfo,
377   destination-ip      NetworkPeerInfo,
378   translation-type     ENUMERATED {
379     static-nat (0) ,
380     dynamic-nat (1) ,
381     source-nat (2) ,
382     destination-nat (3) ,
383     pat (4)
384   }
385 }
386
387
388 --- Data network header
389 DataNetworkCdrHeader ::= SEQUENCE {

```

--- причина завершения соединения

--- входящий пучок

--- исходящий пучок

--- идентификаторы медиашлюзов, обслуживавших

--- ДБО при соединении

--- идентификатор оператора связи или филиала

--- идентификатор точки подключения к сети передачи данных, с

--- дата и время трансляции

--- тип записи о трансляции сетевого адреса

--- начало сессии трансляции

--- окончание сессии трансляции

--- внутренний адрес

--- внешний адрес

--- адрес назначения

--- тип трансляции сетевых адресов

--- статическая

--- динамическая

--- источника

--- получателя

--- адрес-порт

```

390 id TAGGED.&id ({DataNetworkHeaderVariants}),
391 data TAGGED.&Data ({DataNetworkHeaderVariants}){@id}
392 }
393
394 DataNetworkHeaderVariants TAGGED ::= { dataNetworkCdrHeader }
395
396 dataNetworkCdrHeader TAGGED ::= {
397   OID { sorm-report-connection-ipdr-header }
398   DATA DataNetworkCdrHeaderData
399 }
400
401 DataNetworkCdrHeaderData ::= SEQUENCE {
402   telco-id
403   begin-connection-time DateAndTime,
404   end-connection-time DateAndTime,
405   client-info NetworkPeerInfo,
406   server-info NetworkPeerInfo,
407   protocol-code INTEGER (0 .. 65535),
408   номер порта для TCP/UDP
409   point-id INTEGER (0 .. 1000) OPTIONAL
410   передачи данных, с которой получена запись о соединении
411 }
412
413 — информация о входе в ресурс
414 IP-AAAInformation ::= SEQUENCE {
415   username UTF8String (SIZE (0..64)),
416   aaaResult IP-AAAAResult OPTIONAL
417 }
418
419 — результат операции входа в ресурс
420 IP-AAAAResult ::= ENUMERATED {
421   aaaUnknown(1),
422   aaaFailed(2),
423   aaaSucceeded(3)
424 }
425
426 — идентификатор оператора связи или филиала
427 — дата и время начала соединения
428 — дата и время завершения соединения
429 — информация о клиенте (IP/порт)
430 — информация о сервере (IP/порт)
431 — код протокола в соответствии с RFC1700 либо
432
433 — идентификатор точки подключения к сети
434
435 — пользовательский логин
436 — результат операции входа
437 — результат неизвестен
438 — неудачная попытка входа
439 — успешный вход

```

## ReportsLocations.asn

```

1 ReportsLocations DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3

```

```

4 EXPORTS LocationReport;
5
6 IMPORTS DateAndTime
7 FROM Sorm
8
9 TelcoID
10 FROM Dictionaries
11
12 Location
13 FROM Locations
14
15 ReportedIdentifier
16 FROM ReportedIdentifiers;
17
18 LocationReport ::= SEQUENCE OF ValidateLocationRecord
19
20 ValidateLocationRecord ::= SEQUENCE {
21 telco-id TelcoID,
22 connection-time DateAndTime,
23 ident ReportedIdentifier,
24 connection-location Location
25 }
26
27 END

```

— идентификатор оператора связи или филиала  
 — время определения местоположения  
 — идентификатор мобильного абонента  
 — местоположение мобильного абонента

---

### ReportsNonFormalized.asn

```

1 ReportsNonFormalized DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS NonFormalizedReport;
5
6 IMPORTS
7 EntityId,
8 NonFormalizedEntityAttributeData
9 FROM TasksNonFormalized
10
11 StandardInterval
12 FROM ReportsPresence;
13
14 — отчет задачи по обработке неформализованных данных
15 NonFormalizedReport ::= CHOICE {
16 nonformalized-report [0] NonFormalizedRecords,
    данных

```

— отчет по задаче обработки неформализованных

```

17 nonformalized-presense [1] NonFormalizedPresenseInfo
18     заданного вида
19 }
20
21 NonFormalizedRecords ::= SEQUENCE OF NonFormalizedRecord
22
23 NonFormalizedPresenseInfo ::= SEQUENCE OF StandardInterval
24     неформализованные данные
25
26 NonFormalizedRecord ::= SEQUENCE OF NonFormalizedEntityAttributeData
27     запись неформализованных данных
28
29 END

```

## ReportsPayments.asn

```

1 ReportsPayments DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS PaymentsReport;
5
6 IMPORTS TAGGED,
7     sorm-report-payment-bank-transaction,
8     sorm-report-payment-express-pays,
9     sorm-report-payment-terminal-pays,
10    sorm-report-payment-service-center,
11    sorm-report-payment-cross-account,
12    sorm-report-payment-telephone-card,
13    sorm-report-payment-balance-fillups,
14    sorm-report-payment-division-transfer,
15    sorm-report-payment-bank-card-transfer,
16    sorm-report-payment-bank-account-transfer
17 FROM Classification
18
19     DateAndTime
20 FROM Sorm
21
22     TelcoID
23 FROM Dictionaries
24
25     Location
26 FROM Locations

```

```

27         ReportedIdentifier
28     FROM ReportedIdentifiers
29
30
31         ReportedAddress
32     FROM Addresses;
33
34 PaymentsReport ::= SEQUENCE {
35     id TAGGED.&id ({ReportedPaymentsVariants}),
36     data TAGGED.&Data ({ReportedPaymentsVariants}@id})
37 }
38
39 — варианты запрашиваемых параметров связей
40 ReportedPaymentsVariants TAGGED ::= {
41     bankTransactionReport
42 | expressCardReport
43 | publicTerminalReport
44 | serviceCenterReport
45 | crossAccountReport
46 | telephoneCardReport
47 | balanceFillupReport
48 | bankDivisionTransferReport
49 | bankCardTransferReport
50 | bankAccountTransferReport
51 }
52
53 — отчет задачи на поиск пополнения баланса через банковский перевод
54 bankTransactionReport TAGGED ::= {
55     OID { sorm-report-payment-bank-transaction }
56     DATA SEQUENCE OF BankTransactionRecord
57 }
58
59 BankTransactionRecord ::= SEQUENCE {
60     telco-id          TelcoID,
61     device-id        ReportedIdentifier,
62     date-time-fillup DateAndTime,
63     bank-account     UTF8String (SIZE (1 .. 64)),
64     bank-name        UTF8String (SIZE (1 .. 512)),
65                                     перевод
66     bank-address     ReportedAddress,
67     amount           UTF8String (SIZE (1 .. 64))
68 }
69 — отчет задачи на поиск пополнения баланса через карты экспресс-оплаты

```

```

— идентификатор оператора связи или филиала
— идентификатор абонента
— время и дата пополнения баланса
— номер банковского счета, с которого совершен платеж
— наименование банка, со счета которого совершен
— адрес банка, со счета которого совершен перевод
— сумма перевода

```

```

70 expressCardReport TAGGED ::= {
71   OID { sorm-report-payment-express-pays }
72   DATA SEQUENCE OF ExpressPaysRecord
73 }
74
75 ExpressPaysRecord ::= SEQUENCE {
76   telco-id          TelcoID,
77   device-id        ReportedIdentifier,
78   date-time-fillup DateAndTime,
79   card-number      UTF8String (SIZE (1 .. 64)),
80   amount           UTF8String (SIZE (1 .. 64))
81 }
82
83 --- отчет задачи на поиск пополнения баланса через терминалы моментальных платежей
84 publicTerminalReport TAGGED ::= {
85   OID { sorm-report-payment-terminal-pays }
86   DATA SEQUENCE OF PublicTerminalRecord
87 }
88
89 PublicTerminalRecord ::= SEQUENCE {
90   telco-id          TelcoID,
91   device-id        ReportedIdentifier,
92   date-time-fillup DateAndTime,
93   terminal-id       UTF8String (SIZE (1 .. 64)),
94   terminal-number   NumericString (SIZE (2 .. 20)),
95   terminal-address  ReportedAddress,
96   amount           UTF8String (SIZE (1 .. 64)),
97   location         Location OPTIONAL
98 }
99
100 --- отчет задачи на поиск пополнения баланса через центры обслуживания клиентов
101 serviceCenterReport TAGGED ::= {
102   OID { sorm-report-payment-service-center }
103   DATA SEQUENCE OF ServiceCenterRecord
104 }
105
106 ServiceCenterRecord ::= SEQUENCE {
107   telco-id          TelcoID,
108   device-id        ReportedIdentifier,
109   date-time-fillup DateAndTime,
110   center-id        UTF8String (SIZE (1 .. 64)),
111   center-address   ReportedAddress,
112   amount           UTF8String (SIZE (1 .. 64))
113 }

```

--- идентификатор оператора связи или филиала  
 --- идентификатор абонента  
 --- время и дата пополнения баланса  
 --- номер карты  
 --- сумма перевода

--- идентификатор оператора связи или филиала  
 --- идентификатор абонента  
 --- время и дата пополнения баланса  
 --- идентификатор терминала  
 --- номер терминала  
 --- адрес терминала  
 --- сумма перевода  
 --- адрес совершения платежа

--- идентификатор оператора связи или филиала  
 --- идентификатор абонента  
 --- время и дата пополнения баланса  
 --- идентификатор центра обслуживания клиентов  
 --- адрес центра обслуживания клиентов  
 --- сумма перевода

```

114
115 — отчет задачи на поиск пополнения баланса посредством снятия денег со счета другого абонента
116 crossAccountReport TAGGED ::= {
117   OID { sorm-report-payment-cross-account }
118   DATA SEQUENCE OF CrossAccountRecord
119 }
120
121 CrossAccountRecord ::= SEQUENCE {
122   telco-id          TelcoID,
123   device-id        ReportedIdentifier,
124   date-time-fillup DateAndTime,
125   donated-id       ReportedIdentifier,
126   amount           UTF8String (SIZE (1 .. 64))
127 }
128
129
130 — отчет задачи на поиск пополнения баланса через телефонные карты
131 telephoneCardReport TAGGED ::= {
132   OID { sorm-report-payment-telephone-card }
133   DATA SEQUENCE OF ValidateTelephoneCardRecord
134 }
135
136 ValidateTelephoneCardRecord ::= SEQUENCE {
137   telco-id          TelcoID,
138   activator-device-id ReportedIdentifier,
139   date-time-fillup DateAndTime,
140   card-number       NumericString (SIZE (2 .. 20)),
141   amount            UTF8String (SIZE (1 .. 64))
142 }
143
144
145 — отчет задачи на поиск пополнения баланса личного счета абонента
146 balanceFillupReport TAGGED ::= {
147   OID { sorm-report-payment-balance-fillups }
148   DATA SEQUENCE OF ValidateBalanceFillupRecord
149 }
150
151 ValidateBalanceFillupRecord ::= SEQUENCE {
152   telco-id          TelcoID,
153   pay-type-id       INTEGER (0 .. 65535),
154   device-id         ReportedIdentifier,
155   date-time-fillup DateAndTime,
156   amount            UTF8String (SIZE (1 .. 64)),
157   pay-parameters   UTF8String (SIZE (1 .. 512)) OPTIONAL

```

```

158 }
159
160 — отчет задачи по переводу средств со счета абонента для их снятия в отделении банка
161 bankDivisionTransferReport TAGGED ::= {
162   OID { sorm-report-payment-bank-division-transfer }
163   DATA SEQUENCE OF ValidateBankDivisionTransferRecord
164 }
165
166 ValidateBankDivisionTransferRecord ::= SEQUENCE {
167   telco-id          TelcoID,
168   donated-id       ReportedIdentifier,
169   date-time-fillup DateAndTime,
170   person-recvied  UTF8String (SIZE (1 .. 512)),
171                   информация)
172   bank-name        UTF8String (SIZE (1 .. 256)),
173   bank-division-name UTF8String (SIZE (1 .. 512)),
174   bank-division-address ReportedAddress,
175   amount           UTF8String (SIZE (1 .. 64))
176 }
177
178 — отчет задачи по переводу средств со счета абонента на банковскую карту
179 bankCardTransferReport TAGGED ::= {
180   OID { sorm-report-payment-bank-card-transfer }
181   DATA SEQUENCE OF ValidateBankCardTransferRecord
182 }
183
184 ValidateBankCardTransferRecord ::= SEQUENCE {
185   telco-id          TelcoID,
186   donated-id       ReportedIdentifier,
187   bank-card-id     NumericString (SIZE (1 .. 12)),
188   date-time-fillup DateAndTime,
189   amount           UTF8String (SIZE (1 .. 64))
190 }
191
192 — отчет задачи по переводу средств со счета абонента на счет в банке
193 bankAccountTransferReport TAGGED ::= {
194   OID { sorm-report-payment-bank-account-transfer }
195   DATA SEQUENCE OF ValidateBankAccountTransferRecord
196 }
197
198 ValidateBankAccountTransferRecord ::= SEQUENCE {
199   telco-id          TelcoID,
200   donated-id       ReportedIdentifier,

```



```

201 bank-name
202 bank-account
203 date-time-fillup
204 amount
205 }
206
207
208
209 END

```

UTF8String (SIZE (1 .. 256)),  
 UTF8String (SIZE (1 .. 64)),  
 DateAndTime,  
 UTF8String (SIZE (1 .. 64))

— наименование банка получателя  
 — номер банковского счета получателя  
 — время и дата перевода средств  
 — сумма перевода

### ReportsPresense.asn

```

1 ReportsPresense DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS PresenseReport,
5 StandardInterval;
6
7 IMPORTS FindRange,
8 DateAndTime
9 FROM Sorm
10
11 TelcoID
12 FROM Dictionaries
13
14 Location
15 FROM Locations
16
17 ReportedIdentifier
18 FROM ReportedIdentifiers
19
20 NetworkType
21 FROM NetworkIdentifiers
22
23 TAGGED,
24 sorm-report-presense-abonents,
25 sorm-report-presense-connections,
26 sorm-report-presense-payments,
27 sorm-report-presense-dictionaries,
28 sorm-report-presense-locations
29 FROM Classification;
30
31 — отчет по запросу наличия информации
32 PresenseReport ::= SEQUENCE {

```

```

33 id TAGGED.&id ( {ReportedPresencesVariants} ),
34 data TAGGED.&Data ( {ReportedPresencesVariants} { @id } )
35 }
36
37 ReportedPresencesVariants TAGGED ::= {
38   subsPresence
39   | connectionsPresence
40   | paymentsPresence
41   | dictionariesPresence
42   | locationPresence
43 }
44
45
46 --- отчет по наличию информации по абонентам и их идентификаторам.
47 --- Для каждого стандарта может быть указано более одного, либо ни
48 --- одного интервала (фактические периоды наличия информации);
49 subsPresence TAGGED ::= {
50   OID { sorm-report-presence-abonents }
51   DATA SEQUENCE OF StandardInterval
52 }
53
54 --- отчет по наличию информации по соединениям. Для каждого стандарта может быть указано более одного, либо ни одного, интервала (
55   фактические периоды наличия информации);
56 connectionsPresence TAGGED ::= {
57   OID { sorm-report-presence-connections }
58   DATA SEQUENCE OF ConnectionsPresenceRecord
59 }
60
61 ConnectionsPresenceRecord ::= SEQUENCE {
62   standard-interval StandardInterval ,
63   data-type ENUMERATED {
64     ИС СОРМ
65     telephone-pstn (0) ,
66     telephone-mobile (1) ,
67     pager (2) ,
68     data-aaa (3) ,
69     data-resource (4) ,
70     портал
71     data-email (5) ,
72     data-im (6) ,
73     пользователями
74     data-voip (7) ,
75     data-file (8) ,
76     data-term (9) ,

```

```

--- вид соединений передачи данных, информация по которым есть в
--- телефонные ТФФП-соединения
--- телефонные соединения мобильной связи
--- соединения сети персонального радиовызова
--- подключения/отключения абонента к сети связи
--- HTTP-обращения к информационному ресурсу сети связи (сайт,
--- передача почтовых сообщений
--- передача мгновенных электронных сообщений между
--- голосовая связь посредством сети передачи данных
--- передача файловых данных
--- терминальный доступ к оборудованию

```

```

73  data-raw (10),
74  помощи закрытых протоколов обмена
75  data-address-translations (11)
76  }
77  }
78
79  paymentsPresense TAGGED ::= {
80  OID { sorm-report-presense-payments }
81  DATA SEQUENCE OF StandardInterval
82  абонентов
83  }
84
85  --- отчет о наличии информации справочников в ИС. Если какой-либо из справочников не публикуется ИС СОРМ, запись о нем
86  отсутствует
87  dictionariesPresense TAGGED ::= {
88  OID { sorm-report-presense-dictionaries }
89  DATA SEQUENCE OF DictionaryInfo
90  }
91  --- запись отчета о наличии информации справочников
92  DictionaryInfo ::= SEQUENCE {
93  telco-id          TelcoID,
94  dict              ObjectDescriptor,
95  count             INTEGER (1 .. 4294967295),
96  change-dates     FindRange
97  справочнике
98  }
99
100 --- отчет по наличию информации по местоположению абонентов
101 locationPresence TAGGED ::= {
102  OID { sorm-report-presense-locations }
103  DATA SEQUENCE OF StandardInterval
104  }
105
106 --- интервал времени, на котором имеются данные по абонентам, соединениям и платежам, в рамках стандарта связи
107 StandardInterval ::= SEQUENCE {
108  telco-id          TelcoID,
109  standard          NetworkType,
110  range             FindRange,
111  count             INTEGER OPTIONAL
112  }

```

--- прочие данные, принимаемые, получаемые пользователем при  
--- трансляции сетевых адресов  
--- описание имеющейся информации по пополнениям балансов  
--- идентификатор оператора связи или филиала  
--- тип справочника, по которому есть информация  
--- (идентификатор запроса справочника) (Requested...)  
--- количество записей в справочнике  
---Ц минимальное и максимальное дата/время изменения записей в  
--- идентификатор оператора связи или филиала  
--- стандарт связи, информация по которому имеется в ИС  
--- интервал времени, на который имеются данные в ИС  
--- количество записей

113  
114

**END**

---

RequestedAbonents.asn

```
1 RequestedAbonents DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS RequestedAbonent ;
5
6 IMPORTS TAGGED,
7     sorm-request-abonent-person ,
8     sorm-request-abonent-organization
9     FROM Classification
10
11     RequestedAddress FROM Addresses;
12
13
14 RequestedAbonent ::= SEQUENCE {
15     id TAGGED.&id ({RequestedAbonentVariants}),
16     data TAGGED.&Data ({RequestedAbonentVariants}){@id}}
17 }
18
19 — варианты запрашиваемых идентификаторов
20 RequestedAbonentVariants TAGGED ::= {
21     requestedPerson |
22     requestedOrganization
23 }
24
25 — поля параметра запроса на физическое лицо
26 requestedPerson TAGGED ::= {
27     OID { sorm-request-abonent-person }
28     DATA RequestedPerson
29 }
30
31 RequestedPerson ::= SEQUENCE {
32     given-name [0] UTF8String (SIZE (1 .. 128)) OPTIONAL,
33     initial [1] UTF8String (SIZE (1 .. 128)) OPTIONAL,
34     family-name [2] UTF8String (SIZE (1 .. 128)) OPTIONAL,
35     passport-info [3] RequestedPassport OPTIONAL,
36     address [4] RequestedAddress OPTIONAL,
37     icc [5] NumericString (SIZE (19 .. 20)) OPTIONAL,
38     contract [6] UTF8String (SIZE (1 .. 64)) OPTIONAL
39 }
```

— физическое лицо  
— юридическое лицо – организация

— ИМЯ  
— отчество  
— фамилия  
— паспортные ланные  
— адресные данные  
— идентификатор SIM-карты абонента  
— номер договора

```

40 }
41 --- поля паспортных данных
42 RequestedPassport ::= SEQUENCE {
43   doc-type-id [0] INTEGER (0 .. 65535) OPTIONAL,
44   passport-serial [1] UTF8String (SIZE (1..16)) OPTIONAL,
45   passport-number [2] NumericString (SIZE (1..16)) OPTIONAL
46 }
47
48
49 --- поля параметра запроса на юридическое лицо
50 requestedOrganization TAGGED ::= {
51   OID { sorm-request-abonent-organization }
52   DATA RequestedOrganization
53 }
54
55 RequestedOrganization ::= SEQUENCE {
56   full-name [0] UTF8String (SIZE (1 .. 128)) OPTIONAL,
57   address [1] RequestedAddress OPTIONAL,
58   inn [2] NumericString (SIZE (1 .. 64)) OPTIONAL,
59   internal-user [3] UTF8String (SIZE (1 .. 64)) OPTIONAL,
60   contract [4] UTF8String (SIZE (1 .. 64)) OPTIONAL
61 }
62
63
64
65 END

```

--- идентификатор типа документа, удостоверяющего  
 --- серия паспорта  
 --- номер паспорта  
 --- полное наименование организации  
 --- юридический адрес организации  
 --- ИНН  
 --- внутренний пользователь  
 --- номер договора

---

## RequestedConnections.asn

```

1 RequestedConnections DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS RequestedConnection;
5
6 IMPORTS TAGGED,
7   sorm-request-connection-pstn,
8   sorm-request-connection-mobile,
9   sorm-request-connection-entrance,
10  sorm-request-connection-aaa-login,
11  sorm-request-connection-resource,
12  sorm-request-connection-email,
13  sorm-request-connection-im,
14  sorm-request-connection-voip,

```

```

15 sorm-request-connection-file-transfer,
16 sorm-request-connection-term-access,
17 sorm-request-connection-raw-flows,
18 sorm-request-connection-address-translations
19 FROM Classification
20
21 requestedPagerIdentifier,
22 requestedPstnIdentifier,
23 requestedGsmIdentifier,
24 requestedCdmaIdentifier
25 FROM RequestedIdentifiers
26
27 PhoneAbonentType
28 FROM Dictionaries
29
30 Bunch,
31 DataNetworkEquipment,
32 IPAddress,
33 NetworkPeerInfo,
34 DataVoipNumber,
35 VoipProtocol,
36 IMPProtocol
37 FROM NetworkIdentifiers
38
39 Location
40 FROM Locations;
41
42 RequestedConnection ::= SEQUENCE {
43   id TAGGED&id ({RequestedConnectionVariants}),
44   data TAGGED.&Data ({RequestedConnectionVariants}){@id}
45 }
46
47
48 — варианты запрашиваемых параметров связей
49 RequestedConnectionVariants TAGGED ::= {
50   requestedPagerIdentifier
51 | requestedConnectionPstn
52 | requestedConnectionMobile
53 | requestedConnectionEntrance
54 | requestedAAALogin
55 | requestedResource
56 | requestedEmail
57 | requestedIm
58 | requestedVoip

```

— параметры соединений абонента ТФОП  
— параметры соединений абонента сети мобильной связи  
— параметры соединений подключения к сети связи

```

59 | requestedFileTransfer
60 | requestedTermAccess
61 | requestedRawFlows
62 | requestedAddressTranslations
63 }
64
65 requestedConnectionEntrance TAGGED ::= {
66   OID { sorm-request-connection-entrance }
67   DATA CHOICE {
68     directory-number [0] UTF8String (SIZE (2 .. 32)),
69     imsi [1] NumericString (SIZE (10 .. 18)),
70     ip-address [2] IPAddress,
71     mac [3] OCTET STRING (SIZE (6))
72 }
73 }
74
75 — параметры соединений абонента ТХЮП
76 requestedConnectionPstn TAGGED ::= {
77   OID { sorm-request-connection-pstn }
78   DATA CHOICE {
79     duration [0] INTEGER (0 .. 86399),
80     call-type-id [1] INTEGER (0 .. 65535),
81     in-abonent-type [2] PhoneAbonentType,
82     out-abonent-type [3] PhoneAbonentType,
83     switch-id [4] UTF8String (SIZE (1 .. 128)),
84     inbound-bunch [5] INTEGER (0 .. 4294967295),
85     outbound-bunch [6] INTEGER (0 .. 4294967295),
86     border-switch-id [7] UTF8String (SIZE (1 .. 128)),
87     term-cause [8] INTEGER (0 .. 16384),
88     supplement-service-id [9] INTEGER (0 .. 65535),
89     phone-card-number [10] NumericString (SIZE (1.. 20)),
90     in-info [11] RequestedConnectionPstnIdentifier,
91     out-info [12] RequestedConnectionPstnIdentifier,
92     forwarding-identifier [13] UTF8String (SIZE (2 .. 32))
93 }
94 }
95
96 — Идентификаторы PSTN
97 RequestedConnectionPstnIdentifier ::= SEQUENCE {
98   id TAGGED.&id ({RequestedConnectionPstnIdentifierVariants}),
99   data TAGGED.&Data ({RequestedConnectionPstnIdentifierVariants}){@id}
100 }
101
102 RequestedConnectionPstnIdentifierVariants TAGGED ::= {

```

```

— телефонный номер в международном формате
— идентификатор мобильного абонента
— IP-адрес абонента
— MAC-адрес устройства абонента

— время соединения
— тип соединения
— тип вызывающего абонента
— тип вызываемого абонента
— код коммутатора обслуживающего вызов
— входящий пучок
— исходящий пучок
— код пограничного коммутатора
— причина завершения соединения
— ДФО при соединении
— номер телефонной карты
— идентификаторы вызывающего абонента
— идентификаторы вызываемого абонента
— телефонный номер при переадресации

```

```

103     requestedPstnIdentifier
104 }
105
106
107 requestedConnectionMobile TAGGED ::= {
108     OID { sorm-request-connection-mobile }
109     DATA CHOICE {
110         duration [0] INTEGER (0 .. 86399),
111         call-type-id [1] INTEGER (0 .. 65535),
112         supplement-service-id [2] INTEGER (0 .. 65535),
113         in-abonent-type [3] PhoneAbonentType,
114         out-abonent-type [4] PhoneAbonentType,
115         switch-id [5] UTF8String (SIZE (1 .. 128)),
116
117         inbound-bunch [6] Bunch,
118         outbound-bunch [7] Bunch,
119         border-switch-id [8] UTF8String (SIZE (1 .. 128)),
120         roaming-partner-id [9] INTEGER (0 .. 65535),
121         term-cause [10] INTEGER (0 .. 16384),
122         in-info [11] RequestedConnectionMobileIdentifier,
123         in-end-location [12] Location,
124         in-begin-location [13] Location,
125         dialed-digits [14] UTF8String (SIZE (2 .. 32)),
126         out-info [15] RequestedConnectionMobileIdentifier,
127         out-begin-location [16] Location,
128         out-end-location [17] Location,
129         forwarding-identifier [18] UTF8String (SIZE (2 .. 32))
130     }
131 }
132
133 -- Идентификаторы мобильных абонентов
134 RequestedConnectionMobileIdentifier ::= SEQUENCE {
135     id TAGGED.&id ({RequestedConnectionMobileIdentifierVariants}),
136     data TAGGED.&Data ({RequestedConnectionMobileIdentifierVariants}){@id}
137 }
138
139 RequestedConnectionMobileIdentifierVariants TAGGED ::= {
140     requestedGsmIdentifier |
141     requestedCdmaIdentifier
142 }
143
144 requestedAAALogin TAGGED ::= {
145     OID { sorm-request-connection-aaa-login }
146     DATA CHOICE {

```



```

147 point-id [0] INTEGER (0 .. 1000),
    данных, с которой получены записи
148 login-type [1] ENUMERATED {
149 connect (0),
150 disconnect (1)
151 },
152 user-equipment [2] DataNetworkEquipment,
153 allocated-ip [3] IPAddress,
154 user-name [4] UTF8String (SIZE (1 .. 64)),
155 user-password [5] UTF8String (SIZE (1 .. 16)),
156 connect-type [6] INTEGER (1 .. 65535),
    порта для TCP/UDP
157 calling-number [7] UTF8String (SIZE (2 .. 18)),
158 called-number [8] UTF8String (SIZE (2 .. 18)),
159 nas [9] NetworkPeerInfo,
160 apn [10] UTF8String (SIZE (1 .. 32)),
161 sgsn-ip [11] IPAddress,
162 ggsn-ip [12] IPAddress,
163 service-area-code [13] INTEGER (0 .. 65535),
164 location-start [14] Location,
    (передача данных посредством подвижной сети связи)
165 location-end [15] Location,
    передача данных посредством подвижной сети связи)
166 phone-card-number [16] NumericString (SIZE (20)),
167 imsi [17] NumericString (SIZE (10 .. 18)),
168 imei [18] NumericString (SIZE (10 .. 18)),
169 esn [19] NumericString (SIZE (10 .. 18)),
170 pool-number [20] UTF8String (SIZE (2 .. 18))
171 }
172 }
173
174 requestedResource TAGGED ::= {
175 OID { sorm-request-connection-resource }
176 DATA CHOICE {
177 point-id [0] INTEGER (0 .. 1000),
    данных, с которой получены записи
178 client-info [1] NetworkPeerInfo,
179 server-info [2] NetworkPeerInfo,
180 url [3] UTF8String (SIZE (1 .. 1024)),
181 term-cause [4] INTEGER (0 .. 16384)
182 }
183 }
184
185 requestedEmail TAGGED ::= {

```

```

— идентификатор точки подключения к сети передачи
— тип соединения
— подключение к сети передачи данных
— отключение от сети передачи данных
— идентификатор пользовательского оборудования
— выделенный динамический IP-адрес
— имя пользователя (логин)
— пользовательский пароль
— код протокола в соответствии с RFC1700 либо номер
— вызывающий номер
— вызываемый номер
— IP-адрес/порт NAS-сервера
— наименование точки доступа (Access Point Name)
— IP-адрес GPRS/EDGE SGSN
— IP-адрес GPRS/EDGE GGSN
— код зоны обслуживания (SAC) GPRS/EDGE
— базовая станция, через которую установлено соединение
— базовая станция, через которую завершено соединение (
— номер телефонной карты
— IMSI мобильного абонента
— идентификатор мобильной станции абонента
— идентификатор мобильной станции абонента
— номер модемного пула
— идентификатор точки подключения к сети передачи
— идентификатор абонента сети передачи данных
— идентификатор сервера сети передачи данных
— Наименование информационного ресурса
— причина завершения соединения

```

```

186 OID { sorm-request-connection-email }
187 DATA CHOICE {
188   point-id [0] INTEGER (0 .. 1000),
      данных, с которой получены записи
189   client-info [1] NetworkPeerInfo,
190   server-info [2] NetworkPeerInfo,
191   sender [3] UTF8String (SIZE (1 .. 256)),
192   receiver [4] UTF8String (SIZE (1 .. 256)),
193   cc [5] UTF8String (SIZE (1 .. 256)),
194   subject [6] UTF8String (SIZE (1 .. 256)),
195   attachments [7] BOOLEAN,
196   mail-server [8] UTF8String (SIZE (1 .. 256)),
      отправлено письмо (в т.ч. сервер веб-почты)
197   term-cause [9] INTEGER (0 .. 16384)
198 }
199 }
200
201 requestedIm TAGGED ::= {
202   OID { sorm-request-connection-im }
203   DATA CHOICE {
204     point-id [0] INTEGER (0 .. 1000),
      данных, с которой получены записи
205     client-info [1] NetworkPeerInfo,
206     server-info [2] NetworkPeerInfo,
207     user-login [3] UTF8String (SIZE (1 .. 20)),
208     user-password [4] UTF8String (SIZE (1 .. 16)),
209     sender-screen-name [5] UTF8String (SIZE (1 .. 32)),
210     sender-uin [6] UTF8String (SIZE (1 .. 256)),
      для веб-чата)
211     receiver-screen-name [7] UTF8String (SIZE (1 .. 32)),
212     receiver-uin [8] UTF8String (SIZE (1 .. 256)),
      веб-чата)
213     protocol [9] IMPProtocol,
214     term-cause [10] INTEGER (0 .. 16384)
215   }
216 }
217
218 requestedVoip TAGGED ::= {
219   OID { sorm-request-connection-voip }
220   DATA CHOICE {
221     point-id [0] INTEGER (0 .. 1000),
      данных, с которой получены записи
222     client-info [1] NetworkPeerInfo,
223     server-info [2] NetworkPeerInfo,

```

— идентификатор точки подключения к сети передачи

— идентификатор абонента сети передачи данных

— отправитель почтового сообщения

— получатель почтового сообщения

— получатель-копия почтового сообщения

— тема почтового сообщения

— наличие прикрепленных файлов в письме (да/нет)

— текстовое наименование сервера, через который

— причина завершения соединения

— идентификатор точки подключения к сети передачи

— идентификатор абонента сети передачи данных

— идентификатор сервера сети передачи данных

— учетная запись пользователя при подключении

— пользовательский пароль при подключении

— общедоступное имя отправителя

— пользовательский идентификатор отправителя (в т.ч.

— общедоступное имя получателя

— пользовательский идентификатор получателя (в т.ч. для

— причина завершения соединения

— идентификатор точки подключения к сети передачи

— идентификатор абонента сети передачи данных

— идентификатор сервера сети передачи данных

```

224 duration [3] INTEGER (0 .. 86399),
225 originator-name [4] UTF8String (SIZE (1 .. 32)),
226 call-type-id [5] INTEGER (0 .. 65535),
227 voip-calling-number [6] DataVoipNumber,
228 voip-called-number [7] DataVoipNumber,
229 inbound-bunch [8] Bunch,
230 outbound-bunch [9] Bunch,
231 conference-id [10] UTF8String (SIZE (1..64)),
232 protocol [11] VoipProtocol,
233 term-cause [12] INTEGER (0 .. 16384)
234 }
235 }
236
237 requestedFileTransfer TAGGED ::= {
238   OID { sorm-request-connection-file-transfer }
239   DATA CHOICE {
240     point-id [0] INTEGER (0 .. 1000),
241     client-info [1] NetworkPeerInfo,
242     server-info [2] NetworkPeerInfo,
243     server-name [3] UTF8String (SIZE (1 .. 256)),
244     user-name [4] UTF8String (SIZE (1 .. 64)),
245     user-password [5] UTF8String (SIZE (1 .. 256)),
246     term-cause [6] INTEGER (1 .. 16384)
247   }
248 }
249
250 requestedTermAccess TAGGED ::= {
251   OID { sorm-request-connection-term-access }
252   DATA CHOICE {
253     point-id [0] INTEGER (0 .. 1000),
254     client-info [1] NetworkPeerInfo,
255     server-info [2] NetworkPeerInfo
256   }
257 }
258
259 — идентификаторы для соединений передачи данных (закрытые протоколы обмена)
260 requestedRawFlows TAGGED ::= {
261   OID { sorm-request-connection-raw-flows }
262   DATA CHOICE {
263     point-id [0] INTEGER (0 .. 1000),
264     protocol-code [1] INTEGER (0 .. 65535),

```

— длительность разговора, сек.  
— общедоступное имя инициатора связи  
— способ подключения  
— номер вызывающего абонента  
— номер вызываемого абонента  
— входящий пучок  
— исходящий пучок  
— идентификатор конференции  
— причина завершения соединения

— идентификатор точки подключения к сети передачи  
— идентификатор абонента сети передачи данных  
— идентификатор сервера сети передачи данных  
— название сервера  
— имя пользователя, наименование учетной записи  
— пользовательский пароль  
— причина завершения соединения

— идентификатор точки подключения к сети передачи  
— идентификатор абонента сети передачи данных  
— идентификатор сервера сети передачи данных

— идентификатор точки подключения к сети передачи  
— код протокола в соответствии с RFC1700

```

265 client-info [2] NetworkPeerInfo,
266 server-info [3] NetworkPeerInfo
267 }
268 }
269
270 — идентификаторы для соединений передачи данных (закрытые протоколы обмена)
271 requestedAddressTranslations TAGGED ::= {
272   OID { somr-request -connection -address -translations }
273   DATA CHOICE {
274     point-id [0] INTEGER (0 .. 1000),
275     — идентификатор точки подключения к сети передачи данных, с
276     — идентификатор точки подключения к сети передачи данных, с
277     которой получена запись о соединении
278     record-type [2] ENUMERATED {
279       session-start (0),
280       session-end (1),
281       },
282     private-ip [3] NetworkPeerInfo,
283     public-ip [4] NetworkPeerInfo,
284     destination-ip [5] NetworkPeerInfo,
285     translation-type [6] ENUMERATED {
286       static-nat (0),
287       dynamic-nat (1),
288       source-nat (2),
289       destination-nat (3),
290       pat (4)
291     }
292   }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

## RequestedIdentifiers.asn

```

1 RequestedIdentifiers DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS
5   RequestedIdentifier,
6   requestedPagerIdentifier,
7   requestedPstnIdentifier,
8   requestedGsmIdentifier,
9   requestedCdmaIdentifier
10 ;
11
12 IMPORTS TAGGED,

```

```

13      sorm-request-identifier-pager,
14      sorm-request-identifier-pstn,
15      sorm-request-identifier-gsm,
16      sorm-request-identifier-cdma,
17      sorm-request-identifier-data-network
18      FROM Classification
19
20      DataNetworkEquipment,
21      IPAddress
22      FROM NetworkIdentifiers
23      ;
24
25      RequestedIdentifier ::= SEQUENCE {
26      id TAGGED.&id ({RequestedIdentifierVariants}),
27      data TAGGED.&Data ({RequestedIdentifierVariants}){@id}
28      }
29
30
31      — варианты запрашиваемых идентификаторов
32      RequestedIdentifierVariants TAGGED ::= {
33      requestedPagerIdentifier |
34      requestedPstnIdentifier |
35      requestedGsmIdentifier |
36      requestedCdmaIdentifier |
37      requestedDataNetworkIdentifier
38      }
39
40      — идентификатор сети персонального радиовызова
41      requestedPagerIdentifier TAGGED ::= {
42      OID { sorm-request-identifier-pager }
43      DATA RequestedPagerIdentifier
44      }
45
46      RequestedPagerIdentifier ::= NumericString (SIZE (2 .. 18))
47
48      — идентификатор телефонной сети общего пользования
49      requestedPstnIdentifier TAGGED ::= {
50      OID { sorm-request-identifier-pstn }
51      DATA RequestedPstnIdentifier
52      }
53
54      RequestedPstnIdentifier ::= SEQUENCE {
55      directory-number UTF8String (SIZE (2 .. 32)),
56      internal-number NumericString (SIZE (1 .. 32)) OPTIONAL

```

— идентификатор сети персонального радиовызова  
 — идентификатор ТФОП  
 — идентификатор GSM  
 — идентификатор CDMA  
 — идентификатор сети передачи данных

— телефонный номер в международном формате  
 — дополнительный внутренний номер, если есть

```

57 }
58
59 -- идентификатор абонента GSM
60 requestedGsmIdentifier TAGGED ::= {
61   OID { sorm-request-identifier-gsm }
62   DATA RequestedGsmIdentifier
63 }
64
65 RequestedGsmIdentifier ::= CHOICE {
66   directory-number [0] UTF8String (SIZE (2 .. 32)),
67   imsi [1] NumericString (SIZE (10 .. 18)),
68   imei [2] NumericString (SIZE (10 .. 18))
69 }
70
71 -- идентификатор абонента CDMA
72 requestedCdmaIdentifier TAGGED ::= {
73   OID { sorm-request-identifier-cdma }
74   DATA RequestedCdmaIdentifier
75 }
76
77 RequestedCdmaIdentifier ::= CHOICE {
78   directory-number [0] UTF8String (SIZE (2 .. 32)),
79   imsi [1] NumericString (SIZE (10 .. 18)),
80   esn [2] NumericString (SIZE (10 .. 18)),
81   min [3] NumericString (SIZE (10 .. 18))
82 }
83
84 -- Идентификатор сети передачи данных
85 requestedDataNetworkIdentifier TAGGED ::= {
86   OID { sorm-request-identifier-data-network }
87   DATA RequestedDataNetworkIdentifier
88 }
89
90 RequestedDataNetworkIdentifier ::= CHOICE {
91   user-equipment [0] DataNetworkEquipment,
92   login [1] UTF8String (SIZE (1 .. 128)),
93   ip-address [2] IP-Address,
94   e-mail [3] UTF8String (SIZE (1 .. 128)),
95   voip-phone-number [5] UTF8String (SIZE (2 .. 32))
96 }
97
98
99 END

```

— телефонный номер в международном формате  
 — идентификатор мобильного абонента  
 — идентификатор мобильной станции

— телефонный номер в международном формате  
 — идентификатор мобильного абонента  
 — идентификатор мобильной станции  
 — идентификатор мобильного абонента (CDMA)

— идентификатор пользовательского оборудования  
 — имя пользователя — login  
 — IP адрес  
 — адрес электронной почты  
 — номер телефона

## Sessions.asn

```

1 Sessions DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS sessionMessage;
5
6 IMPORTS TAGGED
7     , sorm-message-session
8     FROM Classification;
9
10 sessionMessage TAGGED ::= {
11     OID { sorm-message-session }
12     DATA CHOICE {
13         connect [0]
14         connect-response [1]
15
16         adjustment [2]
17         adjustment-response [3]
18
19         disconnect [4]
20         disconnect-response [5]
21     }
22 }
23
24 --- запрос создания сессии
25 ConnectRequest ::= SEQUENCE {
26     session-timeout
27     max-data-length
28     data-packet-window-size
29
30     data-load-timeout
31     request-response-timeout
32     data-packet-response-timeout
33     data-packet-response-timeout
34 }
35
36 --- ответ на запрос создания сессии
37 ConnectResponse ::= SEQUENCE {
38     confirmed-data-packet-window-size INTEGER (4 .. 256),
39
40     confirmed-session-timeout
41     confirmed-session-timeout
42     confirmed-session-timeout
43     confirmed-session-timeout
44     confirmed-session-timeout
45     confirmed-session-timeout
46     confirmed-session-timeout
47     confirmed-session-timeout
48     confirmed-session-timeout
49     confirmed-session-timeout
50     confirmed-session-timeout
51     confirmed-session-timeout
52     confirmed-session-timeout
53     confirmed-session-timeout
54     confirmed-session-timeout
55     confirmed-session-timeout
56     confirmed-session-timeout
57     confirmed-session-timeout
58     confirmed-session-timeout
59     confirmed-session-timeout
60     confirmed-session-timeout
61     confirmed-session-timeout
62     confirmed-session-timeout
63     confirmed-session-timeout
64     confirmed-session-timeout
65     confirmed-session-timeout
66     confirmed-session-timeout
67     confirmed-session-timeout
68     confirmed-session-timeout
69     confirmed-session-timeout
70     confirmed-session-timeout
71     confirmed-session-timeout
72     confirmed-session-timeout
73     confirmed-session-timeout
74     confirmed-session-timeout
75     confirmed-session-timeout
76     confirmed-session-timeout
77     confirmed-session-timeout
78     confirmed-session-timeout
79     confirmed-session-timeout
80     confirmed-session-timeout
81     confirmed-session-timeout
82     confirmed-session-timeout
83     confirmed-session-timeout
84     confirmed-session-timeout
85     confirmed-session-timeout
86     confirmed-session-timeout
87     confirmed-session-timeout
88     confirmed-session-timeout
89     confirmed-session-timeout
90     confirmed-session-timeout
91     confirmed-session-timeout
92     confirmed-session-timeout
93     confirmed-session-timeout
94     confirmed-session-timeout
95     confirmed-session-timeout
96     confirmed-session-timeout
97     confirmed-session-timeout
98     confirmed-session-timeout
99     confirmed-session-timeout
100    confirmed-session-timeout

```

--- запрос на открытие сессии  
 --- ответ на запрос открытия сессии

--- согласование поддерживаемых типов со стороны ПУ  
 --- ответ на запрос согласования данных

--- запрос на закрытие сессии  
 --- ответ на запрос закрытия сессии

--- максимальное время неактивности  
 --- максимальная длина блока отчета (в строках)  
 --- окно канала передачи данных  
 --- максимальное число блоков данных, которое может быть  
 --- отправлено без подтверждения приема  
 --- таймаут начала передачи блоков отчетов  
 --- таймаут ответа на запрос  
 --- таймаут подтверждения приема блока данных отчета

--- подтвержденное окно передачи данных  
 --- то окно, которое может обеспечить ИС СОФМ  
 --- ConnectRequest  
 --- подтвержденное максимальное время неактивности  
 --- должно быть больше или равно значению, переданному в

```

43 confirmed-data-load-timeout          INTEGER (1 .. 60),
44                                     ConnectRequest
    --- подтвержденный таймаут начала передачи блоков отчетов
    --- должен быть больше или равен значению, переданному в
    ConnectRequest
45 confirmed-request-response-timeout  INTEGER (1 .. 60),
46                                     ConnectRequest
    --- подтвержденный таймаут ответа на запрос
    --- должен быть больше или равен значению, переданному в
    ConnectRequest
47 supports SEQUENCE OF ObjectDescriptor
    отчетов
48 }
49 --- согласование поддерживаемых типов со стороны ПУ
50 AdjustmentRequest ::= SEQUENCE {
51 supports SEQUENCE OF ObjectDescriptor
    отчетов
52 }
53
54 }
55
56 --- ответ на согласование списка поддерживаемых типов
57 AdjustmentResponse ::= NULL
58
59
60 --- запрос завершения сессии
61 DisconnectRequest ::= NULL
62
63 --- ответ на запрос завершения сессии
64 DisconnectResponse ::= NULL
65
66
67 END

```

---

### Sorm.asn

```

1 Sorm DEFINITIONS IMPLICIT TAGS ::=
2 BEGIN
3
4 EXPORTS DateAndTime,
5 FindRange,
6 MessageID,
7 Message;
8
9 IMPORTS TAGGED FROM Classification
10 sessionMessage FROM Sessions

```



```

11 trapMessage FROM Traps
12 taskMessage FROM Tasks
13 reportMessage FROM Reports
14 managementMessage FROM Management
15 unformattedMessage FROM Unformatted;
16
17
18 Version ::= PrintableString
19 vers Version ::= "2.0.0.0"
20
21 --- Оболочка сообщения СОФМ ---
22 Message ::= SEQUENCE {
23   version Version DEFAULT vers,
24   message-id MessageID,
25   message-time DateAndTime,
26   operator-name PrintableString (SIZE (6 .. 128)) OPTIONAL,
27
28   id TAGGED.&id ({SormPDUs}),
29   data TAGGED.&Data({SormPDUs}@id)
30 }
31
32 --- Блок данных сообщения
33 SormPDUs TAGGED ::= {
34   sessionMessage
35   | trapMessage
36   | taskMessage
37   | reportMessage
38   | managementMessage
39   | unformattedMessage
40 }
41
42 --- Общие данные
43
44 --- Номер сообщения
45 MessageID ::= INTEGER (0 .. 4294967295)
46
47 --- Дата и время
48 DateAndTime ::= UTCTime
49
50 --- Диапазон поиска
51 FindRange ::= SEQUENCE {
52   begin-find [0] DateAndTime OPTIONAL,
53   end-find [1] DateAndTime OPTIONAL
54 }

```

--- текущая версия протокола  
--- версия протокола  
--- номер запроса  
--- время и дата запроса  
--- наименование оператора связи  
--- идентификатор блока данных  
--- данные блока данных  
--- сообщения организации сессии  
--- сообщения сигналов  
--- сообщения работы с задачами  
--- сообщения работы с отчетами  
--- сообщения канала передачи мониторинга (КПМ)  
--- сообщения канала передачи неформатированных данных (КПФФ)  
--- время и дата начала поиска информации  
--- время и дата окончания поиска информации

55  
56

END

---

## TasksAbonents.asn

```
1  TasksAbonents DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS AbonentsTask;
5
6  IMPORTS
7  LogicalOperation FROM Tasks
8  RequestedIdentifier FROM RequestedIdentifiers
9  RequestedAbonent FROM RequestedAbonents;
10
11  AbonentsTask ::= CHOICE {
12  validate-abonents-task [0]      ValidateAbonentsTask,
13
14
15  validate-identifiers [1]        ValidateIdentifiersTask,
16
17
18  validate-services [2]          ValidateServicesTask
19
20  }
21
22  — задача на поиск информации о принадлежности идентификаторов абонентов сети оператора связи
23  ValidateAbonentsTask ::= RequestedIdentifiers
24  — запрашиваемые идентификаторы
25  RequestedIdentifiers ::= SEQUENCE OF RequestedIdentifierParameters
26  — последовательность из идентификаторов и логических
27  операций
28  RequestedIdentifierParameters ::= CHOICE {
29  separator [0] LogicalOperation,
30  find-mask [1] RequestedIdentifier
31  }
32
33  — задача на поиск информации об идентификаторах абонентов сети оператора связи зарегистрированных на физическое или юридическое
34  лицо
35  ValidateIdentifiersTask ::= RequestedAbonents
36  — запрашиваемые абоненты
37  RequestedAbonents ::= SEQUENCE OF RequestedAbonentsParameters
38  — последовательность логических операций и параметров
```

```

37 RequestedAbonentsParameters ::= CHOICE {
38   separator [0] LogicalOperation ,
39   find-mask [1] RequestedAbonent
40 }
41
42
43
44 --- задача на поиск истории услуг связи абонента
45 --- запрос по маске запрещен, идентификатор должен быть указан полностью
46 ValidateServicesTask ::= SEQUENCE OF ValidateServicesParameters
47 ValidateServicesParameters ::= CHOICE {
48   separator [0] LogicalOperation ,
49   find-mask [1] ValidateServicesParameter
50 }
51
52 ValidateServicesParameter ::= CHOICE {
53   contract [0] UTF8String (SIZE (1 .. 64)) ,
54   identifier [1] RequestedIdentifier
55 }
56
57
58 END

```

--- логический оператор связи  
 --- информация запроса об абоненте

--- задача на поиск истории услуг связи абонента  
 --- запрос по маске запрещен, идентификатор должен быть указан полностью

--- логическая операция или скобка  
 --- параметр — идентификатор

--- номер договора

## Tasks.asn

```

1  Tasks DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS taskMessage ,
5         TaskID ,
6         LogicalOperation ,
7         CreateTaskResponse ;
8
9  IMPORTS TAGGED,
10         sorm-message-task
11         FROM Classification
12
13         FindRange ,
14         MessageID
15         FROM Sorm
16
17         TelcoList ,
18         DictionaryTask
19         FROM Dictionaries

```

```

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
}

AbonentsTask
FROM TasksAbonents

ConnectionsTask
FROM TasksConnections

LocationTask
FROM TasksLocation

PaymentsTask
FROM TasksPayments

PresenseTask
FROM TasksPresense

CombinationTask
FROM TasksCombination

NonFormalizedTaskRequest,
NonFormalizedTaskResponse
FROM TasksNonFormalized;

taskMessage TAGGED ::= {
  OID {sorm-message-task}
  DATA CHOICE {
    data-ready-request [0]
    data-ready-response [1]
    data-load-request [2]
    data-load-response [3]
    data-drop-request [4]
    data-drop-response [5]
    data-interrupt-request [6]
    data-interrupt-response [7]
    create-task-request [8]
    информации
    create-task-response [9]
    non-formalized-task-request [10] NonFormalizedTaskRequest,
    неформализованных данных
    non-formalized-task-response [11] NonFormalizedTaskResponse
    неформализованных данных
  }
}

--- запрос готовности данных
--- ответ на запрос готовности данных
--- запрос загрузки данных
--- ответ на запрос загрузки данных
--- запрос удаления данных
--- ответ на запрос удаления данных
--- запрос прерывания загрузки данных
--- ответ на запрос прерывания загрузки данных
--- запрос на создание задачи по обработке
--- ответ на запрос создания задачи по обработке
--- запрос на создание задачи по обработке
--- ответ на запрос создания задачи по обработке

```

```

61 }
62
63 --- в этом запросе не параметров
64 DataReadyRequest ::= NULL
65
66 --- запрос загрузки данных конкретной задачи
67 DataLoadRequest ::= TaskID
68
69 --- запрос удаления данных конкретной задачи
70 DataDropRequest ::= TaskID
71
72 --- запрос прерывания загрузки данных
73 DataInterruptRequest ::= TaskID
74
75 --- запрос на создание задачи поиска
76 CreateTaskRequest ::= SEQUENCE {
77     telcos [0] TelcoList OPTIONAL,
78     range [1] FindRange OPTIONAL,
79     report-limit [2] INTEGER (1 .. 10000000) OPTIONAL,
80     task [3] CHOICE {
81         dictionary [0] DictionaryTask,
82         abonents [1] AbonentsTask,
83         connections [2] ConnectionsTask,
84         location [3] LocationTask,
85         payments [4] PaymentsTask,
86         combination [5] CombinationTask,
87         presense [6] PresenseTask
88     }
89 }
90
91
92
93 --- последовательность записей о готовности данных задач
94 DataReadyResponse ::= SEQUENCE OF DataReadyTaskRecord
95
96 DataReadyTaskRecord ::= SEQUENCE {
97     task-id TaskID,
98     result TaskResult
99 }
100
101 TaskResult ::= SEQUENCE {
102     result ENUMERATED {

```

--- список операторов связи  
 --- временной диапазон поиска  
 --- ограничение на максимальное количество возвращаемых  
 --- задачи пополнения справочников (нормативно-справочная информация)  
 --- задачи поисков по принадлежности абонентов  
 --- задачи поисков по соединениям абонентов  
 --- задача получения данных местоположения абонентов  
 --- задачи поисков по совершенным платежам  
 --- задачи поисков по усечению отобранных данных  
 --- задачи предоставления сведений о наличии данных

--- идентификатор задачи  
 --- результат выполнения задачи

```

103 data-not-ready (0),
104 data-ready (1),
105 data-not-found (2),
106 error (3)
107 },
108 report-records-number [0] INTEGER (0 .. 999999999999) OPTIONAL,
109 report-limit-exceeded [1] BOOLEAN OPTIONAL,
110 error-description [2] UTF8String (SIZE (1 .. 256)) OPTIONAL
111 }
112
113
114
115 DataLoadResponse ::= SEQUENCE {
116     task-id TaskID,
117     data-exists BOOLEAN,
118
119     data-blocks-number INTEGER (0 .. 999999999999) OPTIONAL,
120     error-description UTF8String (SIZE (1 .. 256)) OPTIONAL
121 }
122
123 DataDropResponse ::= SEQUENCE {
124     task-id TaskID,
125     successful BOOLEAN,
126     error-description UTF8String (SIZE (1 .. 256)) OPTIONAL
127 }
128
129 DataInterruptResponse ::= SEQUENCE {
130     request-id MessageID,
131     successful BOOLEAN,
132     data-blocks-available INTEGER (0 .. 999999999999) OPTIONAL,
133     error-description UTF8String (SIZE (1 .. 256)) OPTIONAL
134 }
135
136 CreateTaskResponse ::= SEQUENCE {
137     task-id TaskID OPTIONAL,
138     successful BOOLEAN,
139     error-description UTF8String (SIZE (1 .. 256)) OPTIONAL
140 }
141
142
143 --- идентификатор задачи
144 TaskID ::= INTEGER (0 .. 4294967295)
145

```

— данные не готовы, задача еще выполняется  
— данные есть, задача выполнена  
— данных нет, задача выполнена  
— в процессе выполнения задачи произошла ошибка

— для выполненной задачи — количество записей в отчете  
— количество записей превысило лимит, заданный при создании задачи  
— краткое описание произошедшей ошибки, если обнаружена

— идентификатор задачи, сгенерировавшей данный отчет  
— признак существования результатов исполнения задачи (есть данные или нет)  
— количество блоков в отчете  
— краткое описание ошибки, если обнаружена

— идентификатор задачи, данные которой будут удалены  
— признак успешного выполнения запроса  
— краткое описание ошибки, если обнаружена

— идентификатор прерванного запроса загрузки данных  
— признак успешного выполнения запроса  
— количество оставшихся переданными блоками  
— краткое описание ошибки, если обнаружена

--- идентификатор задачи  
— признак успешного выполнения запроса  
— краткое описание ошибки, если обнаружена

```

146 LogicalOperation ::= ENUMERATED {
147   operation-open-bracket (0), — открывающая скобка — "("
148   operation-close-bracket (1), — закрывающая скобка — ")"
149   operation-or (2), — логическое "или"
150   operation-and (3), — логическое "и"
151   operation-not (4) — логическое "не"
152 }
153
154
155 END

```

## TasksCombination.asn

```

1   TasksCombination DEFINITIONS IMPLICIT TAGS ::=
2   BEGIN
3
4   EXPORTS CombinationTask;
5
6   IMPORTS
7     LogicalOperation, TaskID FROM Tasks
8     RequestedConnection FROM RequestedConnections
9
10    RequestedIdentifier FROM RequestedIdentifiers
11    IPAddress FROM NetworkIdentifiers
12    FindRange FROM Sorm;
13
14  — задачи поисков по усечению ообранных данных
15  CombinationTask ::= ValidateCombinationFilter
16
17  — задача на усечение выборки по критериям
18  ValidateCombinationFilter ::= SEQUENCE {
19    set-a TaskID,
20    set-b TaskID,
21
22    filter-set-action ENUMERATED {
23      intersect (0),
24      combine (1),
25      difference (2)
26    },
27    criteria-a [0] CombinationCriteria OPTIONAL,
28    criteria-b [1] CombinationCriteria OPTIONAL
29  }

```

— задача на усечение выборки по критериям  
 — результат выполненной ранее задачи для обработки  
 — результат выполненной ранее задачи для обработки  
 — действия с результатами выполненных ранее задач  
 — пересечение результатов  
 — объединение результатов  
 — вычитание результатов  
 — значение критерия для выполнения действия с  
 — значение критерия для выполнения действия с

30 CombinationCriteria ::= RequestedIdentifier  
31  
32  
33 **END**

### TasksConnections.asn

```
1  TasksConnections DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS ConnectionsTask ;
5
6  IMPORTS
7    LogicalOperation FROM Tasks
8    RequestedConnection FROM RequestedConnections
9    ;
10
11 ConnectionsTask ::= CHOICE {
12   validate-connections [0] ValidateConnectionsTask ,
13   validate-data [1]    ValidateDataTask ,
14   validate-entrance [2] ValidateEntranceTask
15 }
16
17 -- Зачем разделение ValidateConnectionsTask ValidateDataTask
18
19 --- задача на поиск по соединениям абонентов
20 ValidateConnectionsTask ::= RequestedConnectionIdentifiers
   RequestedDataNetworkIdentifier
21 ValidateDataTask ::= RequestedConnectionIdentifiers
   RequestedDataNetworkIdentifier
22 ValidateEntranceTask ::= RequestedConnectionIdentifiers
   сети связь (выхода на связь)
23
24
25 RequestedConnectionIdentifiers ::= SEQUENCE OF RequestedConnectionParameter
26 RequestedConnectionParameter ::= CHOICE {
27   separator [0] LogicalOperation ,
28   find-mask [1] RequestedConnection
29 }
30
31
32 END
```

--- задача на поиск телефонных соединений между абонентами

--- задача на поиск соединений между абонентами сети

--- задача на поиск информации о появлении абонента в сети связь (выхода на связь)

--- запрашиваемые идентификаторы, указываются все, кроме

--- запрашиваемые идентификаторы, указываются

--- задача на поиск информации о появлении абонента в сети связь (выхода на связь)

--- логический оператор связи

--- параметр запроса



## TasksLocation.asn

```

1  TasksLocation  DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS
5  LocationTask;
6
7  IMPORTS
8  LogicalOperation FROM Tasks
9  RequestedConnection FROM RequestedConnections
10 IPAddress
11 FROM NetworkIdentifiers;
12
13 --- задача получения данных местоположения абонентов
14 LocationTask ::= RequestedLocationIdentifier --- запрашиваемые единтгификаторы для определения местоположения
15
16 RequestedLocationIdentifier ::=
17 CHOICE {
18   directory-number [0]  UTF8String (SIZE (2 .. 32)), --- телефонный номер в международном формате
19   imsi [1]              NumericString (SIZE (10 .. 18)), --- идентификатор мобильного абонента
20   ip-address [2]       IPAddress --- IP-адрес абонента
21 }
22
23 END

```

## TasksNonFormalized.asn

```

1  TasksNonFormalized  DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS
5  EntityId,
6  NonFormalizedTaskRequest,
7  NonFormalizedTaskResponse,
8  NonFormalizedEntityAttributeData;
9
10 IMPORTS
11 FROM TelcoList
12 FROM Dictionaries
13 FindRange,
14 MessageID,
15 DateAndTime

```

```

16 FROM Sorm
17
18 Location
19 FROM Locations
20
21 LogicalOperation ,
22 CreateTaskResponse
23 FROM Tasks
24 ;
25
26 --- TaskID, --- LogicalOperation ;
27
28
29 NonFormalizedTaskRequest ::= CHOICE {
30   get-entities [0] GetEntities,
31   get-attributes [1] GetEntityAttributes,
32   validate-task [2] ValidateNonFormalizedTask,
33   validate-presense [3] NonFormalizedPresenseTask
34   неформализованных данных"
35 }
36
37 NonFormalizedTaskResponse ::= CHOICE {
38   entities [0] GetEntitiesResponse,
39   entity-attributes [1] GetEntityAttributesResponse,
40   validate-task [2] ValidateNonFormalizedTaskResponse,
41   validate-presense [3] NonFormalizedPresenseTaskResponse
42   неформализованных данных
43 }
44
45
46 --- тип сообщения "запрос получения списка типов сущностей"
47 GetEntities ::= NULL
48
49 --- тип сообщения "запрос получения списка атрибутов сущности"
50 GetEntityAttributes ::= EntityId
51
52 --- тип сообщения "задача поиск неформализованных данных"
53 ValidateNonFormalizedTask ::= SEQUENCE {
54   entity-id EntityId,
55   parameters NonFormalizedParameters,
56   range FindRange OPTIONAL,
57   report-limit INTEGER (1 .. 10000000) OPTIONAL

```

--- тип сообщения "запрос получения списка типов сущностей"  
--- тип сообщения "запрос получения списка атрибутов сущности"  
--- тип сообщения "задача поиска неформализованных данных"  
--- тип сообщения "задача предоставления сведений о наличии неформализованных данных"  
--- тип сообщения "запрос получения списка типов сущностей"  
--- ответ на запрос получения списка типов сущностей  
--- ответ на запрос получения списка атрибутов сущности  
--- ответ на запрос задачи поиска неформализованных данных  
--- ответ на запрос задач предоставления сведений о наличии неформализованных данных  
--- сущность для поиска по неформализованному данным  
--- критерии поиска по неформализованному данным  
--- временной диапазон поиска  
--- ограничение на максимальное количество возвращаемых записей

```

58 }
59
60 --- тип сообщения "задача предоставления сведений о наличии неформализованных данных"
61 NonFormalizedPresenseTask ::= EntityId
62
63 NonFormalizedParameters ::= SEQUENCE OF NonFormalizedParameter
64
65 NonFormalizedParameter ::= CHOICE {
66 separator [0] LogicalOperation,
67 find-mask [1] NonFormalizedEntityCondition
68 }
69
70 NonFormalizedEntityCondition ::= SEQUENCE {
71 attribute NonFormalizedEntityAttribute,
72 operation MathOperation,
73 attribute-value NonFormalizedEntityAttributeData
74 }
75
76 NonFormalizedEntityAttribute ::= SEQUENCE {
77 attribute-name UTF8String (SIZE (1 .. 256)),
78 attribute-type AttributeType
79 }
80
81 NonFormalizedEntityAttributeData ::= CHOICE {
82 datetime [0] DateAndTime,
83 integer [1] INTEGER,
84 string [2] UTF8String,
85 boolean [3] BOOLEAN,
86 float [4] REAL,
87 location [5] Location,
88 empty [6] NULL
89 }
90
91 --- математические операции сравнения
92 MathOperation ::= ENUMERATED {
93 equal (0), --- равно
94 less (1), --- меньше
95 greater (2), --- больше
96 not-equal (3), --- не равно
97 less-or-equal (4), --- меньше или равно
98 greater-or-equal (5) --- больше или равно
99 }
100
101

```

--- логическая операция  
--- условие  
--- атрибут сущности  
--- операция  
--- значение атрибута  
--- текстовое наименование атрибута сущности  
--- тип данных атрибута  
--- дата и время с точностью до секунд  
--- целочисленный  
--- строковый  
--- булевый  
--- с плавающей запятой  
--- местоположение  
--- пустое значение ( null)

---

```

102 GetEntitiesResponse ::= SEQUENCE OF NonFormalizedEntity
103
104 NonFormalizedEntity ::= SEQUENCE {
105   entity-id EntityId,
106   entity-name UTF8String (SIZE (1 .. 256))
107 }
108
109 GetEntityAttributesResponse ::= SEQUENCE {
110   entity-id EntityId,
111   entity-attributes SEQUENCE OF NonFormalizedEntityAttribute
112 }
113
114 ValidateNonFormalizedTaskResponse ::= CreateTaskResponse
115
116 NonFormalizedPresenceTaskResponse ::= CreateTaskResponse
117
118
119
120
121 --- типы данных атрибутов
122 AttributeType ::= ENUMERATED {
123   date-time (0), --- дата и время с точностью до секунд
124   integer (1), --- целочисленный
125   string (2), --- строковый
126   boolean (3), --- булевый
127   float (4), --- с плавающей запятой
128   location (5), --- местоположение
129   empty (6) --- пустое значение
130 }
131
132 --- Идентификатор сущности
133 EntityId ::= INTEGER (0 .. 4294967296)
134
135 END

```

---

## TasksPayments.asn

```

1  TasksPayments DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS PaymentsTask;
5
6  IMPORTS LogicalOperation
7  FROM Tasks

```

```

8
9 RequestedConnection
10 FROM RequestedConnections
11
12 RequestedIdentifier
13 FROM RequestedIdentifiers
14
15 RequestedAddress
16 FROM Addresses
17
18 TAGGED,
19 sorm-request-payment-bank-transaction ,
20 sorm-request-payment-express-pays ,
21 sorm-request-payment-terminal-pays ,
22 sorm-request-payment-service-center ,
23 sorm-request-payment-cross-account ,
24 sorm-request-payment-telephone-card ,
25 sorm-request-payment-balance-fillups ,
26 sorm-request-payment-bank-division-transfer ,
27 sorm-request-payment-bank-card-transfer ,
28 sorm-request-payment-bank-account-transfer
29 FROM Classification ;
30
31
32
33 PaymentsTask ::= SEQUENCE {
34   id TAGGED.&id ({RequestedPaymentsVariants}),
35   data TAGGED.&Data ({RequestedPaymentsVariants}){@id}
36 }
37
38 — варианты запрашиваемых параметров связей
39 RequestedPaymentsVariants TAGGED ::= {
40   bankTransactionTask
41   | expressCardTask
42   | publicTerminalTask
43   | serviceCenterTask
44   | crossAccountTask
45   | telephoneCardTask
46   | balanceFillupTask
47   | bankDivisionTransferTask
48   | bankCardTransferTask
49   | bankAccountTransferTask
50 }
51

```

```

52  — задача на поиск пополнения баланса через банковский перевод
53  bankTransactionTask TAGGED ::= {
54  OID { sorm-request-payment-bank-transaction }
55  DATA RequestedBankTransactionPays
56  }
57
58  RequestedBankTransactionPays ::= SEQUENCE OF RequestedBankTransactionPaysParameters
59  — последовательность логических операций и параметров
60  RequestedBankTransactionPaysParameters ::= CHOICE {
61  separator [0] LogicalOperation,
62  bank-account [1] UTF8String (SIZE (1 .. 64)),
63  bank-name [2] UTF8String (SIZE (1 .. 512))
64  перевод
65  }
66
67  — задача на поиск пополнения баланса через карты экспресс-оплаты
68  expressCardTask TAGGED ::= {
69  OID { sorm-request-payment-express-pays }
70  DATA RequestedExpressPays
71  }
72
73  RequestedExpressPays ::= SEQUENCE OF RequestedExpressPaysParameters
74  RequestedExpressPaysParameters ::= CHOICE {
75  separator [0] LogicalOperation,
76  express-card [1] NumericString (SIZE (2 .. 20))
77  }
78
79
80
81  — задача на поиск пополнения баланса через терминалы моментальных платежей
82  publicTerminalTask TAGGED ::= {
83  OID { sorm-request-payment-terminal-pays }
84  DATA RequestedTerminalPays
85  }
86
87  RequestedTerminalPays ::= SEQUENCE OF RequestedTerminalPaysParameters
88  RequestedTerminalPaysParameters ::= CHOICE {
89  separator [0] LogicalOperation,
90  terminal-id [1] UTF8String (SIZE (1 .. 64)),
91  terminal-number [2] NumericString (SIZE (2 .. 20)),
92  terminal-address [3] RequestedAddress
93  }
94

```

— логический оператор связи  
— номер банковского счета, с которого совершен платеж  
— наименование банка, со счета которого совершен перевод  
— последовательность логических операций и параметров  
— логический оператор связи  
— номер банковской карты экспресс-оплаты  
— последовательность логических операций и параметров  
— логический оператор связи  
— идентификатор терминала  
— номер терминала  
— адрес терминала

95 — задача на поиск пополнения баланса через центры обслуживания клиентов (ЦОК)  
96 serviceCenterTask TAGGED ::= {  
97   OID { sorm-request-payment-service-center }  
98   DATA RequestedServiceCenterPays  
99 }  
100 }  
101 }  
102 RequestedServiceCenterPays ::= **SEQUENCE OF** RequestedServiceCenterPaysParameters — последовательность логических операций и параметров  
103 RequestedServiceCenterPaysParameters ::= **CHOICE** {  
104   separator [0] LogicalOperation, — логический оператор связи  
105   center-id [1] **UTF8String** (SIZE (1 .. 64)), — идентификатор центра обслуживания клиентов  
106   center-address [2] RequestedAddress — адрес центра обслуживания клиентов  
107 }  
108 }  
109 }  
110 — задача на поиск пополнения баланса посредством снятия денег со счета другого абонента  
111 crossAccountTask TAGGED ::= {  
112   OID { sorm-request-payment-cross-account }  
113   DATA RequestedCrossAccountPays  
114 }  
115 }  
116 RequestedCrossAccountPays ::= **SEQUENCE OF** RequestedCrossAccountPaysParameters — последовательность логических операций и параметров  
117 RequestedCrossAccountPaysParameters ::= **CHOICE** {  
118   separator [0] LogicalOperation, — логический оператор связи  
119   source-identifier [1] RequestedIdentifier, — идентификатор абонента, со счета которого переводятся средства  
120   dest-identifier [2] RequestedIdentifier — идентификатор абонента, на счет которого переводятся средства  
121 }  
122 }  
123 }  
124 — задача на поиск пополнения баланса через телефонные карты  
125 telephoneCardTask TAGGED ::= {  
126   OID { sorm-request-payment-telephone-card }  
127   DATA RequestedTelephoneCardPays  
128 }  
129 }  
130 RequestedTelephoneCardPays ::= **SEQUENCE OF** RequestedTelephoneCardPaysParameters — последовательность логических операций и параметров  
131 RequestedTelephoneCardPaysParameters ::= **CHOICE** {  
132   separator [0] LogicalOperation, — логический оператор связи  
133   card-number [1] **NumericString** (SIZE (2 .. 20)) — номер телефонной карты

134 }  
135 }  
136 }  
137 ———— общая задача на поиск пополнения баланса личного счета абонента  
138 balanceFillupTask TAGGED ::= {  
139   OID { sorm-request-payment-balance-fillups }  
140   DATA RequestedBalanceFillups  
141 }  
142 }  
143 RequestedBalanceFillups ::= **SEQUENCE OF** RequestedBalanceFillupsParameters ———— последовательность логических операций и параметров  
144 RequestedBalanceFillupsParameters ::= **CHOICE** {  
145   separator [0] LogicalOperation ,  
146   identifier [1] RequestedIdentifier  
147 }  
148 }  
149 }  
150 ———— задача на поиск перевода средств со счета абонента для их снятия в отделении банка  
151 bankDivisionTransferTask TAGGED ::= {  
152   OID { sorm-request-payment-bank-division-transfer }  
153   DATA RequestedBankDivisionTransferPays  
154 }  
155 }  
156 RequestedBankDivisionTransferPays ::= **SEQUENCE OF** RequestedTransferParameters ———— последовательность логических операций и параметров  
157 }  
158 }  
159 ———— задача на поиск перевода средств со счета абонента на банковскую карту  
160 bankCardTransferTask TAGGED ::= {  
161   OID { sorm-request-payment-bank-card-transfer }  
162   DATA RequestedBankCardTransferPays  
163 }  
164 }  
165 RequestedBankCardTransferPays ::= **SEQUENCE OF** RequestedTransferParameters ———— последовательность логических операций и параметров  
166 }  
167 }  
168 ———— задача на поиск перевода средств со счета абонента на счет в банке  
169 bankAccountTransferTask TAGGED ::= {  
170   OID { sorm-request-payment-bank-account-transfer }  
171   DATA RequestedBankAccountTransferPays  
172 }  
173 }  
174 RequestedBankAccountTransferPays ::= **SEQUENCE OF** RequestedTransferParameters ———— последовательность логических операций и параметров  
175 }



```

176 RequestedTransferParameters ::= CHOICE {
177   separator [0] LogicalOperation,
178   source-identifier [1] RequestedIdentifier
179 }
180
181
182
183 END

```

— логический оператор связи  
 — идентификатор абонента инициатора перевода средств

### TasksPresence.asn

```

1  TasksPresence DEFINITIONS IMPLICIT TAGS ::=
2  BEGIN
3
4  EXPORTS PresenceTask;
5
6  IMPORTS
7    TAGGED,
8    sorm-request-presense
9    FROM Classification;
10
11 PresenceTask ::= SEQUENCE {
12   id TAGGED.&id ({PresenceListVariants}),
13   data TAGGED.&Data ({PresenceListVariants}){@id}}
14 }
15
16 PresenceListVariants TAGGED ::= { presenseInfo }
17
18 presenseInfo TAGGED ::= {
19   OID { sorm-request-presense }
20   DATA ENUMERATED {
21     subscribers (0),
22     connections (1),
23     payments (2),
24     dictionaries (3),
25     locations (4)
26   }
27 }
28
29 END

```

— запрос наличия информации по абонентам и их идентификаторам  
 — запрос наличия информации по соединениям  
 — запрос наличия имеющейся информации по платежам  
 — запрос наличия справочников  
 — запрос наличия информации по местоположениям абонентов

### Traps.asn

```

1  Traps DEFINITIONS IMPLICIT TAGS ::=

```

```

2 BEGIN
3
4 EXPORTS trapMessage ;
5
6 IMPORTS
7   TAGGED,
8   sorm-message-trap
9   FROM Classification
10
11   MessageID
12   FROM Sorm
13   ;
14
15 trapMessage TAGGED ::= {
16   OID { sorm-message-trap }
17   DATA CHOICE {
18     trap [0] Trap,
19     trap-ack [1] TrapAck
20   }
21 }
22
23
24 -- Блок данных сообщения типа "сигнал"
25 Trap ::= SEQUENCE {
26   trap-type TrapType,
27   trap-message UTF8String (SIZE (1 .. 256)) OPTIONAL,
28   reference-message MessageID OPTIONAL
29 }
30
31
32 TrapType ::= ENUMERATED {
33   heartbeat (0),
34   restart-software (1),
35   unauthorized-access (2),
36   critical-error (3),
37   работа невозможна
38   major-error (4),
39   работа возможна
40   minor-error (5)
41   дальнейшая работа возможна
42 }
43
44 -- Блок данных сообщения типа "подтверждение сигнала"

```

--- тип сообщения "сигнал"  
--- тип сообщения "подтверждение сигнала"

--- тип сообщения  
--- описание сообщения  
--- номер сообщения к которому относится данный сигнал  
--- (например номер сообщения запросившего отчет при прерывании передачи)

--- тестовый пакет  
--- перезапуск ПО  
--- попытка несанкционированного доступа  
--- критическая ошибка ПО, потеря данных, дальнейшая работа невозможна  
--- серьезная ошибка ПО, потеря данных, но дальнейшая работа возможна  
--- незначительная ошибка ПО, данные не потеряны, дальнейшая работа возможна

42 -- Номер сообщения TrarAck должен соответствовать номеру сообщения Trar  
 43 TrarAck ::= **NULL**  
 44  
 45 **END**

---

Unformatted.asn

```

1 Unformatted DEFINITIONS IMPLICIT TAGS ::=
2
3 BEGIN
4
5 EXPORTS unformattedMessage;
6
7 IMPORTS TAGGED,
8   sorm-message-unformatted
9   FROM Classification
10
11   TelcoList
12   FROM Dictionaries
13
14   Acknowledgement
15   FROM Reports
16
17   CallsRecords
18   FROM ReportsConnections
19
20   DateAndTime,
21   MessageID
22   FROM Sorm;
23
24 unformattedMessage TAGGED ::= {
25   OID { sorm-message-unformatted }
26   DATA CHOICE {
27     request [0] RawRequest,
28     response [1] RawResponse,
29     report [2] RawReport,
30     report-ack [3] RawAcknowledgement
31   }
32 }
33
34 RawRequest ::= SEQUENCE {
35   telcos TelcoList, --- список операторов связи
36   raw-task RawRequestTask --- запрос получения неформатированных данных
37 }

```

```

38
39
40 RawRequestTask ::= CHOICE {
41   data-types-request [0]      DataTypesRequest,
42   data-start-request [1]     DataStartRequest,
43   data-stop-request [2]      DataStopRequest
44 }
45
46
47 --- типы данных, передаваемых ИС СОФМ
48 RawDataType ::= ENUMERATED {
49   data-reports (0), --- записи о соединениях в формате сообщений кплд2
50   raw-cdr (1), --- "сырые" CDR-файлы
51   raw-ip (2) --- записанные сессии сетевого трафика в TCPDUMP-формате
52 }
53
54 ControlCommandType ::= ENUMERATED {
55   start (0), --- команда начала передачи ИС СОФМ данных
56   stop (1) --- команда прерывания передачи ИС СОФМ данных
57 }
58
59
60 DataTypesRequest ::= RawDataType
61
62 DataStartRequest ::= SEQUENCE {
63   time-from DateAndTime, --- начало временного периода в буфере, с которого необходимо получить данные
64   time-to DateAndTime, --- конец временного периода в буфере, с которого необходимо получить данные
65   raw-type RawDataType --- тип неформатированных данных передачи
66 }
67
68 DataStopRequest ::= NULL
69
70
71 RawResponse ::= CHOICE {
72   data-types-response [0] DataTypesResponse, --- ответ на запрос проверки наличия неформатированных вида данных в ИС
   СОФМ
73   data-start-response [1] DataStartResponse, --- ответ на запрос начала передачи неформатированных данных
74   data-stop-response [2] DataStopResponse --- ответ на запрос остановки передачи неформатированных данных
75 }
76
77
78 DataTypesResponse ::= SEQUENCE {
79   successful BOOLEAN, --- признак наличия в ИС СОФМ запрошенного вида неформатированных данных
80   selected-type RawDataType, --- выбранный вид данных для передачи

```

```

81 time-from DateAndTime,  --- начало временного периода в буфере, начиная с которого накоплены данные
82 time-to   DateAndTime  --- конец временного периода в буфере, по которому накоплены данные
83 }
84
85 DataStartResponse ::= BOOLEAN  --- признак успешности выполнения команды
86 DataStopResponse  ::= BOOLEAN  --- признак успешности выполнения команды
87
88
89 RawReport ::= SEQUENCE {
90   request-id      MessageID,  --- идентификатор запроса
91   stream-id       UTF8String (SIZE (1 .. 256)),  --- идентификатор потока в сессии
92   total-blocks-number INTEGER (0 .. 999999999999),  --- общее количество блоков в отчете
93   block-number    INTEGER (1 .. 1000000000000),  --- порядковый номер текущего блока
94   report-block    RawDataBlock  --- блок данных отчета
95 }
96
97
98 RawDataBlock ::= CHOICE {
99   reports [0] CallsRecords,  --- записи отчетов о соединениях в формате КИД
100  raw-cdr  [1] RawBytesBlock,
101  raw-ip   [2] RawBytesBlock
102 }
103
104 RawBytesBlock ::= SEQUENCE OF RawBytes
105 RawBytes ::= OCTET STRING (SIZE (1 .. 4096))
106
107
108 RawAcknowledgement ::= Acknowledgement
109
110
111 END

```

---