



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

**МСЭ-Т**

СЕКТОР СТАНДАРТИЗАЦИИ  
ЭЛЕКТРОСВЯЗИ МСЭ

**J.177**

(05/2003)

СЕРИЯ J: КАБЕЛЬНЫЕ СЕТИ И ПЕРЕДАЧА  
СИГНАЛОВ ТЕЛЕВИЗИОННЫХ И ЗВУКОВЫХ  
ПРОГРАММ И ДРУГИХ МУЛЬТИМЕДИЙНЫХ  
СИГНАЛОВ

Проект IPCom

---

**Спецификация обслуживания абонентов  
сервером управления вызовами (CMS)  
в сетях IPCom**

Рекомендация МСЭ-Т J.177

---

## РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ J

КАБЕЛЬНЫЕ СЕТИ И ПЕРЕДАЧА СИГНАЛОВ ТЕЛЕВИЗИОННЫХ И ЗВУКОВЫХ ПРОГРАММ  
И ДРУГИХ МУЛЬТИМЕДИЙНЫХ СИГНАЛОВ

Общие Рекомендации	J.1–J.9
Общие спецификации для аналоговой передачи звуковых программ	J.10–J.19
Характеристики показателей качества аналоговых каналов звуковых программ	J.20–J.29
Оборудование и линии, используемые для аналоговых каналов звуковых программ	J.30–J.39
Цифровые кодеры для аналоговых сигналов звуковых программ	J.40–J.49
Цифровая передача сигналов звуковых программ	J.50–J.59
Каналы для аналоговой телевизионной передачи	J.60–J.69
Аналоговая телевизионная передача по металлическим линиям и соединение с радиорелейными линиями	J.70–J.79
Цифровая передача телевизионных сигналов	J.80–J.89
Вспомогательные цифровые услуги для телевизионной передачи	J.90–J.99
Эксплуатационные требования и методы телевизионной передачи	J.100–J.109
Интерактивные системы для распределения цифрового телевидения	J.110–J.129
Транспортирование сигналов MPEG-2 по сетям с пакетной обработкой	J.130–J.139
Измерение качества обслуживания	J.140–J.149
Распределение цифрового телевидения по местным абонентским сетям	J.150–J.159
<b>Проект IP-Cablecom</b>	<b>J.160–J.179</b>
Некоторые аспекты	J.180–J.199
Приложение для интерактивного цифрового телевидения	J.200–J.209

*Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.*

## **Рекомендация МСЭ-Т J.177**

### **Спецификация обслуживания абонентов сервером управления вызовами (CMS) в сетях IPCom**

#### **Резюме**

Обслуживание абонентов в сети IPCom 1.0 может рассматриваться как две различные операции: обслуживание адаптера медиатерминала (МТА) и обслуживание абонентов сервера управления вызовами (CMS). В этой Рекомендации определяется интерфейс, действующий между CMS и сервером обслуживания для обмена информацией об обслуживании.

#### **Источник**

Рекомендация МСЭ-Т J.177 была утверждена 29 мая 2003 года 9-й Исследовательской комиссией МСЭ-Т в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8.

## ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

Всемирная ассамблея по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяет темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

## ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соответствие положениям данной Рекомендации является добровольным делом. Однако в Рекомендации могут содержаться определенные обязательные положения (для обеспечения, например, возможности взаимодействия или применимости), и тогда соответствие данной Рекомендации достигается в том случае, если выполняются все эти обязательные положения. Для выражения требований используются слова "shall" ("должен", "обязан") или некоторые другие обязывающие термины, такие как "must" ("должен"), а также их отрицательные эквиваленты. Использование таких слов не предполагает, что соответствие данной Рекомендации требуется от каждой стороны.

## ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на то, что практическое применение или реализация этой Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для реализации этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ.

© МСЭ 2004

Все права сохранены. Никакая часть данной публикации не может быть воспроизведена с помощью каких-либо средств без письменного разрешения МСЭ.

## СОДЕРЖАНИЕ

	Стр.	
1	Область применения.....	1
1.1	Цели данной Рекомендации .....	1
1.2	Область применения данной Рекомендации .....	1
1.3	Требования.....	1
2	Ссылки.....	2
2.1	Нормативные ссылки.....	2
2.2	Источники ссылок.....	3
3	Термины и определения.....	3
4	Сокращения.....	4
5	Основные положения.....	6
5.1	Задачи службы.....	6
5.2	Эталонная архитектура сети IPCom.....	6
5.3	Компоненты и интерфейсы .....	6
5.4	Компоненты.....	7
5.5	Описание интерфейсов.....	8
6	Допущения .....	8
7	Обслуживание абонентов .....	8
7.1	Учетные документы (биллинг) клиентов.....	8
7.2	Установка и конфигурирование оборудования.....	8
7.3	Статические или динамические данные обслуживания абонентов.....	9
8	Требования .....	9
8.1	Общие требования .....	9
8.2	Требования транспортировки .....	10
9	Модель данных .....	10
9.1	Обзор .....	10
9.2	Связи, кодируемые с использованием объекта PspRelation. Кодирование на языке XML.....	13
10	Передача сообщений.....	14
10.1	Обзор .....	14
10.2	Требования к ролевым именам для передачи сообщений серверов CMS и PS.....	14
10.3	Спецификация языка WSDL .....	15
11	Защита.....	15
	Приложение А – Схема PCSP XML.....	16
	Приложение В – Спецификация языка WSDL для передачи PCSP-сообщений.....	35

	<b>Стр.</b>
Добавление I – Примеры кодирования объектов.....	40
I.1    Пример объекта PccspService.....	40
I.2    Пример объекта PccspEndpoint.....	42
I.3    Пример объекта PccspMta .....	42
I.4    Пример объекта PccspCms .....	43
I.5    Пример PccspRelation .....	43
Добавление II – Пример расширения объекта .....	44
II.1    Пример расширенного объекта PccspService .....	44
II.2    Схема расширения .....	45
Добавление III – Оценка кодирования данных.....	46
III.1    Язык XML.....	46
III.2    Язык ASN.1/BER.....	46
III.3    Патентованный код ASCII .....	46
III.4    SDP (протокол описания сеансов).....	46
III.5    Служба RADIUS.....	46
III.6    Язык структурированных запросов SQL .....	46
III.7    Сводные данные вариантов.....	47
III.8    Рекомендация: язык XML .....	47
Добавление IV – Оценка протокола транспортировки .....	48
IV.1    Протокол TFTP с защитой IPSec .....	48
IV.2    Пакетная служба RADIUS – Несколько записей в одном запросе посредством передачи сообщений о событии.....	48
IV.3    Протокол Diameter .....	48
IV.4    Системы распределенного объекта .....	48
IV.5    Протокол HTTP .....	49
IV.6    Сводные данные вариантов.....	50
IV.7    Рекомендация: протокол HTTP 1.1.....	51
Добавление V – Литература.....	51

## Рекомендация МСЭ-Т J.177

### Спецификация обслуживания абонентов сервером управления вызовами (CMS) в сетях IP-Cablecom

#### 1 Область применения

##### 1.1 Цели данной Рекомендации

В интерфейсе используется модель службы Web. Специфицируемый на языке описания Web-сервиса (WSDL 1.1) [9], этот интерфейс транспортирует закодированные на языке XML объекты в составе сообщений, закодированных по простому протоколу доступа к объектам (SOAP) 1.1 [10], используя процедуру транспортировки HTTP 1.1. Защита этого интерфейса обеспечивается протоколом IPSec.

Модель данных, транспортируемых через этот интерфейс, специально спроектирована в расчете на последующее расширение и допускает включение еще не определенных возможностей сети IP-Cablecom и конкретных расширений фирм-поставщиков.

##### 1.2 Область применения данной Рекомендации

Область применения данной Рекомендации ограничивается обслуживанием сервером CMS сети IP-Cablecom 1.0, выполняемым одним провайдером службы. Кроме того:

- Интерфейс обслуживания CMS ограничивается обменом данных активизации службы между CMS и PS. Интерфейс между PS и системой поддержки операций (OSS) выходит за рамки данной Рекомендации.
- Управление серверными элементами сети и обслуживание элементов сети (планы наборов и т. п.) выходят за рамки данной Рекомендации.
- Создание записей пользователей/выписывание счетов рассматривается как часть применения OSS и в настоящее время выходит за рамки рассмотрения.
- Данная Рекомендация служит структурной базой обслуживания абонентов сети IP-Cablecom сервером CMS. Технические реализации должны соответствовать данной Рекомендации, но не обязательно всегда поддерживать все элементы, параметры и функции, как показано в приложениях.

##### 1.3 Требования

При использовании данной Рекомендации ключевые слова "ДОЛЖЕН" и "СЛЕДУЕТ" нужно рассматривать как указывающее на обязательный характер данной Рекомендации. Ключевые слова, поясняющие определенный уровень важности конкретных требований и используемые на протяжении всей Рекомендации, приводятся ниже:

- |             |  |
|-------------|--|
| "ДОЛЖЕН"    | Это слово, как и прилагательное "ТРЕБУЕМЫЙ", указывает, что соответствующее положение является абсолютным требованием данной Рекомендации.   |
| "НЕ ДОЛЖЕН" | Это выражение означает, что соответствующее положение является абсолютным запрещением в данной Рекомендации.   |
| "СЛЕДУЕТ"   | Это слово, как и прилагательное "РЕКОМЕНДУЕМЫЙ", означает, что в конкретных обстоятельствах могут иметься серьезные причины игнорировать данное положение, но при этом необходимо полностью учитывать последствия и тщательно взвешивать решение перед выбором другого варианта. |

- "НЕ СЛЕДУЕТ" Это выражение означает, что в конкретных обстоятельствах могут существовать серьезные причины, по которым указанное поведение приемлемо или даже полезно, но необходимо полностью понимать последствия и тщательно взвешивать решение перед реализацией любых действий, имеющих такое указание.
- "МОЖЕТ" Это слово, как и прилагательное "НЕОБЯЗАТЕЛЬНЫЙ", означает полную факультативность данного пункта. Например, одна из фирм-поставщиков может учесть данное положение из-за конкретных требований рынка или с целью повысить качество продукции, тогда как другая фирма-поставщик может его не учитывать.

## 2 Ссылки

### 2.1 Нормативные ссылки

Следующие Рекомендации МСЭ-Т и другие ссылки содержат положения, которые через ссылки в данном тексте подтверждают положения данной Рекомендации. На момент публикации действуют указанные издания. Все Рекомендации и другие ссылки могут обновляться; поэтому пользователям Рекомендации предлагается изучать возможность применения более позднего издания данной Рекомендации и других ссылок, перечисленных ниже. Список действующих на данный момент Рекомендаций МСЭ-Т публикуется регулярно. Ссылка на какой-либо документ в данной Рекомендации не придает ему в отдельности статуса Рекомендации.

- [1] ITU-T Recommendation J.160 (2002), *Architectural framework for the delivery of time-critical services over cable television networks using cable modems.*
- [2] ITU-T Recommendation J.167 (2001), *Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems.*
- [3] ITU-T Recommendation J.170 (2002), *IPCablecom security specification.*
- [4] ANSI/SCTE 24-13 (2001), *IPCablecom Part 13: Electronic Surveillance Standard.*
- [5] ISO 639-2:1998, *Codes for the representation of names of languages – Part 2: Alpha-3 code.*
- [6] IETF RFC 1123 (1989), *Requirements for Internet Hosts – Application and Support.*
- [7] ITU-T Recommendation J.161 (2001), *Audio codec requirements for the provision of bidirectional audio service over cable television networks using cable modems.*
- [8] ITU-T Recommendation J.162 (2001), *Network call signalling protocol for the delivery of time-critical services over cable television networks using cable modems.*
- [9] Web Services Description Language. <http://www.w3.org/TR/wsdl>.
- [10] Simple Object Access Protocol. <http://www.w3.org/TR/SOAP>.
- [11] XML Protocol. <http://www.w3.org/2000/xml>.



## 2.2 Источники ссылок

Запросы на комментарии (RFC) IETF:

- Internet Engineering Task Force (IETF) Secretariat c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, Phone 703-620-8990, Fax 703-620-9071, Internet: [www.ietf.org/](http://www.ietf.org/).

Стандарты ИСО:

- International Organization for Standardization (ISO) 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland, Phone 41-22-749-01-11; Fax 41-22-733-34-30, Internet: [www.iso.ch/](http://www.iso.ch/).

Документы Telcordia:

- Telcordia Technologies, 445 South Street, Morristown, NJ 07960-6438, Phone 973-829-2000, Internet: [www.telcordia.com/](http://www.telcordia.com/).

## 3 Термины и определения

В данной Рекомендации определяются следующие термины:

**3.1 активный:** Поток службы называют "активным", когда ему разрешено переносить пакеты данных. До того как стать активным, поток службы должен быть допустимым.

**3.2 оконечная система кабельных модемов:** Устройство на головной станции кабельной сети, которое выполняет MAC-протокол DOCSIS RFI и подключено к кабельным модемам CM по гибридной сети волокну-коаксиал.

**3.3 сервер управления вызовом:** Управляет звуковыми соединениями. В терминологии MGCP/SGCP именуется также "агентом вызова". Является примером прикладного сервера.

**3.4 идентификатор канала связи:** Однозначно идентифицирует канал ISUP DS0 на медиашлюзе (Media Gateway). Является комбинацией кода шлюзового узла канала SS7 и кода идентификации линии связи (CIC). SS7 DPC связан с шлюзом сигнализации, который содержит домен, охватывающий рассматриваемый канал.

**3.5 динамическое качество службы:** Оперативно ("на пролете") присваивается каждой коммуникации в зависимости от запрошенного качества службы QoS.

**3.6 оконечный пункт:** Терминал, межсетевой интерфейс или блок многоточечной конференции.

**3.7 шлюз:** Устройства, сопрягающие сферу связи "Голос по IP" сети IPcablecom и протокол PSTN. Примеры: медиашлюз, который обеспечивает интерфейсы каналов с PSTN и транскодирует медиapotок, и шлюз сигнализации, который передает и принимает сигнализацию по сети коммутируемых каналов на границе сети IPcablecom.

**3.8 заголовок:** Информация управления протокола, помещаемая в начале блока данных протокола.

**3.9 целевая группа по инженерным проблемам Интернет:** Орган, который отвечает помимо других проблем за разработку стандартов, используемых в Интернет. Подробнее см. <http://www.ietf.org/>.

**3.10 обмен ключами Интернет:** Механизм управления ключами, используемый для согласования и получения ключей для SA в IPSec.

**3.11 межсетевой протокол (IP):** Протокол межсетевого (IP) уровня.

**3.12 защита межсетевого протокола (IP):** Комплект стандартов Интернет для защиты IP-пакетов с помощью шифрования и аутентификации.

**3.13 инкапсулированная полезная нагрузка с защитой IPSec:** Протокол, обеспечивающий одновременно шифрование пакета IP и целостность произвольного сообщения, не охватывающий заголовок пакета IP.

**3.14 ключ:** Математическое значение, вводимое в выбранный алгоритм шифрования.

- 3.15 управление ключами:** Процесс распределения совместно используемых симметричных ключей, необходимый для запуска протокола защиты.
- 3.16 переносимость местного номера:** Позволяет пользователю сохранить один и тот же номер при коммутации от одного провайдера местной службы к другому.
- 3.17 медиашлюз:** Обеспечивает интерфейсы канала с коммутируемой телефонной сетью общего пользования (PSTN) и транскодирует медиапоток.
- 3.18 протокол управления медиашлюзом:** Протокол, сопровождающий SGCP. См. IETF RFC 2705.
- 3.19 контроллер медиашлюза:** Общий контроллер, работающий на шлюзе PSTN. Принимает, управляет и служит в качестве посредника для информации сигнализации вызова между IPcablecom и PSTN.
- 3.20 адаптер медиатерминала:** Содержит интерфейс с физическим устройством звука, сетевой интерфейс, кодеки и обеспечивает все функции сигнализации и инкапсуляции, необходимые для транспортировки "ГОЛОСА по IP", сигнализации о классе функций и сигнализации QoS.
- 3.21 система поддержки операций:** Программная поддержка, используемая для управления конфигурированием, производительностью, отказами, калькуляцией и защитой.
- 3.22 заранее установленный ключ совместного использования:** Совместно используемый секретный ключ, сообщаемый обоим участникам коммуникационного обмена с использованием неспецифицированного ручного или внеполосного механизма.
- 3.23 открытый ключ:** Ключ, используемый при засекречивании с открытым ключом, который принадлежит отдельным объектам и распределяется открыто. Другие объекты используют этот ключ при шифровании данных, которые должны передаваться владельцу ключа.
- 3.24 импульсно-кодовая модуляция:** Широко используемый алгоритм цифрового преобразования аналогового сигнала (например, человеческого голоса) в цифровой двоичный поток, использующий метод простого аналого-цифрового преобразования.
- 3.25 регистрация, допуски и состояние:** Канал RAS является ненадежным каналом, используемым для передачи сообщений RAS и для обмена полосами частот между двумя объектами H.323.
- 3.26 запрос комментариев:** Документы технической политики, одобренные целевой группой по инженерным проблемам Интернет (IETF), которые доступны во всемирной паутине (World Wide Web) на <http://www.ietf.cnri.reston.va.us/rfc.html>.
- 3.27 пункт переноса сигнализации:** Узел в сети SS7, который маршрутирует сообщения передачи на основе их адреса назначения. По существу это пакетная коммутация для SS7. Он может также выполнять дополнительные услуги маршрутирования, например глобальный перенос титров (Global Title Translation).
- 3.28 тип-длина-значение:** Кортеж в файле конфигурации DOCSIS.
- 3.29 протокол датаграмм пользователя:** Протокол без установления соединения, построенный на межсетевом протоколе (IP).

#### 4 Сокращения

В данной Рекомендации используются следующие аббревиатуры:

AAA	Аутентификация, авторизация и калькуляция
BPP	Базовое обслуживание обычной телефонной службой
CFP	Обслуживание функциями вызова
CID	Идентификатор канала
CM	Кабельный модем системы DOCSIS
CMS	Синтаксис криптографических сообщений
CMS	Сервер управления вызовами

CMTS	Оконечная система кабельных модемов
Codec	Кодек (COder+DECoder)
CSR	Ответственный за сервис абонента
DOCSIS™	Спецификация интерфейса службы передачи данных по кабелю
DQoS	Динамическое качество службы
ESP	Защита IPSec с инкапсуляцией полезной нагрузки
FQDN	Полностью определенное имя домена
HTTP	Протокол передачи гипертекста. См. IETF RFC 1945 и RFC 2068
IC	Межстанционный носитель
IETF	Целевая группа по инженерным проблемам Интернет
IKE	Обмен ключами в Интернет
IP	Межсетевой протокол
IPSec	Защита меж сетевого протокола
ITU	Международный союз электросвязи (МСЭ)
ITU-T	Международный союз электросвязи – Сектор стандартизации электросвязи (МСЭ-Т)
LNP	Переносимость местного номера
MC	Контроллер многоточечной связи
MG	Медиашлюз
MGC	Контроллер медиашлюза
MGCP	Протокол управления медиашлюза
MIB	База данных управления
MTA	Адаптер медиатерминала
NCS	Сигнализация вызова в сети
OSS	Система поддержки операций
PCM	Импульсно-кодовая модуляция (ИКМ)
PCSP	Обслуживание абонентов сервером управления вызовами (CMS) в сетях IPCom
PS	Сервер обслуживания
PSTN	Телефонная сеть общего пользования (ТСОП)
QCIF	Четвертной общий промежуточный формат
RAS	Регистрация, доступ и состояние
RFC	Запрос комментариев
SDP	Протокол описания сеанса
SNMP	Простой протокол управления сетью
SOAP	Простой протокол доступа к объекту
STP	Пункт передачи сигнализации
TD	Останов для разъединения
TFTP	Простейший протокол передачи файлов

TLV	Тип-Длина-Значение
UDP	Протокол датаграммы пользователя
VoIP	Протокол передачи "Голос по IP"

## 5 Основные положения

### 5.1 Задачи службы

### 5.2 Эталонная архитектура сети IP-Cablecom

На рисунке 1 представлена эталонная архитектура сети IP-Cablecom 1.0. Более подробная информация об эталонной архитектуре IP-Cablecom содержится в [1].

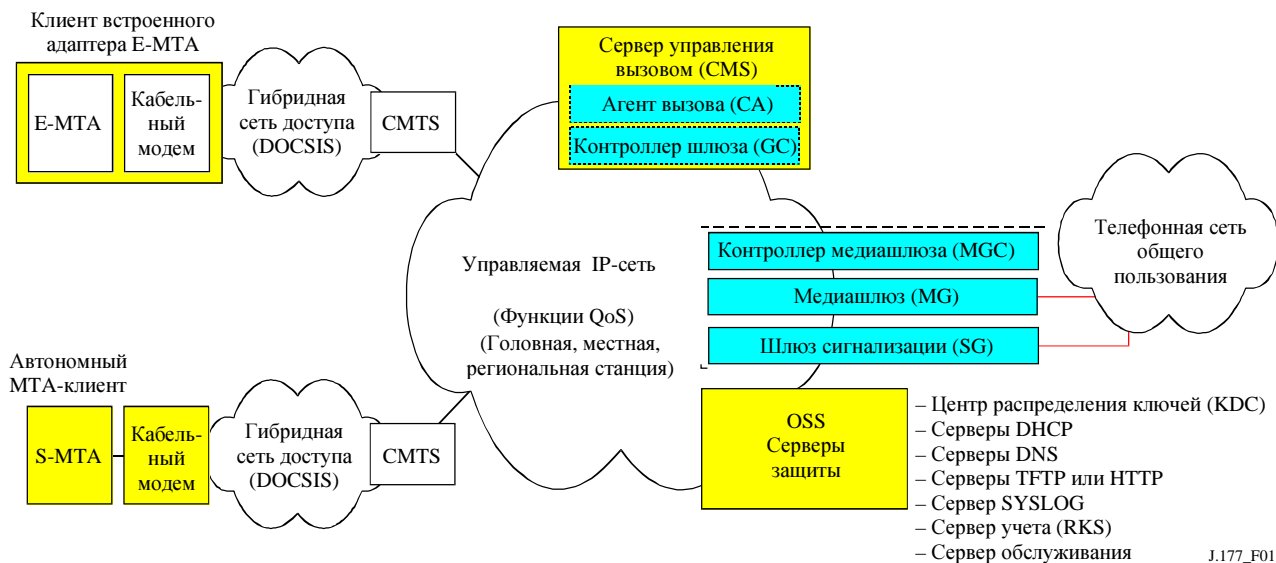


Рисунок 1/J.177 – Эталонная модель (частичная) компонентов сети IP-Cablecom 1.0

### 5.3 Компоненты и интерфейсы

Обслуживание определяется как операции, необходимые для предоставления пользователю определенного сервиса. Обслуживание сервисом IP-Cablecom может рассматриваться как две различные операции: обслуживание MTA и обслуживание абонента сервером CMS. На рисунке 2 показаны связанные с обслуживанием интерфейсы, поддерживаемые сервером PS для различных элементов IP-Cablecom. Интерфейсы, у которых не имеется явной маркировки, являются неопределенными и выходят за рамки IP-Cablecom.

Целью данной Рекомендации является установление требований по обслуживанию интерфейса между серверами CMS и PS (Pkt-prov-p1).

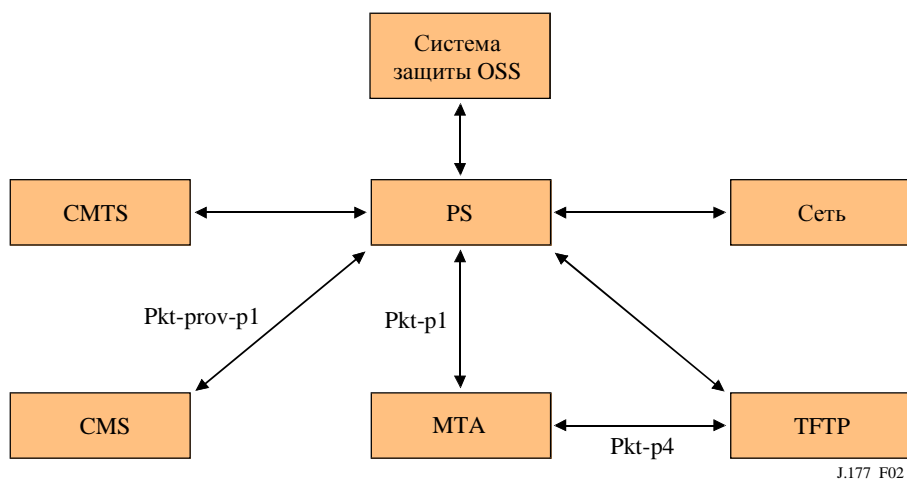


Рисунок 2/J.177 – Обслуживание интерфейсов компонентов

## 5.4 Компоненты

### 5.4.1 Система поддержки операций OSS (обязанности провайдера службы и системы управления службой)

Существуют системы поддержки операций OSS, которые провайдер службы использует для управления пользователями и другими компонентами, что входит в его обязанности. Эти системы обеспечивают процесс обслуживания в сети IP-Cablecom с помощью заказов на выполнение услуг, активизирующих службы для абонентов. Эти системы могут также принимать данные учета или использования для выставления счетов абонентам.

### 5.4.2 Сервер обслуживания

Эта система образует интерфейс между системами OSS провайдеров и элементами IP-Cablecom. IP-Cablecom не обращается относительно реализации этой системы или ее взаимодействия с другими OSS, которую может использовать провайдер службы.

Сервер обслуживания определен в [2] как включающий приложение обслуживания, которое охватывает логику обслуживания и обслуживающий элемент протокола SNMP, который обеспечивает доступ к активным компонентам. Здесь мы будем использовать название сервер обслуживания, не различая эти два объекта.

### 5.4.3 Сервер управления вызовами (СМS)

Сервер управления вызовами описан в [1]. Этот компонент обеспечивает управление вызовами и службы, связанные с сигнализацией, для компонентов МТА и СМТS сети IP-Cablecom.

### 5.4.4 Адаптер медиатерминала (МТА)

Адаптер медиатерминала – это клиентское устройство в IP-Cablecom, которое содержит интерфейс со стороны пользователя для его СРЕ-аппаратуры (например, телефонного аппарата) и интерфейс сигнализации со стороны сети для элементов управления вызовом в сети. Этот компонент описан в [1].

### 5.4.5 Протокол ТFТP

Это служба конфигурационных файлов, которая является основой для конфигурации большинства устройств в сети IP-Cablecom. Она может быть автономной ТFТP-службой, которая доставляет статически определенные файлы на устройства, либо динамической службой, которая "на пролете" создает конфигурации из других источников данных.

### 5.4.6 Оконечная система кабельных модемов (СМТS)

Оконечная система кабельных модемов обеспечивает управление модемами и доставку пакетов в РЧ-сеть и из нее. СМТS также обеспечивает возможности взаимодействия данных в глобальных IP-сетях. Этот компонент описан в [1].

## 5.4.7 Сеть

Совокупность сетевых элементов, которые могут потребоваться в IP-Cablecom, чтобы обеспечить сигнализацию, управление и транспортировку пакетов в/из сетей провайдеров служб.

## 5.5 Описания интерфейсов

### 5.5.1 Pkt-p1

Этот интерфейс описан в [2].

### 5.5.2 Pkt-p4

Этот интерфейс описан в [2].

### 5.5.3 Pkt-prov-p1

Этот интерфейс определен в данной Рекомендации.

## 6 Допущения

- Система поддержки OSS отвечает за обновление координации конечных пунктов с затрагиваемыми объектами соответствующей сети (MTA, CMTS и т. д.) и CMS.
- CMS не выполняет роль администратора и не определяет связь протокола SNMP с адаптером MTA при обслуживании сервера CMS.
- Серверы CMS и PS располагаются в одной области обеспечения защиты. Информация, относящаяся к защите, будет описана в Рекомендации по защите IP-Cablecom [3].

## 7 Обслуживание абонентов

Обслуживание абонентов предусматривает:

- ведение учетных документов/счетов клиентов;
- установку/конфигурирование оборудования.

### 7.1 Учетные документы (биллинг) клиентов

Ведение учетных записей клиентов, которые содержат информацию, необходимую для предоставления службы, биллинга (выписывания счетов) и сбора оплаты от пользователей. Формирование записей/выписывание счетов пользователей рассматривается как часть приложения службы поддержки OSS и в настоящее время выходит за рамки IP-Cablecom.

### 7.2 Установка и конфигурирование оборудования

Эти процедуры могут включать физические установки и/или подключение оборудования, а также любое обновление программного обеспечения и/или баз данных, необходимое для действительного предоставления службы пользователю. Установка оборудования влияет на два основных компонента в среде IP-Cablecom.

- Оборудование в помещении пользователя (CPE). Для IP-Cablecom это адаптер MTA. Обслуживание MTA описано в [2] и не рассматривается в данной Рекомендации.
- Сервер управления вызовами. Обслуживание самого CMS может быть разбито на две основные части: базовое обслуживание POTS и обслуживание функций вызова.

#### 7.2.1 Базовое обслуживание POTS сервером CMS (BPP)

BPP обеспечивает сервер CMS минимальным набором данных, необходимых для маршрутизации простой телефонной службы (POTS) в сети IP-Cablecom. В этот минимальный набор данных входит телефонный номер, отображаемый во взаимодействующих с ним адаптерах MTA доменным именем FQDN и идентификатором конечного пункта NCS. Эти данные будут использоваться для создания таблиц трансляции, разрешающих CMS маршрутизировать вызовы на соответствующее устройство/порт, имеющие конкретный телефонный номер. Обслуживание BPP должно выполняться для каждого пользователя, прежде чем он сможет получить вызовы по сети IP-Cablecom.

## 7.2.2 Обслуживание функций вызова SMS (CFP)

Наряду с ВРР для предоставления пользователю функций вызова выполняется CFP. CFP сложнее, чем ВРР, поскольку проходящие параметры могут меняться от функции к функции, а также могут зависеть от конкретных реализаций фирм-поставщиков. Дополнительный подробный анализ CFP предполагается выполнить в следующих выпусках Рекомендации.

## 7.3 Статические и динамические данные обслуживания абонентов

Данные, необходимые SMS для обслуживания абонентов, можно разбить на два класса:

- 1) Статическое состояние службы с выписанным счетом и постоянным назначением. Эти данные не изменяются от вызова к вызову. Примерами могут служить установки DQoS, состояния оплаченных/неоплаченных функций вызова, информация идентификации вызывающей стороны и т. д.
- 2) Динамическое состояние службы в отсутствие выписанного счета и с полупостоянным назначением. Зачастую эта информация может изменяться абонентом либо в конечной точке с помощью кодового ключа \*XX, либо на SMS с помощью интерфейса Web. Примером может служить устанавливаемая абонентом функция вызова, такая как продвижение вызова по занятой линии (CFBL). Продвижение номера CFBL соответствует динамическому состоянию службы без выписывания счета. Оплаченное/неоплаченное состояние CFBL представляет собой статические данные, поддерживаемые сервером PS.

В сфере действия SMS/PS сети IP-Cablecom к серверу PS относятся все состояния статического обслуживания, а к серверу SMS – все состояния динамического обслуживания.

## 8 Требования

### 8.1 Общие требования

- Интерфейс НЕ ДОЛЖЕН делать допущений относительно технологии реализации серверов PS и SMS.

Серверы SMS и PS, выпускаемые разными фирмами-поставщиками, без сомнения, будут реализованы на различных платформах, отличающихся программным и аппаратным обеспечением и языками разработки. Требуется нейтральный интерфейс для платформ и языков.

- Интерфейс ДОЛЖЕН поддерживать POTS.

Модель данных интерфейса ДОЛЖНА содержать минимальное количество информации, необходимой для поддержки базового обслуживания POTS.

- Интерфейс ДОЛЖЕН поддерживать обслуживание функций вызова.

Модель данных интерфейса ДОЛЖНА поддерживать абонирование на любые функции вызова IP-Cablecom, как определено в [12] и [13].

- Модель данных интерфейса ДОЛЖНА допускать возможность расширения.

В настоящее время интерфейс используется в основном для телефонных данных. Однако интерфейс должен допускать расширение для мультимедийных служб IP-Cablecom. Желательно иметь единую, расширяемую по обслуживанию модель данных и транспортировки для поддержки всех функций и возможностей сети IP-Cablecom, включая те, что еще не определены.

- Интерфейс НЕ ДОЛЖЕН оказывать влияния на любые выполняемые MTA операции.

На адаптере MTA могут быть добавлены, удалены или модифицированы конкретные данные конечных пунктов, без влияния на другие MTA конечных пунктов или на текущие сеансы. Необходимо тщательно документировать сценарий обслуживания сервера SMS конечного пункта, который может привести к выводу из эксплуатации конечного пункта/MTA.

- В интерфейсе ДОЛЖНА быть предусмотрена возможность работы с существующим (NCS) и будущими протоколами сигнализации.

## 8.2 Требования транспортировки

Транспортировка НЕ ДОЛЖНА основываться на предположениях относительно инфраструктуры физической сети между серверами PS и CMS.

Ожидается совместная работа нескольких провайдеров службы в сети с единым управлением. В этих условиях несколько компаний будут связываться между собой, используя, возможно, реализации серверов CMS и PS от различных фирм-поставщиков и при сетях различной инфраструктуры (с разными брандмауэрами, программами-посредниками и т. п.). Протокол транспортировки CMS/PS должен облегчать возможность вхождения в сети с произвольной инфраструктурой.

- Транспортировка ДОЛЖНА поддерживать однонаправленный перенос объектов единой модели данных от PS к CMS.
- Транспортировка ДОЛЖНА поддерживать эффективные транзакции объектов с различными моделями данных от PS к CMS.
- Транспортировка МОЖЕТ поддерживать однонаправленный перенос объектов единой модели данных от CMS к PS.
- Транспортировка МОЖЕТ поддерживать эффективные потоки объектов с различными моделями данных от CMS к PS.
- Транспортировка МОЖЕТ предусматривать семантику для поддержки объектов с новыми, обновленными и удаленными моделями данных.
- Транспортировка ДОЛЖНА поддерживать информационные запросы между серверами PS и CMS.
- Транспортировка ДОЛЖНА понимать такие режимы CMS, как занятость, ошибки и т. п.
- Транспортировка ДОЛЖНА обеспечивать положительное/отрицательное квитирование принятой операции.
- Транспортировка ДОЛЖНА реализовывать хотя бы один тип семантики сообщений. Передатчик НЕ ДОЛЖЕН сбрасывать свой запрос вплоть до квитирования его приемником (сами эти квитирования не квитируются). Транспортировка должна быть способна обнаруживать искажения данных при транспортировке и т. п. и должна уведомлять отправителя о таких ситуациях.
- Транспортировка ДОЛЖНА обеспечивать положительное/отрицательное квитирование выполняемой операции.
- Сервер PS ДОЛЖЕН иметь возможность инициировать передачу объектов модели данных.
- Транспортировка ДОЛЖНА быть защищена.

## 9 Модель данных

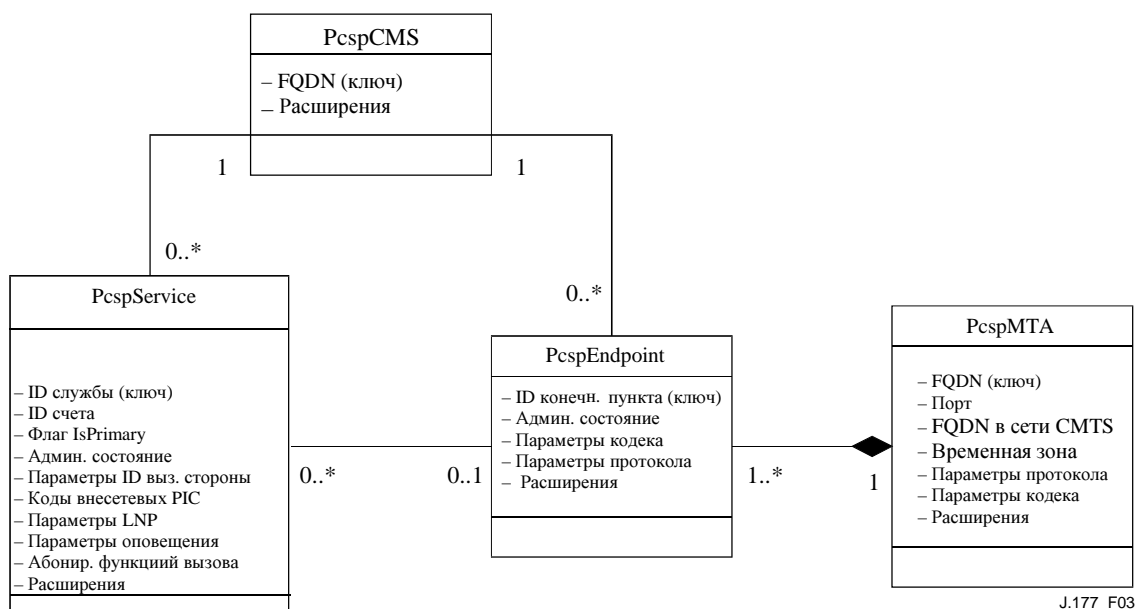
В этом пункте приводится высокоуровневое описание протокола PCSP для модели данных и для ее кодирования на языке XML. Нормативное официальное определение модели данных и ее кодировка приводятся на схеме PCSP XML в Приложении А.

### 9.1 Обзор

Модель данных для обслуживания сервером CMS в сети IP-Cablecom приведена на рисунке 3. Она содержит две категории элементов:

- объекты;
- отношения между объектами.





**Рисунок 3/J.177 – Модель данных при обслуживании сервером CMS<sup>1</sup>**

ДОЛЖНЫ поддерживаться следующие элементы:

- Объект PcspsService – это элемент, на который подписывается пользователь IPCablecom 1.0. Он представляет номер телефона и все связанные с ним функции (вызов и т. п.).
- Объект PcspsMTA представляет адаптер медиатерминала (МТА), который объединяет один или несколько конечных пунктов, физически входящих в его состав.
- Объект PcspsEndpoint представляет физический конечный пункт на адаптере МТА/межсетевом шлюзе.
- Объект PcspsCMS обеспечивает соединения между конечными пунктами/серверами CMS и между службами/серверами CMS.
- Объект PcspsRelations представляет соединения между объектами. На рисунке 3 они представлены как связи между объектами.

PcspsService и PcspsEndpoint сделаны различными объектами, чтобы поддерживать по несколько служб (телефонных номеров) на каждом конечном пункте. Различия между объектами PcspsMta и PcspsEndpoint дают возможность различным провайдерам служб управлять адаптерами МТА конечных пунктов. Объект PcspsCms по существу поддерживает некоторый набор конечных пунктов и служб.

Все объекты допускают расширение.

### 9.1.1 Объект PcspsService

Объект службы – это объект, на который подписывается абонент сети IPCablecom 1.0. Он представляет собой номер телефона и все связанные с ним функции. Модель данных позволяет обеспечивать более одной службы на одном конечном пункте.

Объект PcspsService содержит следующую общую информацию (подробное описание приведено на XML-схеме PCSP):

- ServiceId – уникальный идентификатор службы;
- BillingId – идентификатор другой службы, которой может быть выставлен счет за действия на данной службе;

<sup>1</sup> Проект содержит один или несколько показателей; "1..\*" и другие цепочки, относящиеся к взаимосвязи проекта и активности, называются *многообразием* класса активности в пределах отношения. Дальнейшие детали см. в ссылке, относящейся к Унифицированному языку моделирования (UML).

- Флаг IsPrimary – из множества служб, обеспечиваемых на конечном пункте, одна служба ДОЛЖНА иметь этот флаг, устанавливаемый по умолчанию, для индикации ее использования при исходящих вызовах;
- PrimaryRingPattern – индекс в таблице каденций (последовательностей тонов) МТА, определяющий выбор звуковой структуры вызова для этой службы;
- Административное состояние данной службы (ожидание, разрешение работы, изменение номера и т. п.);
- DisplayName – отображаемая информация, используемая в функции доставки имени вызова (CNAM);
- DisplayNumber – отображаемая информация, используемая в функции доставки номера вызова (CND);
- Установки объявлений (разрешение работы, язык, временная зона и т. п.);
- Коды носителей (код носителя междугородной связи, код носителя местной телефонной сети, код носителя международной связи);
- Управление переносимостью местного номера (состояние переноса, флаг просмотра пункта переноса сигнализации STP и т. п.);
- Функции вызова – служба содержит список объектов характеристик абонированного вызова (как описано в [12] и [13]);
- Расширения – этот объект допускает два вида расширения: в основном теле объекта и в перечне функций вызова.

### 9.1.2 Объект PcsPMTa

Адаптер медиатерминала объединяет один или несколько конечных пунктов (физически содержащихся в адаптере МТА). Он содержит следующую общую информацию (более подробная информация содержится в схеме PCSP XML):

- доменное имя FQDN адаптера МТА, которое однозначно его идентифицирует;
- порт приемника NCS в адаптере МТА (по умолчанию: 2427);
- доменное имя FQDN управляющей CMTS;
- временная зона физического размещения данного МТА;
- назначение протокола сигнализации – выбор протокола по умолчанию для всех конечных пунктов, если он не отменяется на отдельных конечных пунктах;
- назначение кодека – выбор кодека по умолчанию для всех конечных пунктов, если он не переопределяется на отдельных конечных пунктах;
- единственная точка для расширения.

### 9.1.3 Объект PcsPEndpoint

Конечный пункт – это физический порт адаптера МТА/межсетевое шлюза. Он содержит следующую общую информацию (более подробная информация содержится в схеме PCSP XML):

- EndpointId – однозначно идентифицирует этот конечный пункт;
- выбор протокола сигнализации – возможно принудительное переопределение установки МТА;
- административное состояние конечного пункта (отключен, нормальная служба, режим проверки и т. п.);
- выбор кодека – возможно принудительное переопределение установки МТА;
- единственная точка расширения.

### 9.1.4 Объект PcsPCms

Этот объект поддерживает взаимосвязь между конечными пунктами/серверами CMS и службами/серверами CMS. Он содержит следующую общую информацию (более подробная информация содержится в схеме PCSP XML):

- доменное имя FQDN, однозначно идентифицирующее данный CMS;
- единственная точка расширения.

### 9.1.5 Внутриобъектовые связи

На рисунке 3 это линии, которые соединяют классы, соответствующие "отношениям" объекта (иногда называемым ассоциациями). ДОЛЖНЫ поддерживаться следующие отношения, представленные на рисунке 3:

- сервис/CMS – типовой CMS имеет в распоряжении блок телефонных номеров;
- конечный пункт/CMS – конечному пункту требуется CMS для целей сигнализации;
- служба/конечный пункт – телефонный номер должен быть присвоен физическому конечному пункту ;
- конечный пункт/МТА – адаптеры МТА физически содержат конечные пункты.

### 9.2 Связи, кодируемые с использованием объекта PccspRelation. Кодирование на языке XML

Объекты модели данных кодируются с использованием языка XML.

#### 9.2.1 Схема PCSP XML

В Приложении А приведена XML-схема PCSP. Схема определяет синтаксис XML-кодирования для следующих объектов (объекты ДОЛЖНЫ соответствовать этой схеме):

- объекты PccspService, PccspEndpoint, PccspMta и PccspCms – это основные объекты модели данных;
- PccspRelation – этот объект используется для установления или прерывания отношений между объектами;
- PccspImportExport – формат документа общего вида, который может содержать большое число объектов или отношений. В типовом случае он используется для экспорта полных наборов данных из PS в CMS.

Данную схему СЛЕДУЕТ использовать для подтверждения правильности XML-программы синтаксического анализа для определения синтаксической корректности закодированных объектов.

#### 9.2.2 Примеры кодировок PCSP

Примеры XML-кодировок для всех объектов модели данных PCSP приводятся в Добавлении I.

#### 9.2.3 Расширения объектов

XML-схема PCSP обеспечивает возможность расширения для всех объектов (PccspService, PccspEndpoint, PccspMta и PccspCms). Расширения выполняются с помощью элемента <Extension>, помещаемого в каждый объект. Большинство объектов задают этот элемент в конце основного тела объекта. PccspService содержит дополнительный элемент <Extension> в конце перечня функций вызова.

Для элемента <Extension> существует несколько простых правил.

- Все элементы <Extension> ДОЛЖНЫ специфицировать определение пространства имен.
- Все субэлементы <Extension> должны определяться в пространстве имен.

Эти два правила позволяют XML-системе синтаксического анализа проверить достоверность содержания <Extension> относительно файла XML-схемы, поставляемого фирмой-поставщиком. Пример расширения приводится в Добавлении II.

## 10 Передача сообщений

### 10.1 Обзор

Интерфейс PCSP соответствует системе понятий службы Web. Интерфейс должен использовать сообщения SOAP 1.1 для передачи XML-кодированных объектов (от модели данных PCSP) между клиентом и сервером. Сообщения ДОЛЖНЫ транспортироваться между клиентом и сервером с использованием протокола HTTP 1.1. Детальное рассмотрение условий транспортировки приводится в Добавлении III.

Интерфейс моделируется на синхронной структуре запросов/откликов (или дистанционном вызове процедуры – RPC). Между клиентом и сервером поддерживаются следующие структуры передачи сообщений:

- Сообщение ПОМЕСТИТЬ – клиент записывает на сервере один или несколько XML-кодированных объектов или отношений. Поддерживается как создание новых, так и модификация существующих объектов.
- Сообщение УДАЛИТЬ – клиент запрашивает удаление с сервера одного или нескольких объектов или отношений.
- Сообщение ПОЛУЧИТЬ – считать с сервера один или несколько XML-кодированных объектов (только для объектов, для отношений не поддерживается).
- Сообщение СМД-СОСТОЯНИЕ – используется для переноса "внеполосных" команд и состояний между клиентом и сервером. Клиент может уведомить сервер о различных условиях состояния. Клиент может приказать серверу выполнить различные действия. Это сообщение может расширяться фирмами-поставщиками.

### 10.2 Требования к ролевым именам для передачи сообщений серверов CMS и PS

В общем, оба сервера CMS и PS МОГУТ быть реализованы с полной поддержкой ролевых имен для передачи сообщений клиента и сервера. Однако в рамках обслуживания сервером CMS сети IPcablecom серверы CMS и PS принимают ролевые имена, определяемые в таблице 1.

Таблица 1/J.177 – Ролевые имена передачи сообщений серверами CMS и PS

Сообщение	CMS как клиент	CMS как сервер	PS как клиент	PS как сервер
ПОЛУЧИТЬ	НЕ ОБЯЗАТЕЛЬНО	ДОЛЖЕН	ДОЛЖЕН	НЕ ОБЯЗАТЕЛЬНО
ПОМЕСТИТЬ	НЕ ОБЯЗАТЕЛЬНО	ДОЛЖЕН	ДОЛЖЕН	НЕ ОБЯЗАТЕЛЬНО
УДАЛИТЬ	НЕ ОБЯЗАТЕЛЬНО	ДОЛЖЕН	ДОЛЖЕН	НЕ ОБЯЗАТЕЛЬНО
СМД-СОСТОЯНИЕ	ДОЛЖЕН	ДОЛЖЕН	ДОЛЖЕН	ДОЛЖЕН

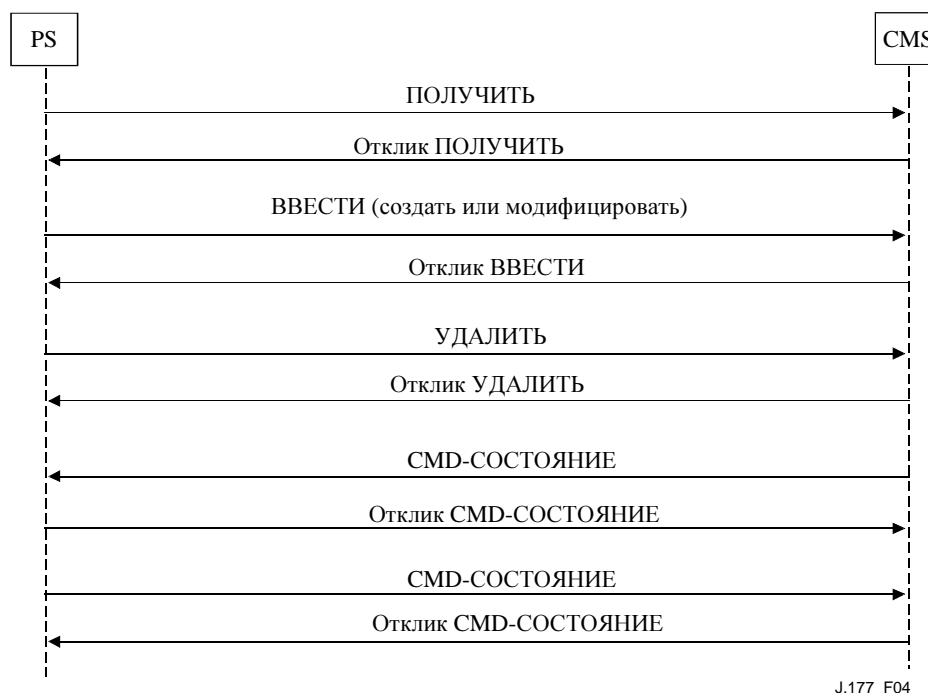
Необходимо отметить следующие моменты:

- Сервер CMS ДОЛЖЕН поддерживать ролевое имя сервера для функций ПОЛУЧИТЬ, ПОМЕСТИТЬ и УДАЛИТЬ.
- Сервер PS ДОЛЖЕН поддерживать ролевое имя клиента для функций ПОЛУЧИТЬ, ПОМЕСТИТЬ и УДАЛИТЬ.
- Серверы CMS и PS ДОЛЖНЫ поддерживать ролевые имена клиента и сервера для функции СМД-СОСТОЯНИЕ.
- Все остальные режимы работы НЕ ОБЯЗАТЕЛЬНЫ.

Эти требования обеспечивают предоставление потоков данных от PS к CMS, а также позволяют CMS обходиться без продвижения динамических изменений данных (изменений характеристик вызова, выполняемых пользователем) назад к PS.

Сервер PS может считывать конкретные объекты с сервера CMS. Такой вариант использования поддерживается прежде всего для того, чтобы позволить серверу PS получать выполняемые пользователем установки функций вызова ("динамические данные"), принадлежащие серверу CMS. Это реализуется путем считывания конкретных объектов PcsService с CMS.

На рисунке 4 показаны все ролевые имена, необходимые для передачи сообщений.



**Рисунок 4/J.177 – Необходимые потоки передачи сообщений**

### 10.3 Спецификация языка WSDL

Интерфейс PCSP специфицируется с помощью описательного языка службы Web 1.1. Как и в случае применения CORBA IDL, WSDL-определение интерфейса специфицирует на интерфейсе дистанционные методы, аргументы для принятых методов, обратные значения для методов и любые типы конкретных данных интерфейса, которые требуют определения. Кроме того, WSDL-определение специфицирует также формат кодировки сообщений (SOAP 1.1) и транспортную привязку (HTTP 1.1).

WSDL вводится в различные инструментальные средства Web-сервиса, доступные для большинства операционных систем и языков, с целью автоматического генерирования клиентских "заглушек" интерфейсов, структур серверных программ и поддержки упорядочения протоколов SOAP.

Клиенты и серверы PCSP ДОЛЖНЫ базироваться на языке WSDL 1.1, ДОЛЖНЫ выполнять транспортную привязку HTTP 1.1 и ДОЛЖНЫ использовать WSDL-определение, представленное в Приложении В.

## 11 Защита

Защита интерфейса PCSP обеспечивается применением ESP-протокола IPsec в режиме транспортировки. Управление ключами реализуется благодаря применению IKE с заранее установленными (pre-shared) ключами совместного использования. Такая инфраструктура защиты уже используется серверами CMS для различных интерфейсов. Подробное описание см. в [3].

## Приложение А

### Схема PCSP XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!--отредактировано с помощью XML Spy v4.0.1 U (http://www.xmlspy.com) Полем
Дюффи (Paul Duffy) (частное сообщение) -->
<!--
  Схема обслуживания абонентов/предоставления услуг сервером CMS в сетях
  IPCablecom (PCSP).

  PCSP определяет интерфейс передачи сообщений и формат XML-кодировки для
  объектов, передаваемых через этот интерфейс. Эта схема определяет синтаксис
  XML-кодировки для объектов, передаваемых через интерфейс PCSP.

  Специфицируются кодировки для объектов PccspService, PccspEndpoint, PccspMta и
  PccspCms.
  Кодировка PccspRelation описывает связи между объектами.
  Кодировка PccspImportExport используется для создания отдельных документов,
  содержащих большое число объектов.
-->
<xs:schema targetNamespace="unique_fully_qualified_namespace"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="unique_fully_qualified_namespace" elementFormDefault="qualified">
<!-- ПРИМЕЧАНИЕ: для реализации XML-схемы параметр
<unique_fully_qualified_namespace> ДОЛЖЕН быть заменен на уникальный
идентификатор.
-->
  ===== ОПРЕДЕЛЕНИЯ ТИПА =====
  -->
  <!--
  Непустая цепочка.
  -->
  <xs:simpleType name="nonEmptyString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <!--
  Идентификатор службы.
  Ненулевая цепочка, содержащая атрибут "формата" (перечисление), который по
  умолчанию является NSN.
  -->
  <xs:complexType name="ServiceIdType">
    <xs:simpleContent>
      <xs:extension base="nonEmptyString">
        <xs:attribute name="format" default="NSN">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="NSN"/>
              <xs:enumeration value="E164"/>
              <xs:enumeration value="ENUM"/>
              <xs:enumeration value="URL"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <!--
  Тип операции отношения.
  Используется для индикации "добавляемого" или "удаляемого" отношения.
  -->
  <xs:simpleType name="RelationOpType">
    <xs:restriction base="xs:string">
```

```

        <xs:enumeration value="add"/>
        <xs:enumeration value="delete"/>
    </xs:restriction>
</xs:simpleType>
<!--
Перечисление имен "классов" разрешенных объектов.
-->
<xs:simpleType name="classType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="PcspService"/>
        <xs:enumeration value="PcspCms"/>
        <xs:enumeration value="PcspEndpoint"/>
        <xs:enumeration value="PcspMta"/>
    </xs:restriction>
</xs:simpleType>
<!--
Список ключей объектов.
-->
<xs:complexType name="ListOfKeys">
    <xs:sequence>
        <xs:element name="Key" type="xs:string" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<!--
Типы кодеков.
Перечисление, согласующееся с объектом PktcCodecType из Спецификации
Аудио/Видео кодеков сети IPCablecom™ PKT-SP-CODEC-I03-011221.
Это перечисление должно синхронизироваться с вышеупомянутой спецификацией.

В случае объекта PcspEndpoint следует понимать значение для кодека
"неизвестен" как "использовать спецификацию кодека МТА-адаптера".
-->
<xs:simpleType name="codecType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="other"/>
        <xs:enumeration value="unknown"/>
        <xs:enumeration value="G729"/>
        <xs:enumeration value="reserved"/>
        <xs:enumeration value="G729E"/>
        <xs:enumeration value="PCMU"/>
        <xs:enumeration value="G726-32"/>
        <xs:enumeration value="G728"/>
        <xs:enumeration value="PCMA"/>
        <xs:enumeration value="G726-16"/>
        <xs:enumeration value="G726-24"/>
        <xs:enumeration value="G726-40"/>
    </xs:restriction>
</xs:simpleType>
<!--
Обозначения протоколов сигнализации.
На PcspEndpoint используется "MtaDefault" для принудительного использования
установки МТА-адаптера по умолчанию. -->
<xs:simpleType name="protocolType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="MCGP 1.0 NCS 1.0"/>
        <xs:enumeration value="MtaDefault"/>
    </xs:restriction>
</xs:simpleType>
<!--
Цифровое обозначение временной зоны по RFC 1123.
-->

```

```

<xs:simpleType name="NumTimezoneType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[\+\-]\d{4}"/>
  </xs:restriction>
</xs:simpleType>
<!--
  Обозначение временной зоны МОС (ISO) для континента/города или страны/города.
-->
<xs:simpleType name="IsoTimezoneType">
  <xs:element name="Country" type="xs:string"/>
  <xs:element name="City" type="xs:string"/>
</xs:simpleType>

<xs:complexType name="timezoneType">
  <xs:choice>
    <xs:element ref="NumTimezoneType">
    <xs:element ref="IsoTimezoneType">
  </xs:choice>
</xs:complexType>
<!--
  Состояние переноса местного номера

    0: не переносится.
    1: загружается (принадлежит другому TSP)
    2: выгружается (загружается на другой TSP)
-->
<xs:simpleType name="portingStatusType">
  <xs:restriction base="xs:integer">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1"/>
    <xs:enumeration value="2"/>
  </xs:restriction>
</xs:simpleType>
<!--
  ===== ОПИСАНИЯ ЭЛЕМЕНТОВ ПОДДЕРЖКИ =====
-->
<!--
  Управление сетевыми объявлениями. Содержит...

  Язык - тип языка для XML-схемы.

  Временная зона - см. предыдущее определение.
-->
<xs:element name="Announcements">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Language" type="xs:language"/>
      <xs:element name="Timezone" type="timezoneType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
  Коды межстанционных носителей. Используются для конкретных носителей при
  маршрутизации внесетевых региональных вызовов, междугородных вызовов и
  международных вызовов.

  PIC - Заранее назначенный межстанционный носитель (междугородный)

  LPIC - Заранее назначенный носитель местной телефонной сети
  IPIC - Заранее назначенный носитель международного обмена
-->
<xs:element name="InterExchange">
  <xs:complexType>
    <xs:sequence>

```



```

    <xs:element name="PIC" type="xs:int"/>
    <xs:element name="LPIC" type="xs:int"/>
    <xs:element name="IPIC" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<!--

```

Параметры переносимости местного номера.

Состояние переноса – см. `portingStatusType` (тип состояния переноса)

Коммутация LNPT – LNP определяет, находится ли номер в состоянии перехода. ложно/0: табличный преобразователь STP не требуется. истинно/1: табличный преобразователь STP требуется для определения LRN коммутатора назначения.

```

-->
<xs:element name="LNP">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PortingStatus" type="portingStatusType"/>
      <xs:element name="LNPT" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--

```

Элемент расширения фирмы-поставщика.

Используется в составе объектов `PcspService`, `PcspCms`, `PcspMta` и `PcspEndpoint` для разрешения расширений фирм-поставщиков.

Используется также для расширения списка функций вызова в объекте `PcspService`.

```

-->
<xs:element name="Extension">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" processContents="strict" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--

```

===== ФУНКЦИИ ВЫЗОВА =====

Служба содержит список объектов с функциями вызова, каждый из которых кодирует одну из функций вызова, описанных в PКТ-TR-VOIPBRF-R01-000608 и PКТ-TR-VOIPERF-R01-000831.

Каждая функция вызова содержит данные ее "статического" состояния (принадлежащие серверу PS):

- имя функции (неявно, как имя элемента),
- абонированное/неабонированное состояние,
- административное состояние функции.

Многие функции вызова содержат только эту информацию.

Отсутствие конкретной функции вызова означает, что эта функция не абонируется.

Абонированное состояние используется для указания того, что явно перечисленная функция вызова не абонируется (нетипичный случай).

Некоторые характеристики расширяют набор "статических" параметров за счет данных, присущих конкретной функции.

Эти присущие конкретной функции данные в типичном случае конфигурируются пользователем (с помощью микротелефона или вызова CSR).

Спецификация PCSP классифицирует настраиваемые пользователем данные как "динамические", подразумевая, что они принадлежат CMS. Для продвижения данных назад на PS не требуется изменять динамические данные на CMS.

```

-->

```

```

<!--
  Всегда -
    ложно/0: Абонент может изменить продвигаемый номер;
    истинно/1: Только провайдер службы может изменить продвигаемый номер.
-->
<xs:element name="Always" type="xs:boolean"/>
<!--
  ForwardTo - Идентификатор службы, на которую нужно продвинуть вызов.
  ПРИМЕЧАНИЕ: допускаются пустые цепочки.
-->
<xs:element name="ForwardTo" type="xs:string"/>
<!--
  ListOfServiceId - список идентификаторов служб.
-->
<xs:element name="ListOfServiceId">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ServiceId" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
  ListOfSpeedDial - список идентификаторов службы / скоростей набора # пар.
  Каждая пара содержит однозначное или двузначное число скорости набора и
идентификатор соответствующей службы.
-->
<xs:element name="SdPair">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SdNum">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="99"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="ServiceId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ListOfSpeedDial">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="SdPair" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
  Определение каждой поддерживаемой функции вызова...

  Все функции вызова можно рассматривать как состоящие из двух частей.
  1. Общая секция содержит административное состояние функции ("статические"
данные) .
  2. Необязательная секция, содержащая параметры, присущие функции, в
типичном случае устанавливаемые конечным пользователем ("динамические" данные).
-->
<!--
  "Базовый объект" для всех функций вызова, содержащий:

  Абонирование -
    0/ложно: функция не абонируется.
    1/истинно: функция абонируется.

```

Административное состояние (AdminStatus) -

0: функция приостановлена провайдером службы.

1: функция разрешена провайдером службы.

В общем случае присутствие функции вызова подразумевает ее абонирование.

Флаг абонирования поддерживается в случае нетипичной необходимости указать, что явно перечисленная характеристика вызова не абонируется.

```
-->
<xs:complexType name="CfBase">
  <xs:sequence>
    <xs:element name="Subscribed" type="xs:boolean"/>
    <xs:element name="AdminStatus">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:enumeration value="0"/>
          <xs:enumeration value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<!--
  "CND": Доставка номера вызова
-->
<xs:element name="CfCND">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CNAM": Доставка имени вызова
-->
<xs:element name="CfCNAM">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CIDCW": Доставка идентичности вызова после ожидания вызова
-->
<xs:element name="CfCIDCW">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CW": Ожидание вызова
-->
<xs:element name="CfCW">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CCW": Отмена ожидания вызова (*70)
```

```
-->
<xs:element name="CfCCW">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CFV": параметр продвижения вызова и продвижение вызова, критичного
относительно применения (*72/*73)
```

Расширяет CfBase следующим образом:

Активное -

0/ложно: пользователь имеет отключенную функцию (\*73).

1/истинно: пользователь имеет включенную функцию (\*72).

```
-->
<xs:element name="CfCFV">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "AR": Автоматический повторный вызов (*69)
-->
<xs:element name="CfAR">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "AC": Автоматический обратный вызов (*66)
-->
<xs:element name="CfAC">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "VMWI": Визуальный индикатор ожидания сообщения
-->
<xs:element name="CfVMWI">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
```

```

</xs:element>
<!--
  "COT": Трасса, созданная абонентом (*57)
-->
<xs:element name="CfCOT">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "TWC": Трехсторонний вызов/Чувствительный к применению трехсторонний вызов (*71)
-->
<xs:element name="CfTWC">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "RACF": Дистанционная активация продвижения вызова
-->
<xs:element name="CfRACF">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "OCAА": Область оповещения для внешнего вызова
-->
<xs:element name="CfOCAА">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CIES": Идентификация вызова с усиленным просеиванием
-->
<xs:element name="CfCIES">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "ACR": Отклонение анонимного вызова (*77 / *87)

  Расширяет CfBase следующим образом:

  Активный -
  0/ложно: пользователь обладает деактивированной функцией (*87).
  1/истинно: пользователь обладает активированной функцией (*77).
-->
<xs:element name="CfACR">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">

```

```

    <xs:sequence>
      <xs:element name="UserParams" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Active" type="xs:boolean"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<!--
  "AC-R": Автоматический обратный вызов - Ограничение
-->
<xs:element name="CfAC-R">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "ACB": Блокировка автоматического повторного вызова
-->
<xs:element name="CfACB">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CIDB": Блокировка доставки идентичности вызова (*67 / *82)

Расширяет CfBase следующим образом:

Флаг -
  "PUBLIC": доставляет информацию идентификатора вызывающей стороны.
  "ANONYMOUS": не доставляет информацию идентификатора вызывающей стороны.
-->
<xs:element name="CfCIDB">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Flag">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="PUBLIC"/>
                      <xs:enumeration value="ANONYMOUS"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:element>

```

```

    </xs:complexType>
</xs:element>
<!--
    "CFBL": Занятая линия продвижения вызова ( *68 / *40 / *88 )

    Расширяет CfBase за счет следующих "динамических", настраиваемых
    пользователем параметров (принадлежит CMS).

    Активный -
    0/ложно: пользователь обладает деактивированной функцией (*88).
    1/истинно: пользователь обладает активированной функцией (*68/*40).

```

Всегда - см. предыдущее определение.

Продвинуть к (ForwardTo) - см. предыдущее определение.

```

-->
<xs:element name="CfCFBL">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element ref="Always"/>
                <xs:element ref="ForwardTo"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--

```

"CFDA": Не отвечает на продвижение вызова (\*68 / \*42 / \*88)

Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

```

Активный -
    0/ложно: пользователь обладает деактивированной функцией (*88).
    1/истинно: пользователь обладает активированной функцией (*68/*42).

```

Всегда - см. предыдущее определение.

Период посылки вызова (RingPeriod) - число периодов посылок вызова, после которого запускается продвижение.

Продвинуть к (ForwardTo) - см. предыдущее определение.

```

-->
<xs:element name="CfCFDA">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element ref="Always"/>
                <xs:element name="RingPeriod" type="xs:int"/>
                <xs:element ref="ForwardTo"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

```

        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<!--
  "CFC": Комбинация продвижения вызова

  Расширяет CfBase за счет следующих "динамических", настраиваемых
  пользователем параметров (принадлежит CMS):

  Активный -
    0/ложно: пользователь обладает деактивированной функцией (*88).
    1/истинно: пользователь обладает активированной функцией (*68).

  Всегда - см. предыдущее определение.

  Период вызова (RingPeriod) - число периодов посылок вызова, после которого
  запускается продвижение.

  Продвинуть к (ForwardTo) - см. предыдущее определение.
-->
<xs:element name="CfCFC">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element ref="Always"/>
                <xs:element name="RingPeriod" type="xs:int"/>
                <xs:element ref="ForwardTo"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "SCF": Продвижение избирательного вызова (*63/*83)

  Расширяет CfBase за счет следующих "динамических", настраиваемых
  пользователем параметров (принадлежит CMS):

  Активный -
    0/ложно: пользователь обладает деактивированной функцией (*83).
    1/истинно: пользователь обладает активированной функцией (*63).

  Список идентификаторов служб (ListOfServiceId) - список идентификаторов
  служб, которые будут продвинуты. См. определение предыдущего элемента.
  Продвинуть к (ForwardTo) - см. предыдущее определение.
-->
<xs:element name="CfSCF">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">

```



```

        <xs:complexType>
          <xs:sequence>
            <xs:element name="Active" type="xs:boolean"/>
            <xs:element ref="ListOfServiceId"/>
            <xs:element ref="ForwardTo"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<!--

```

"SCR": Отклонение избирательного вызова (\*60 / \*80 )

Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

Активный -

- 0/ложно: пользователь обладает деактивированной функцией (\*80).
- 1/истинно: пользователь обладает активированной функцией (\*60).

Список идентификаторов служб (ListOfServiceId) - список идентификаторов служб, которые будут отклонены. См. определение предыдущего элемента.

```

-->
<xs:element name="CfSCR">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element ref="ListOfServiceId"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--

```

"DRCW": Отчетливый вызов(звон)/Ожидание вызова (\*61 / \*81)

Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

Активный -

- 0/ложно: пользователь обладает деактивированной функцией (\*81).
- 1/истинно: пользователь обладает активированной функцией (\*61).

Список идентификаторов служб (ListOfServiceId) - список идентификаторов входящих служб, которые будут принимать отличительный звонок(вызов) (по сравнению со стандартным звонком или тона ожидания вызова). См. определение предыдущего элемента.

```

-->
<xs:element name="CfDRCW">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">

```

```

    <xs:sequence>
      <xs:element name="UserParams" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Active" type="xs:boolean"/>
            <xs:element ref="ListOfServiceId"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>

```

<!--

"SPCALL": Сокращенный набор номера (\*74 / \*75)

Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

Список скоростей набора (ListOfSpeedDial) – см. определение предыдущего элемента.

-->

```

<xs:element name="CfSPCALL">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="ListOfSpeedDial"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

<!--

"RDA": Служба отличительного оповещения по местожительству

-->

```

<xs:element name="CfRDA">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

<!--

"LSR": Ограничение линейной службы

Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

BlkDomLongDist – блок исходящих внутренних междугородных вызовов.

0/ложно: не блокируется.

1/истинно: блокируется.

BlkIntlLongDist – блок исходящих международных вызовов.

0/ложно: не блокируется.

1/истинно: блокируется.

BlkPayPerCall – блок исходящих вызовов с оплатой за вызов (900/976).

0/ложно: не блокируется.

1/истинно: блокируется.

BlkOperatorAssist - блок исходящих вызовов с помощью оператора.  
 0/ложно: не блокируется.  
 1/истинно: блокируется.

BlkDirAssist - блок исходящих вызовов с помощью справочника.  
 0/ложно: не блокируется.  
 1/истинно: блокируется.

BlkTollFree - блок исходящих бесплатных вызовов.  
 0/ложно: не блокируется.  
 1/истинно: блокируется.

ServiceList - список идентификаторов служб для постоянно разрешенных внутренних междугородных вызовов.

```
-->
<xs:element name="CfLSR">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="BlkDomLongDist" type="xs:boolean"/>
                <xs:element name="BlkIntLongDist" type="xs:boolean"/>
                <xs:element name="BlkPayPerCall" type="xs:boolean"/>
                <xs:element name="BlkOperatorAssist" type="xs:boolean"/>
                <xs:element name="BlkDirAssist" type="xs:boolean"/>
                <xs:element name="BlkTollFree" type="xs:boolean"/>
                <xs:element ref="ListOfServiceId"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

<!--  
 "DND": Не беспокоить  
 Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

Активный -

0/ложно: пользователь обладает деактивированной функцией.  
 1/истинно: пользователь обладает активированной функцией.

WeekDayStartTod1 - начальное время дня недели для DND.  
 WeekDayStopTod1 - конечное время дня недели для DND.  
 WeekDayStartTod2 - начальное время дня недели для DND.  
 WeekDayStopTod2 - конечное время дня недели для DND.  
 WeekEndStartTod1 - начальное время для конца недели для DND.  
 WeekEndStopTod1 - конечное время для конца недели для DND.  
 WeekEndStartTod2 - начальное время для конца недели для DND.  
 WeekEndStopTod2 - конечное время для конца недели для DND.

```
-->
<xs:element name="CfDND">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element name="WdStartTod1" type="xs:time"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

```

        <xs:element name="WdStopTod1" type="xs:time"/>
        <xs:element name="WdStartTod2" type="xs:time"/>
        <xs:element name="WdStopTod2" type="xs:time"/>
        <xs:element name="WeStartTod1" type="xs:time"/>
        <xs:element name="WeStopTod1" type="xs:time"/>
        <xs:element name="WeStartTod2" type="xs:time"/>
        <xs:element name="WeStopTod2" type="xs:time"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<!--

```

"СОС": "Комендантский час" для вызовов

Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

Активный -

0/ложно: пользователь обладает деактивированной функцией.

1/истинно: пользователь обладает активированной функцией.

StartTod - начальное время для СОС.

StopTod - конечное время для СОС.

ServiceList - список идентификаторов служб для входящих и исходящих служб, которым разрешается игнорировать NSA.

```

-->
<xs:element name="CfCOC">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element name="StartTod" type="xs:time"/>
                <xs:element name="StopTod" type="xs:time"/>
                <xs:element ref="ListOfServiceId"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--

```

"NSA": Уведомление об отсутствии запросов

Расширяет CfBase за счет следующих "динамических", настраиваемых пользователем параметров (принадлежит CMS):

Активный -

0/ложно: пользователь обладает деактивированной функцией.

1/истинно: пользователь обладает активированной функцией.

StartTod - начальное время для СОС.

StopTod - конечное время для СОС.

`ServiceList` - список идентификаторов служб для входящих и исходящих служб, которым разрешается игнорировать NSA.

```
-->
<xs:element name="CfNSA">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element name="StartTod" type="xs:time"/>
                <xs:element name="StopTod" type="xs:time"/>
                <xs:element ref="ListOfServiceId"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

<!--

Список функций вызова. Этот список может содержать не больше одной из перечисленных выше функций совместно с функциями расширения вызова от фирмы-поставщика.

-->

```
<xs:element name="ListOfCallFeatures">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="CfCND" minOccurs="0"/>
      <xs:element ref="CfCNAM" minOccurs="0"/>
      <xs:element ref="CfCIDCW" minOccurs="0"/>
      <xs:element ref="CfCW" minOccurs="0"/>
      <xs:element ref="CfCCW" minOccurs="0"/>
      <xs:element ref="CfCFV" minOccurs="0"/>
      <xs:element ref="CfAR" minOccurs="0"/>
      <xs:element ref="CfAC" minOccurs="0"/>
      <xs:element ref="CfVMWI" minOccurs="0"/>
      <xs:element ref="CfCOT" minOccurs="0"/>
      <xs:element ref="CfTWC" minOccurs="0"/>
      <xs:element ref="CfRACF" minOccurs="0"/>
      <xs:element ref="CfOCAA" minOccurs="0"/>
      <xs:element ref="CfCIES" minOccurs="0"/>
      <xs:element ref="CfACR" minOccurs="0"/>
      <xs:element ref="CfAC-R" minOccurs="0"/>
      <xs:element ref="CfACB" minOccurs="0"/>
      <xs:element ref="CfCIDB" minOccurs="0"/>
      <xs:element ref="CfCFBL" minOccurs="0"/>
      <xs:element ref="CfCFDA" minOccurs="0"/>
      <xs:element ref="CfCFC" minOccurs="0"/>
      <xs:element ref="CfSCF" minOccurs="0"/>
      <xs:element ref="CfSCR" minOccurs="0"/>
      <xs:element ref="CfDRCW" minOccurs="0"/>
      <xs:element ref="CfSPCALL" minOccurs="0"/>
      <xs:element ref="CfRDA" minOccurs="0"/>
      <xs:element ref="CfLSR" minOccurs="0"/>
      <xs:element ref="CfDND" minOccurs="0"/>
      <xs:element ref="CfCOC" minOccurs="0"/>
      <xs:element ref="CfNSA" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

```
</xs:element>
<!--
===== ОПРЕДЕЛЕНИЯ ОСНОВНЫХ ОБЪЕКТОВ =====
```

На схеме PCSP определены 6 кодировок.

4 кодировки основных объектов:

PcspCms – сервер CMS. Набор служб и конечных пунктов.

PcspService – представляет телефонный номер, его конфигурацию и функции вызова.

PcspMta – представляет физический адаптер МТА и его конфигурацию. Набор конечных пунктов.

PcspEndpoint – Представляет конечный пункт на адаптере МТА.

Объект PcspRelation. Этот объект кодирует связи между объектами.

Объект PcspImportExport. Используется для создания файла массива загрузки для сервера CMS.

```
-->
```

```
<!--
```

PcspRelation.

Объект отношений специфицирует межобъектные связи объектов PcspCms, PcspService, PcspEndpoint и PcspMta.

Атрибут "relOp" специфицирует добавление или удаление отношения.

```
-->
```

```
<xs:element name="PcspRelation">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element name="Class1" type="classType"/>
```

```
<xs:element name="Key" type="xs:string"/>
```

```
<xs:element name="Class2" type="classType"/>
```

```
<xs:element name="ListOfKeys" type="ListOfKeys"/>
```

```
</xs:sequence>
```

```
<xs:attribute name="relOp" type="RelationOpType" use="required"/>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<!--
```

Объект PcspCms.

Этот объект поддерживает связи между конечными пунктами, службами и управляющими ими серверами CMS.

Содержание...

CmsFqdn – Имя FQDN, однозначно идентифицирующее данный сервер CMS.

```
-->
```

```
<xs:element name="PcspCms">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element name="CmsFqdn" type="nonEmptyString"/>
```

```
<xs:element ref="Extension" minOccurs="0"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<!--
```

Объект PcspEndpoint.

Конечный пункт – это физический порт адаптера МТА/шлюза.

Содержание...

**EndpointId** – Однозначно идентифицирует данный конечный пункт. Формат согласно "Спецификации протокола сигнализации вызовов, основанной на сети IP-Cablecom".

Пример: "aaln/1@mta01.cablelabs.com"

**AdminStatus** –

0: конечный пункт отключен

1: нормальное – конечный пункт эксплуатируется

2: режим испытаний – конечный пункт проходит испытания.

Протокол – необязательное переопределение установок протокола MTA.

Кодек – необязательное переопределение установок кодека MTA.

-->

```
<xs:element name="PcspEndpoint">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EndpointId" type="nonEmptyString"/>
      <xs:element name="AdminStatus">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
            <xs:enumeration value="2"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Protocol" type="protocolType"/>
      <xs:element name="Codec" type="codecType"/>
      <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
```

Объект PcspMta.

Адаптер медиатерминала объединяет один или несколько конечных пунктов (физически входящих в состав MTA).

Содержание...

**MtaFqdn** – Имя FQDN адаптера MTA, однозначно идентифицирующее этот MTA.

**MtaPort** – Порт прослушивания NCS в адаптере MTA (по умолчанию: 2427)

**CmtsFqdn** – Имя FQDN управляющей CMTS. Серверу CMS оно требуется для установления качества службы DQoS адаптера MTA с надлежащей системой CMTS.

Временная зона – зона, в которой физически размещен данный адаптер MTA, согласно цифровому формату временной зоны по RFC 1123.

Протокол – следует установить значение "MGCP 1.0 NCS 1.0". Это протокол по умолчанию для всех имеющихся конечных пунктов.

Кодек – по умолчанию для всех имеющихся конечных пунктов.

-->

```
<xs:element name="PcspMta">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="MtaFqdn" type="nonEmptyString"/>
      <xs:element name="ListenPort" type="xs:int"/>
      <xs:element name="CmtsFqdn" type="xs:string"/>
      <xs:element name="Timezone" type="timezoneType"/>
      <xs:element name="Protocol" type="protocolType"/>
      <xs:element name="Codec" type="codecType"/>
      <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

</xs:element>

<!--

Объект PcspService.

Содержание...

**ServiceId** – уникальный идентификатор этой службы.

**AdminStatus** –

0: приостановленное (т. е. счет не оплачен).

1: разрешенное (нормальное состояние).

2: номер изменился.

3: не работает.

4: не определено.

**BillingId** – телефонный номер, идентифицирующий другую службу, которой должен быть выставлен счет вместо данной службы.

**ExternalId** – произвольная цепочка, используемая для переноса таких данных, как идентификатор пользователя, и т. п.

**IsPrimary** – при нескольких службах, обеспечиваемых на конечном пункте, одна из служб ДОЛЖНА иметь такую установку флага для индикации службы по умолчанию, обрабатывающей исходящие вызовы.

ложно/0: эта служба не является основной.

истинно/1: эта служба является основной.

**PrimaryRing** – Идентификатор структуры основного вызова. Индекс в таблице каденций (последовательностей тонов) МТА, определяющий выбор звуковой структуры вызова для этой службы.

**DisplayName** – используется в функции доставки имени вызова (CNAM).

**DisplayNumber** – используется в функции доставки номера вызова (CND).

Пароль – различные функции вызова запрашивают пароль перед разрешением любых изменений.

Управление объявлениями в сети. См. предыдущее определение.

Межстанционные коды и установки переносимости местного номера. См. предыдущие определения.

Функции вызова. См. предыдущие определения.

-->

```
<xs:element name="PcspService">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element name="ServiceId" type="ServiceIdType"/>
```

```
<xs:element name="AdminStatus">
```

```
<xs:simpleType>
```

```
<xs:restriction base="xs:integer">
```

```
<xs:enumeration value="0"/>
```

```
<xs:enumeration value="1"/>
```

```
<xs:enumeration value="2"/>
```

```
<xs:enumeration value="3"/>
```

```
<xs:enumeration value="4"/>
```

```
</xs:restriction>
```

```
</xs:simpleType>
```

```
</xs:element>
```

```
<xs:element name="BillingId" type="ServiceIdType"/>
```

```
<xs:element name="ExternalId" type="xs:string"/>
```

```
<xs:element name="IsPrimary" type="xs:boolean"/>
```

```
<xs:element name="PrimaryRing" type="xs:string"/>
```



```

    <xs:element name="DisplayName" type="xs:string"/>
    <xs:element name="DisplayNumber" type="xs:string"/>
    <xs:element name="Password" type="xs:string"/>
    <xs:element ref="Announcements"/>
    <xs:element ref="InterExchange"/>
    <xs:element ref="LNP"/>
    <xs:element ref="ListOfCallFeatures"/>
    <xs:element ref="Extension" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<!--
  Формат файла Import/Export.
  Используется для переноса одного или нескольких объектов и отношений между
  серверами PS/CMS.

  ПРИМЕЧАНИЕ: не охватывает PcsPcMs. В настоящее время нет оснований, чтобы
  сервер получал свой собственный CMS-объект от сервера PS.
-->
<xs:element name="PcspImportExport">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="PcspService"/>
      <xs:element ref="PcspEndpoint"/>
      <xs:element ref="PcspMta"/>
      <xs:element ref="PcspRelation"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:schema>

```

## Приложение В

### Спецификация языка WSDL для передачи PCSP-сообщений

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
  Интерфейс обслуживания абонентов сервером CMS в сетях IPCablecom.
  Специфицируется на языке описания службы (WSDL) Web 1.1.
-->
<definitions name="PcspI01Service"
  targetNamespace="unique_fully_qualified_namespace"
  xmlns:tns="unique_fully_qualified_namespace"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <!-- ПРИМЕЧАНИЕ: при реализации параметр <unique_fully_qualified_namespace>
  ДОЛЖЕН быть заменен полностью определенным уникальным идентификатором.
-->
  <!--
    Раздел <types> определяет индивидуальные типы данных, требующихся для
    интерфейса.

    Для PCSPI01 требуются два индивидуальных типа данных:
      PcspArg (и их массив)
      PcspObj (и их массив).

    // PcspArg (псевдокод)
    //
    класс PcspArg

```

```

{
    // Имя объекта (EntityName) и ключ конкретного объекта.
    // Трафаретные символы в настоящее время не допускаются.
    // Ключ игнорируется для объекта PcspRelation.
    //
    Имя объекта цепочки;
    Ключ цепочки;

    // Резервируется для использования в будущем. В настоящее время
установлен на 0.
    //
    флаги int;
}

// PcspObj (псевдокод).
//
класс PcspObj
{
    // Имя объекта (EntityName) и ключ конкретного объекта.
    // Ключ игнорируется для объекта PcspRelation.
    //
    Имя объекта (entityName) последовательности;
    Ключ цепочки;

    // cmdStatus:
    // PcspObj как способ вывода/результата - ДОЛЖЕН быть установлен в один
из кодов состояния, определенных ниже.
    // PcspObj как вход для Put() - ДОЛЖЕН быть установлен на одно из
следующих значений:
    //     1, создать новый объект
    //     2, модифицировать существующий объект.
    // Это поле игнорируется для объекта PcspRelation.

    //
    int cmdStatus;

    // XML-кодирование для PCSP-схемы модели данных или 0 (нуль)
    //
    Цепочка xmlEncoding;
}

```

Имена объектов (EntityNames); ДОЛЖНО быть одно из следующих имен:

```

"PcspService"
"PcspMta"
"PcspEndpoint"
"PcspCms"
"PcspRelation"

```

Коды состояния: используются для вывода способа или содержатся в поле cmdStatus результата PcspObj (вывода).

```

0 , Операция успешно завершена
1 , Объект не обнаружен
2 , Задан неверный режим Put()
3 , Не удается создать объект, объект уже существует
4 , Не удается выполнить операцию считывания
5 , Не удается выполнить операцию создания
6 , Не удается выполнить операцию модификации
7 , Не удается выполнить операцию удаления
8 , Внутренняя проблема
9 , Сервер занят
10, Неподдерживаемая операция
11, Расширение фирмы-поставщика

```

...расширяется при необходимости...

-->

```

<types>
  <schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.IPCablecom.com/pcsp/i01">
    <complexType name="PcspObj">
      <sequence>
        <element name="entityName" type="string"/>
        <element name="key" type="string"/>
        <element name="cmdStatus" type="int"/>
        <element name="xmlEncoding" type="string"/>
      </sequence>
    </complexType>
    <complexType name="ArrayOfPcspObj">
      <complexContent>
        <restriction base="soapenc:Array">
          <attribute ref="soapenc:arrayType"
wsdl:arrayType="tns:PcspObj[]" />
        </restriction>
      </complexContent>
    </complexType>
    <complexType name="PcspArg">
      <sequence>
        <element name="entityName" type="string"/>
        <element name="key" type="string"/>
        <element name="flags" type="int"/>
      </sequence>
    </complexType>
    <complexType name="ArrayOfPcspArg">
      <complexContent>
        <restriction base="soapenc:Array">
          <attribute ref="soapenc:arrayType"
wsdl:arrayType="tns:PcspArg[]" />
        </restriction>
      </complexContent>
    </complexType>
  </schema>
</types>
<!--
  Раздел сообщения.

```

Активизация способа на интерфейсе включает в себя два "сообщения"...  
входное сообщение и выходное сообщение.

"In" содержит набор входных аргументов вызова метода.

"Out" содержит возвращаемые значения.

```

-->
<message name="Get0In">
  <part name="args" type="tns:ArrayOfPcspArg"/>
</message>
<message name="Get0Out">
  <part name="Result" type="tns:ArrayOfPcspObj"/>
</message>
<message name="Put1In">
  <part name="objs" type="tns:ArrayOfPcspObj"/>
</message>
<message name="Put1Out">
  <part name="Result" type="tns:ArrayOfPcspObj"/>
</message>
<message name="Delete2In">
  <part name="args" type="tns:ArrayOfPcspArg"/>
</message>
<message name="Delete2Out">
  <part name="Result" type="tns:ArrayOfPcspObj"/>
</message>
<message name="CmdStatus3In">
  <part name="isCmd" type="xsd:boolean"/>

```

```

    <part name="code" type="xsd:int"/>
    <part name="subCode" type="xsd:int"/>
    <part name="vendorExtension" type="xsd:string"/>
</message>
<message name="CmdStatus3Out">
    <part name="Result" type="xsd:int"/>
</message>
<!--
    Тип порта определяет интерфейс.

    Каждая "операция" является методом на интерфейсе со связанными входным и
    выходным сообщениями (аргументы и возвращаемые функцией значения).

    // Интерфейс службы PCSP (в псевдокоде).
    //
    интерфейс IPcspI01Service
    {
        // Получить (считать) один или несколько объектов с сервера.
        // Имя объекта (EntityName) "PcspRelation" не допускается (только
объекты)
        //
        PcspObj [] Получить аргументы (PcspArg []);

        // Ввести (записать) объекты и отношения в сервер.
        //
        PcspObj [] Получить объекты (PcspObj []);

        // Удалить объекты и отношения из сервера.
        //
        PcspObj [] Удалить аргументы (PcspArg []);

        // Внеполосная команда и отчет о состоянии.
        //
        // Заранее определенные коды команд:
        // 0 - команда расширения
        //
        // Заранее определенные коды состояний:
        // 0 - состояние расширения
        //
        int CmdStatust(булева команда, // истинно для CMD, ложно для СОСТОЯНИЯ
(STATUS)).

                код int, // код CMD или STATUS (см. выше).
                субкод int // Субкод. Дальнейшее уточнение кода.
                Расширение цепочки );
    }
-->
<portType name="PcspI01Service">
    <operation name="Get" parameterOrder="args">
        <input name="Get0In" message="tns:Get0In"/>
        <output name="Get0Out" message="tns:Get0Out"/>
    </operation>
    <operation name="Put" parameterOrder="objs">
        <input name="Put1In" message="tns:Put1In"/>
        <output name="Put1Out" message="tns:Put1Out"/>
    </operation>
    <operation name="Delete" parameterOrder="args">
        <input name="Delete2In" message="tns>Delete2In"/>
        <output name="Delete2Out" message="tns>Delete2Out"/>
    </operation>
    <operation name="CmdStatus" parameterOrder="isCmd code subCode
vendorExtension">
        <input name="CmdStatus3In" message="tns:CmdStatus3In"/>
        <output name="CmdStatus3Out" message="tns:CmdStatus3Out"/>
    </operation>
</portType>
<!--
    Связывает интерфейс ("portType") с особенностями транспортировки.

```

По существу, ввод и вывод потока для каждого метода ограничен как вызов с дистанционной процедурой, использующий протокол SOAP 1.1.

```
-->
<binding name="PcspI01Service" type="tns:PcspI01Service">
  <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="Get">
    <soap:operation soapAction="Get" style="rpc"/>
    <input name="Get0In">
      <soap:body use="encoded"
namespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output name="Get0Out">
      <soap:body use="encoded" amespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
  <operation name="Put">
    <soap:operation soapAction="Put" style="rpc"/>
    <input name="Put1In">
      <soap:body use="encoded"
namespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output name="Put1Out">
      <soap:body use="encoded"
namespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
  <operation name="Delete">
    <soap:operation soapAction="Delete" style="rpc"/>
    <input name="Delete2In">
      <soap:body use="encoded"
namespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output name="Delete2Out">
      <soap:body use="encoded"
namespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
  <operation name="CmdStatus">
    <soap:operation soapAction="CmdStatus" style="rpc"/>
    <input name="CmdStatus3In">
      <soap:body use="encoded"
namespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output name="CmdStatus3Out">
      <soap:body use="encoded"
namespace="http://www.IPCablecom.com/pcsp/i01"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
<!--
```

Определение верхнего уровня службы PCSP I01.

Заметим, что элемент `<service>` не содержит адреса. Предполагается, что фактический адрес службы в явной форме устанавливается между клиентом и сервером.

```
-->
<service name="PcspI01Service">
  <documentation>IPCablecom CMS Subscriber Provisioning Service
I01</documentation>
  <port name="PcspI01Service" binding="tns:PcspI01Service">
    <soap:address location=""/>
  </port>
</service>
</definitions>
```

## Добавление I

### Примеры кодирования объектов

При описании следующих примеров предполагается, что параметр `<пространство уникальных полностью определенных имен>`, определенный в Приложении А, размещен на `"http://www.cablelabs.com/Pcsp/I01/schema"`.

#### I.1 Пример объекта `PcspService`

```
<?xml version="1.0" encoding="UTF-8"?>
<!--отредактировано с помощью XML Spy v4.0.1 U (http://www.xmlspy.com) Полем
Дюффи (Paul Duffy) (частное сообщение) -->
<!--
```

Пример кодировки объекта службы.

Пространство имен по умолчанию и пространство имен `"pcsp"` установлено на `PcspI01`.

Пространство имен `"pcsp"` удобно и позволяет расширениям фирмы-поставщика ссылаться на элементы из основной схемы PCSP.

```
-->
<PcspService xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="PcspI01.xsd">
<!--
```

Пример объекта службы.

```
-->
<ServiceId format="NSN">9785551212</ServiceId>
<AdminStatus>1</AdminStatus>
<BillingId>9785550000</BillingId>
<ExternalId>0123456789</ExternalId>
<IsPrimary>true</IsPrimary>
<PrimaryRing>IndexIntoCadenceTable</PrimaryRing>
<DisplayName>John Q Public</DisplayName>
<DisplayNumber>(978) - 555-1212</DisplayNumber>
<Password>45hjg3j6gkg6h54j6gkj3g6</Password>
<Announcements>
  <Language>EN</Language>
  <Timezone>+0500</Timezone>
</Announcements>
<InterExchange>
  <PIC>0123</PIC>
  <LPIC>0123</LPIC>
  <IPIC>0123</IPIC>
</InterExchange>
<LNP>
```

```

    <PortingStatus>0</PortingStatus>
    <LNPT>0</LNPT>
</LNP>
<ListOfCallFeatures>
  <CfCND>
    <Subscribed>>true</Subscribed>
    <AdminStatus>1</AdminStatus>
  </CfCND>
  <CfCIDB>
    <Subscribed>0</Subscribed>
    <AdminStatus>1</AdminStatus>
    <UserParams>
      <Flag>PUBLIC</Flag>
    </UserParams>
  </CfCIDB>
  <CfCFBL>
    <Subscribed>true</Subscribed>
    <AdminStatus>1</AdminStatus>
    <UserParams>
      <Active>true</Active>
      <Always>0</Always>
      <ForwardTo>9785551212</ForwardTo>
    </UserParams>
  </CfCFBL>
  <CfSPCALL>
    <Subscribed>0</Subscribed>
    <AdminStatus>1</AdminStatus>
    <UserParams>
      <ListOfSpeedDial>
        <SdPair>
          <SdNum>1</SdNum>
          <ServiceId>9785551212</ServiceId>
        </SdPair>
        <SdPair>
          <SdNum>3</SdNum>
          <ServiceId>9785551000</ServiceId>
        </SdPair>
      </ListOfSpeedDial>
    </UserParams>
  </CfSPCALL>
  <CfRDA>
    <Subscribed>1</Subscribed>
    <AdminStatus>1</AdminStatus>
  </CfRDA>
  <CfLSR>
    <Subscribed>1</Subscribed>
    <AdminStatus>1</AdminStatus>
    <UserParams>
      <BlkDomLongDist>1</BlkDomLongDist>
      <BlkIntLongDist>1</BlkIntLongDist>
      <BlkPayPerCall>1</BlkPayPerCall>
      <BlkOperatorAssist>1</BlkOperatorAssist>
      <BlkDirAssist>1</BlkDirAssist>
      <BlkTollFree>1</BlkTollFree>
      <ListOfServiceId>
        <ServiceId>9895551001</ServiceId>
        <ServiceId>9895551002</ServiceId>
        <ServiceId>9895551003</ServiceId>
      </ListOfServiceId>
    </UserParams>
  </CfLSR>
  <CfDND>
    <Subscribed>1</Subscribed>
    <AdminStatus>1</AdminStatus>

```

```

    <UserParams>
      <Active>true</Active>
      <WdStartTod1>00:00:00+05:00</WdStartTod1>
      <WdStopTod1>06:00:00+05:00</WdStopTod1>
      <WdStartTod2>18:00:00+05:00</WdStartTod2>
      <WdStopTod2>20:00:00+05:00</WdStopTod2>
      <WeStartTod1>00:00:00+05:00</WeStartTod1>
      <WeStopTod1>09:00:00+05:00</WeStopTod1>
      <WeStartTod2>18:00:00+05:00</WeStartTod2>
      <WeStopTod2>20:00:00+05:00</WeStopTod2>
    </UserParams>
  </CfDND>
  <CfCOC>
    <Subscribed>1</Subscribed>
    <AdminStatus>1</AdminStatus>
    <UserParams>
      <Active>true</Active>
      <StartTod>00:00:00+05:00</StartTod>
      <StopTod>06:00:00+05:00</StopTod>
      <ListOfServiceId>
        <ServiceId>9895551001</ServiceId>
        <ServiceId>9895551002</ServiceId>
        <ServiceId>9895551003</ServiceId>
      </ListOfServiceId>
    </UserParams>
  </CfCOC>
  <CfNSA>
    <Subscribed>1</Subscribed>
    <AdminStatus>1</AdminStatus>
    <UserParams>
      <Active>true</Active>
      <StartTod>00:00:00+05:00</StartTod>
      <StopTod>06:00:00+05:00</StopTod>
      <ListOfServiceId>
        <ServiceId>9895551001</ServiceId>
        <ServiceId>9895551002</ServiceId>
        <ServiceId>9895551003</ServiceId>
      </ListOfServiceId>
    </UserParams>
  </CfNSA>
</ListOfCallFeatures>
</PcspService>

```

## I.2 Пример объекта PcspEndpoint

```

<?xml version="1.0" encoding="UTF-8"?>
<PcspEndpoint xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!--
    Пример объекта конечного пункта.
  -->
  <EndpointId>aaln/1@mta01.cablelabs.com</EndpointId>
  <AdminStatus>2</AdminStatus>
  <Protocol>MtaDefault</Protocol>
  <Codec>2</Codec>
</PcspEndpoint>

```

## I.3 Пример объекта PcspMta

```

<?xml version="1.0" encoding="UTF-8"?>
<!--отредактировано с помощью XML Spy v4.0.1 U (http://www.xmlspy.com) Полем
Дюффи (Paul Duffy) (частное сообщение) -->

```



```

<PcspMta xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="PcspI01.xsd">
  <!--
    Пример объекта адаптера МТА.
  -->
  <MtaFqdn>mta01.cablelabs.com</MtaFqdn>
  <ListenPort>2427</ListenPort>
  <CmtsFqdn>cmta01.cablelabs.com</CmtsFqdn>
  <Timezone>-0500</Timezone>
  <Protocol>MCGP 1.0 NCS 1.0</Protocol>
  <Codec>5</Codec>
</PcspMta>

```

#### I.4 Пример объекта PcspCms

```

<?xml version="1.0" encoding="UTF-8"?>
<PcspCms xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!--
    Объект CMS.

    Еще недостаточно определен...только его ключ.
    Используется как набор для служб и конечных пунктов.
  -->
  <CmsFqdn>cma01.cablelabs.com</CmsFqdn>
</PcspCms>

```

#### I.5 Пример PcspRelation

```

<?xml version="1.0" encoding="UTF-8"?>
<PcspRelation xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.cablelabs.com/Pcsp/I01/schema PcspI01.xsd"
relOp="add">
  <!--
    Отношение PcspRelation.

    Это отношение ассоциирует несколько конечных пунктов со службой
    "9785551212".
  -->
  <Class1>PcspService</Class1>
  <Key>9785551212</Key>
  <Class2>PcspEndpoint</Class2>
  <ListOfKeys>
    <Key>aaln/1@mta01.cablelabs.com</Key>
    <Key>aaln/1@mta02.cablelabs.com</Key>
    <Key>aaln/1@mta03.cablelabs.com</Key>
    <Key>aaln/1@mta04.cablelabs.com</Key>
  </ListOfKeys>
</PcspRelation>

```

## Добавление II

### Примеры расширения объектов

При описании следующих примеров предполагается, что параметр <пространство уникальных полностью определенных имен>, определенное в Приложении А, размещено на "http://www.cablelabs.com/Pcsp/I01/schema".

#### II.1 Пример расширенного объекта PcspService

В следующем примере показаны возможности расширения схемы PCSP. В данном примере объект PcspService расширяется посредством новой функции вызова и нескольких новых элементов на основном теле объекта.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

Пример показывает, как расширить объект Pcsp.

В этом примере объект PcspService расширяется с помощью дополнительных полей и функций вызова.

Подробное описание см. ниже.

```
-->
```

```
<PcspService xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema">
```

```
<!--
```

Основное тело объекта службы заполняется примерными данными, которые позволяют подтвердить правильность объекта.

```
-->
```

```
<ServiceId>5551212</ServiceId>
```

```
<AdminStatus>0</AdminStatus>
```

```
<BillingId>5551212</BillingId>
```

```
<ExternalId>5551212</ExternalId>
```

```
<IsPrimary>true</IsPrimary>
```

```
<PrimaryRing/>
```

```
<DisplayName/>
```

```
<DisplayNumber/>
```

```
<Password/>
```

```
<Announcements>
```

```
  <Language>EN</Language>
```

```
  <Timezone>+0500</Timezone>
```

```
</Announcements>
```

```
<InterExchange>
```

```
  <PIC>0</PIC>
```

```
  <LPIC>0</LPIC>
```

```
  <IPIC>0</IPIC>
```

```
</InterExchange>
```

```
<LNP>
```

```
  <PortingStatus>1</PortingStatus>
```

```
  <LNPT>true</LNPT>
```

```
</LNP>
```

```
<!--
```

Объект службы допускает два вида расширения:

1. В основном теле объекта.
2. В списке функций вызова.

Здесь мы расширяем набор функций вызова за счет функции вызова CfXYZ.

1. Элемент VendorExt должен задать действующее пространство имен для схемы расширения. Это позволяет системе синтаксического анализа разместить файл схемы для расширения.

2. Любое содержимое элемента VendorExt должно быть упорядочено в пространстве имен, позволяя проверить правильность схемы расширения.

```
-->
<ListOfCallFeatures>
  <Extension xmlns:ext="http://www.cablelabs.com/SampleExtension">
    <ext:CfXYZ>
      <ext:Subscribed>true</ext:Subscribed>
      <ext:Enabled>true</ext:Enabled>
    </ext:CfXYZ>
  </Extension>
</ListOfCallFeatures>
<!--
  Здесь мы расширяем содержание данных основного тела объекта службы.
-->
<Extension xmlns:ext="http://www.cablelabs.com/SampleExtension">
  <ext:A>Sample extension A</ext:A>
  <ext:B>Sample extension B</ext:B>
  <ext:C>Sample extension C</ext:C>
</Extension>
</PcspService>
```

## II.2 Схема расширения

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

Схема для данного примера расширения службы PcspService.

Эта схема определяет несколько расширений:

A, B и C для основного тела объекта службы.

Функция вызова CfXYZ для списка функций вызова службы.

```
-->
<xs:schema targetNamespace="http://www.cablelabs.com/SampleExtension"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.cablelabs.com/SampleExtension" elementFormDefault="qualified">
  <xs:element name="A" type="xs:string"/>
  <xs:element name="B" type="xs:string"/>
  <xs:element name="C" type="xs:string"/>
  <xs:element name="CfXYZ">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Subscribed" type="xs:boolean"/>
        <xs:element name="Enabled" type="xs:boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## Добавление III

### Оценка кодирования данных

Рассматриваются варианты кодирования объектов данных и передачи сообщений:

#### III.1 Язык XML

Язык XML является стандартным метаязыком, который позволяет организациям разрабатывать собственные языки разметки для публикации документов и для обмена данными. Такие разметки базируются на текстах: они разрабатываются так, чтобы быть понятными и людям, и процессам. Язык XML предлагает:

- открытый, основанный на стандартах, независимый от платформ обмен данными;
- стандартизованные синтаксические анализаторы для ввода данных в память;
- стандартизованные интерфейсы (древовидно-ориентированные и поточно-ориентированные) для обработки данных;
- стандартизованные пути отображения данных;
- стандартизованные пути запроса данных;
- стандартизованные пути для компоновки данных;
- стандартизованное обучение персонала для издательских систем и обработки данных.

Затраты: немного повышенный объем кодировки и увеличенные служебные расходы на синтаксический анализ.

Спецификация языка XML контролируется рабочей группой по языку XML Консорциума системы World Wide Web (W3C). Свои вклады вносят Группы экспертов по отдельным вопросам. Этот стандарт является общедоступным и не принадлежит какой-либо компании. Спецификация v1.0 была принята Консорциумом W3C в качестве Рекомендации 10 февраля 1998 года. Эту спецификацию можно найти в <http://www.w3.org/TR/REC-xml>.

#### III.2 Язык ASN.1/BER

ASN.1 является базирующимся на текстах, независимым от платформ синтаксисом, используемым для отображения произвольных структур данных. Он широко используется в представлениях базы управляющей информации (MIB) протокола SNMP. Основные правила кодировки (BER) сводятся к простому рекурсивному алгоритму, образующему компактные октетные кодировки из ASN.1-описаний. BER кодирует каждый объект как тег, указывающий тип объекта, длину объекта, описывающую его размер, и значение, содержащее реальное содержание объекта.

См. таблицу III.1.

#### III.3 Патентованный код ASCII

Патентованное кодирование выходит за рамки данного документа.

#### III.4 SDP (протокол описания сеансов)

Не отличается гибкостью в отношении содержания. Первично использовался для описания возможностей потоковой передачи медиаинформации.

#### III.5 Служба RADIUS

Кодирование данных (TLV) на основе RADIUS слишком примитивно и не обеспечивает возможности упорядочения. RADIUS сложен и является ограниченным в отношении структур кодов.

#### III.6 Язык структурированных запросов SQL

Связан с реализацией/схемой конкретных реляционных баз данных. Некоторые фирмы-поставщики уже применяют базы данных с несовместимой схемой.

### III.7 Сводные данные вариантов

Таблица III.1/J.177 – Варианты кодирования данных

Вариант	За	Против
Язык XML	<p>Гибкий, основанный на тегах ASCII.</p> <p>Обеспечивает проверку синтаксиса при использовании схемы /DTD.</p> <p>Легко расширяется, не влияет на транспортировку.</p> <p>Допускает расширение фирм-поставщиков.</p> <p>Независим от платформ.</p> <p>Независим от языков.</p>	<p>Не защищен, требует какого-то уровня защиты транспортировки.</p> <p>Больше загружает центральный процессор и сети, чем при двоичном кодировании. Требует времени на синтаксический анализ и т. п.</p>
Язык ASN.1/BER	<p>Иерархическая структура для форматирования данных. Определяет язык для описания форматов данных (или "схемы").</p> <p>Могут быть вложенные структуры данных.</p> <p>Данные форматируются независимо от платформы.</p> <p>Формат может расширяться, ЕСЛИ в проекте предусматривается создание версий формата, с тем чтобы для приложений был известен формат, необходимый для применения синтаксического анализа содержания данных.</p>	<p>Сложности при расширении ASN.1.</p> <p>Могут быть трудности в отношении разработки и реализации обратной совместимости версий формата.</p> <p>В большинстве реализаций используются компилированные синтаксические анализаторы двоичного уровня для каждой схемы, что означает сложности при определении гибких приложений.</p> <p>Могут быть трудности в отладочных приложениях и в отношении интероперабельности, поскольку небольшая ошибка форматирования вызывает невозможность анализа/считывания "пакета" данных.</p>
Патентованный код ASCII	Вне области применения	Вне области применения
SDP (Описание сеанса)		<p>Негибок в отношении содержания.</p> <p>Использовался в первую очередь для описания возможностей потоковой медиатрансляции.</p>
RADIUS		<p>Кодирование RADIUS для данных (TLV) слишком примитивно.</p> <p>Не допускает принудительного упорядочения.</p> <p>Сложен и ограничен в отношении структур кодов.</p>
SQL		Связан с реализацией схемы реляционных баз данных, которые могут не использоваться рядом поставщиков.

### III.8 Рекомендация: язык XML

Язык XML обеспечивает безразличную к платформе и нейтральную к технологии форму структуризации сообщений и упаковки данных. Он весьма подходит для передачи данных между разнотипными приложениями без необходимости того, чтобы каждое приложение имело сведения о специфическом формате другого приложения. Поскольку XML является структурированным языком, он хорошо подходит для иерархических типов сообщений. Данные можно легко отображать в элементы, поэтому XML-документ (будучи древовидной структурой) обеспечивает поддержку иерархии. Издержки: повышенная загрузка линий и увеличенное время сортировки (синтаксического анализа) для объектов.

## Добавление IV

### Оценка протоколов транспортировки

#### IV.1 Протокол TFTP с защитой IPSec

Протокол TFTP уже используется в инфраструктуре системы IP-Cablecom (DOCSIS). Он является облегченным протоколом передачи файла.

#### IV.2 Пакетная служба RADIUS – Несколько записей в одном запросе посредством передачи сообщений о событиях

Служба RADIUS – это стандарт группы IETF, первоначально созданный для обработки кодовых вызовов Интернет с целью аутентификации, авторизации и калькуляции. В настоящее время RADIUS является стандартом де-факто, используемым для этих целей большинством производителей маршрутизаторов. Некоторые фирмы – поставщики аппаратуры телефонных шлюзов уже используют RADIUS для поддержки расширений фирм-поставщиков, чтобы обеспечить доставку информации, необходимой для тарификации.

RADIUS определяет как транспортный протокол, так и спецификацию форматов сообщений. Как транспортный протокол, RADIUS исходит из протокола датаграмм пользователя (UDP) для вещательной передачи сообщений и базируется на портах.

Как формат сообщений, протокол форматирует данные, используя тег-длину-значение (для них используется также название атрибут-длина-значение). Требуется минимальное количество определяемых заранее тегов аутентификации, авторизации и калькуляции. Тем не менее можно добавлять новые атрибуты, не создавая проблем для существующих реализаций протокола. RADIUS имеет минимальную общую длину сообщения 20 знаков и максимальную длину 4096 знаков. Отдельные поля данных поддерживают 247 байтов данных, например 247-значный URL и имя файла.

RADIUS имеет очень плохие характеристики надежности и по существу не имеет средств коррекции ошибок; он весьма ограничен относительно новых тегов (позволяет определить всего 255 дескрипторов) (ср. с 600 имеющимися функциями у некоторых коммутаторов класса 5 для коммутируемых телефонных сетей PSTN).

#### IV.3 Протокол Diameter

Diameter является результатом деятельности группы IETF, спроектированным так, чтобы обеспечить обратную совместимость с RADIUS. Он допускает гораздо большие расширения, обладает существенными преимуществами по защите и разработан, чтобы минимизировать конфигурацию. Кроме того, он очень хорошо поддерживает междоменные AAA (аутентификацию, авторизацию и калькуляцию), допуская разнообразные схемы защиты, в том числе открытые ключи и т. п. Diameter поддерживает завершение сбоя резервного сервера (он разработан для условий низкой частоты отказов (99,99+)). См. Базовый протокол Diameter, выпущенный IETF RFC [18].

Протоколы RADIUS/DIAMETER не обеспечивают двунаправленную связь (обеспечивают только подтверждения) и поэтому не удовлетворяют требованиям.

#### IV.4 Системы распределенного объекта

##### IV.4.1 Протокол CORBA/ПОР

Технология распределенных объектов поддерживается Рабочей группой по управлению объектами (OMG). К этой технологии присоединилось до 800 членов OMG.

Обобщенная архитектура посредника объектных запросов (CORBA) позволяет приложениям устанавливать связи с другими приложениями, используя посредник объектных запросов (ORB). ORB является промежуточным программным обеспечением, которое устанавливает между объектами отношения клиент–сервер. ORB регистрирует клиентов и управляет такими правами, как право на публикацию в сети, абонирование и прием. Используя ORB, клиент может прозрачно активизировать метод на серверном объекте, который может существовать на том же компьютере или в сети. ORB перехватывает местный вызов и отвечает за отыскание объекта, способного выполнить запрос, передать его параметры, активизировать метод и вернуть результаты.

Язык описания интерфейсов (IDL) используется для установления контрактов по протоколу ORB между объектами клиента и сервера. ORB по существу скрывает детали транспортировки от программиста. IDL компилируется в C++, Java и т. д., в реализации клиентских и серверных "заглушек", выполняющих все рутинные операции кодирования и декодирования данных, которые требуются для транспортного протокола ПОР, используемого между клиентами и серверами.

CORBA берет на себя детали поиска сервера для вызова метода, транспортировку аргументов от компьютера клиента на компьютер сервера и транспортировку возврата любого кода на компьютер клиента.

В настоящее время протокол ORB поставляется многими фирмами для более чем трех десятков аппаратных платформ и операционных систем. Архитектура CORBA особенно популярна на платформах Unix. Однако на практике поставщики ORB конкурируют по обеспечиваемым характеристикам. Постоянно возникает проблема интероперабельности, когда не принимаются во внимание основные требования (по защите и т. п.). Вероятность того, что два случайно выбранных ORB смогут успешно установить между собой связь, мала. Что касается развития, то CORBA становится все более сложной. Кроме того, CORBA является довольно дорогим изделием (в отношении лицензий на использование и проектирование).

#### **IV.4.2 Модель DCOM**

Это распределенная модель компонентных объектов фирмы Microsoft. Имеет следующие общие с CORBA характеристики:

- Разделяет интерфейс объекта и его реализацию. Это выполняется с использованием языка MIDL (варианта IDL фирмы Microsoft).
- Допускает прозрачность размещения. Клиенты активизируют методы на удаленных объектах, не зная, какой компьютер на дистанционном объекте будет активизирован.
- Однородная схема обработки исключений (вызовы метода DCOM возвращают однородные состояния HRESULT).

Однако:

- DCOM основывается на протоколе транспортировки DCE ORPC, который несовместим с протоколом транспортировки ПОР, используемым CORBA.
- DCOM по существу является технологией только фирмы Microsoft. Он является стандартным для платформ Win95, Win98 и NT.

#### **IV.5 Протокол HTTP**

Гипертекстовый протокол передачи HTTP является основополагающим протоколом для "Всемирной паутины" World Wide Web. HTTP был разработан для объединения источников неоднородных данных, чтобы создать распределенную информационную систему. При его разработке учитывалась перспектива расширения. В типичной HTTP-транзакции :

- 1) клиент устанавливает соединение с сервером;
- 2) клиент передает запрос на сервер (с параметрами URL);
- 3) сервер передает отклик, содержащий состояние и запрошенный URL;
- 4) любая сторона может прервать соединение.

В HTTP предусматриваются заголовки транзакции как для запросов клиентов, так и для откликов серверов. Заголовок клиентской транзакции может содержать параметры, помогающие доставлять необходимую информацию клиенту (например, тип формата данных, язык и т. п.). Заголовок серверной транзакции может содержать параметры, содержащие информацию об отклике, например состояние отклика (код возврата), длину передаваемых данных, тип содержания, язык содержания и т. п.

При нынешнем состоянии "Всемирной паутины" протокол HTTP является всеохватывающим. К тому же он обеспечивает возможность введения сетевых устройств защиты (брандмауэров).

## IV.6 Сводные данные вариантов

Таблица IV.1/J.177 – Варианты транспортировки

Вариант	За	Против
TFTP с защитой IPSec	Легкость применения. Уже реализован для DOCSIS.	Не обеспечивает двустороннюю связь.
RADIUS	Гибкость: содержит поля для конкретных фирм-производителей и пользователей. Поля в этом пространстве могут определяться сетью IPCablecom. Используется многими системами ведения учета в IP-телефонии. Уже широко используются на таких компонентах IP, как маршрутизаторы.	Не все продукты RADIUS поддерживают AAA. Проблемы надежности должны решаться на прикладном уровне. Встроенная защита неадекватна, требуется независимый надежный протокол или ключи шифрования, используемые совместно с пограничными маршрутизаторами.
Основанный на интерфейсных разъемах фирменный протокол с SSL		Фирменный протокол. Если его использовать, то, вероятно, покончим с записью большинства из того, что в настоящее время доступно в протоколах SOAP или XMLP.
Протокол CORBA/IIOP	Легко реализуем, детализирует разрешение имени, упаковка параметров в сообщения и транспортировка управляются инфраструктурой CORBA.	Имеет тенденцию к удорожанию. Инфраструктура CORBA должна либо разрабатываться, либо приобретаться, но в любом случае должна использоваться вместе с приложением. Все еще остаются проблемы, касающиеся интероперабельности различных продуктов CORBA/ORB.
Протокол DCOM	DCOM и CORBA/IIOP основаны на схожих технологиях.	По существу это вариант только Microsoft Windows. Несовместим с CORBA-IIOP.
Протокол HTTP	HTTP уже широко используется как базовый протокол транспортировки для обмена между источниками данных (Web-серверами) и потребителями данных (браузерами клиентов). Передача данных основана на простых транзакциях. Простой, никому не принадлежащий, основанный на ASCII-текстах протокол. Обеспечивает простой обмен данными через большинство используемых в настоящее время сетевых инфраструктур (брандмауэры и т. п.). Клиент может использовать простые вызовы методов при выполнении таких запросов, как ПОЛУЧИТЬ, ОТПРАВИТЬ СООБЩЕНИЕ, ОЗАГЛАВИТЬ, ВВЕСТИ, УДАЛИТЬ, УСТАНОВИТЬ СВЯЗЬ и ОТМЕНИТЬ СВЯЗЬ (GET, POST, HEAD, PUT, DELETE, LINK и UNLINK).	Будучи никому не принадлежащим, этот протокол не запоминает транзакцию после ее завершения. Может ли он обеспечивать требуемые скорости транзакций на серверах CMS? Относительно дорог в отношении цифровых потоков и требований по обработке.



#### IV.7 Рекомендация: протокол HTTP 1.1

### Добавление V

#### Литература

- [12] *Basic Residential Feature Descriptions for IPCablecom-Based VOIP Services*, PKT-TR-VoIPBRF-R01-000608, Cable Television Laboratories, Inc., June 8, 2000.
- [13] *Enhanced Residential Feature Descriptions for IPCablecom-Based VOIP Services*, PKT-TR-VoIPERF-R01-000831, Cable Television Laboratories, Inc., August 31, 2000.
- [14] *IPCablecom Electronic Surveillance Call Flows Technical Report*, PKT\_TR-ESCF-V01-991229, Cable Television Laboratories, Inc., December 29, 1999.
- [15] IETF RFC 2139 (1997), *RADIUS Accounting*.
- [16] COMMON LANGUAGE® General Codes, *Telecommunications Service Providers IAC Codes, Exchange Carrier Names, Company Codes – Telcordia and Region Number: BR-751-100-112*. Issue 4: April 2002.
- [17] *Telcordia Local Exchange Routing Guide (LERG)*, Telcordia Technologies, Inc.
- [18] IETF Internet Draft, *Diameter Base Protocol*, <http://www.ietf.org/internet-drafts/draft-ietf-aaa-diameter-17.txt>.





## СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия В	Средства выражения: определения, символы, классификация
Серия С	Общая статистика электросвязи
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и средства передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
<b>Серия J</b>	<b>Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов</b>
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	TMN и техническое обслуживание сетей: международные системы передачи, телефонные, телеграфные, факсимильные и арендованные каналы
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных и взаимосвязь открытых систем
Серия Y	Глобальная информационная инфраструктура и аспекты межсетевых протоколов (IP)
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи