

国际电信联盟

ITU-T

国际电信联盟
电信标准化部门

J.177

(11/2005)

J系列：有线网和电视、声音节目和其它多媒体信号的传输
IPCablecom

IPCablecom CMS用户配置规范

ITU-T J.177建议书

ITU-T J.177建议书

IPCablecom CMS用户配置规范

摘要

本建议书为交换业务配置信息定义了呼叫管理服务器（CMS）与配置服务器（PS）之间的接口。接口采用了网络业务模式。该接口使用网络业务语言1.1版（WSDL 1.1），通过HTTP 1.1传输，并在简单对象接入协议1.1版（SOAP 1.1）编码的消息内传送XML编码对象。本接口利用IPSec确保安全性。

来源

ITU-T第9研究组（2005-2008年）按照ITU-T A.8建议书规定的程序，于2005年11月29日批准了ITU-T J.177建议书。

前 言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定 ITU-T 各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA 第 1 号决议规定了批准建议书须遵循的程序。

属 ITU-T 研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性条款（以确保例如互操作性或适用性等），只有满足所有强制性条款的规定，才能达到遵守建议书的目的。“应该”或“必须”等其它一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其它机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联尚未收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：<http://www.itu.int/ITU-T/ipr/>。

© 国际电联2006

版权所有。未经国际电联事先书面许可，不得以任何手段复制本出版物的任何部分。

目录

页码

1	范围	1
1.1	本建议书的目的	1
1.2	本建议书的范围	1
2	参考文献	1
2.1	规范性参考文献	1
2.2	资料性参考文献	2
2.3	参考文献的出处	2
3	术语和定义	2
4	缩写词、首字母缩写词和用词说明.....	2
4.1	缩写词和首字母缩写词	2
4.2	用词说明	4
5	背景	4
5.1	业务目标	4
5.2	IPcablecom 参考结构	4
5.3	组件和接口	5
5.4	组件	6
5.5	接口说明	6
6	假设	6
7	用户配置	7
7.1	客户记录（计费）	7
7.2	设备安装与配置	7
7.3	静态与动态用户配置数据	7
8	要求	8
8.1	一般性要求	8
8.2	传输要求	8
9	数据模型	9
9.1	概述	9
9.2	利用 PcsRelation entity.XML 编码方式对关系进行编码.....	11
10	消息	12
10.1	概述	12
10.2	对CMS 和 PS 消息功能的要求	12
10.3	WSDL 规范	13
11	安全性	13
	附件A – PCSP XML 模式	14
	附录I – 实体编码的范例	34
	I.1 对象扩展的范例	34
	I.2 PcsEndpoint 对象的范例	35

	页码
I.3 PcspMta对象的范例	36
I.4 PcspCms对象的范例	36
I.5 PcspRelation对象的范例	36
附录II – 对象扩展的范例	37
II.1 扩展 PcspService object 对象的范例	37
II.2 扩展方案	38
附录 III – 数据编码评估	38
III.1 XML	38
III.2 ASN.1/BER	39
III.3 专用 ASCII	39
III.4 SDP (会话描述协议)	39
III.5 RADIUS	39
III.6 SQL	39
III.7 可选方案一览表	39
III.8 建议: XML	40
附录IV – 传输协议评估	40
IV.1 具有IPSec的TFTP	40
IV.2 批处理 RADIUS – 通过事件消息在单一的请求中做多个记录	40
IV.3 Diameter	41
IV.4 分布式对象系统	41
IV.5 HTTP	42
IV.6 可选方案摘要	42
IV.7 建议: HTTP 1.1	43
参考书目	44

ITU-T J.177建议书

IPCablecom CMS用户配置规范

1 范围

1.1 本建议书的目的

本建议书为交换业务配置信息定义了呼叫管理服务器（CMS）与配置服务器（PS）之间的接口。接口采用了网络业务模式。该接口使用网络业务语言1.1版（WSDL 1.1），通过HTTP 1.1传输，并在简单对象接入协议1.1版（SOAP 1.1）编码的消息内传送XML编码对象。本接口利用IPSec确保安全性。

此接口传输的数据模式被特意设计成可扩展的模式，允许加入还未定义的IPCablecom功能和特定的厂商扩展功能。

1.2 本建议书的范围

本建议书的范围仅限于单一业务提供商的IPCablecom CMS配置。此外：

- CMS配置接口仅限于交换CMS与PS之间的业务激活数据。PS与后台（back-office）操作支持系统（OSS）之间的接口不在此范围之内。
- CMS元素管理和网元配置（拨号规划等）不在此范围之内。
- 客户记录创建/计费被认为是后台OSS应用的一部分，目前也不在此范围之内。

2 参考文献

2.1 规范性参考文献

下列ITU-T建议书和其他参考文献的条款，在本建议书中的引用而构成本建议书的条款。在出版时，所指出的版本是有效的。所有的建议书和其它参考文献均会得到修订，本建议书的使用者应查证是否有可能使用下列建议书或其它参考文献的最新版本。当前有效的ITU-T建议书清单定期出版。本建议书引用的文件自成一体时不具备建议书的地位。

- ITU-T Recommendation J.161 (2001), *Audio codec requirements for the provision of bidirectional audio service over cable television networks using cable modems.*
- ITU-T Recommendation J.162 (2005), *Network call signalling protocol for the delivery of time-critical services over cable television networks using cable modems.*
- ITU-T Recommendation J.170 (2005), *IPCablecom security specification.*
- IETF RFC 1123 (1989), *Requirements for Internet Hosts – Application and Support.*
- XML Protocol. <http://www.w3.org/2000/xp>.

2.2 资料性参考文献

- ITU-T Recommendation J.160 (2005), *Architectural framework for the delivery of time-critical services over cable television networks using cable modems.*
- ITU-T Recommendation J.167 (2005), *Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems.*
- IETF RFC 3588 (2003), *Diameter Base Protocol.*
- Simple Object Access Protocol. <http://www.w3.org/TR/SOAP>.
- Web Services Description Language. <http://www.w3.org/TR/wsdl>.

2.3 参考文献的出处

IETF RFC:

- 互联网工程任务组 (IETF) 秘书处转交美国国家研究推进机构, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, 电话: 703-620-8990, 传真: 703-620-9071, 互联网: www.ietf.org/。

3 术语和定义

本建议书定义了如下术语:

3.1 有效 (active): 当业务流被允许前转数据包时, 该业务流可称为“有效”。业务流在进入有效状态之前必须先获得批准。

3.2 端点 (endpoint): 终端、网关或多点会议单元 (MCU)。

3.3 互联网密钥交换 (Internet key exchange): 用于协商和为IPSec中的SA获取密钥的密钥交换机制。

3.4 本地号码可携带性 (local number portability): 在客户从一个本地业务提供商转用另一本地业务提供商时, 允许其保留原号码。

3.5 媒体网关 (media gateway): 为PSTN提供承载电路接口并对媒体流进行变码。

3.6 预共享密钥 (pre-shared key): 使用非特定手动或带外机制, 为通信流中的双方提供共享的密钥。

3.7 注册、准入和状态 (registration, admissions and status): RAS信道是一种不可靠的信道, 用于在H.323实体间传递RAS消息和带宽变化信息。

4 缩写词、首字母缩写词和用词说明

4.1 缩写词和首字母缩写词

本建议书使用了下述缩写词:

AAA	鉴权、授权、计费 (Authentication, Authorization and Accounting)
BPP	基本POTS配置 (Basic POTS Provisioning)
CFP	呼叫功能配置 (Call Feature Provisioning)
CID	电路ID (Circuit ID)
CM	DOCSIS电缆调制解调器 (DOCSIS Cable Modem)
CMS	加密消息语法 (Cryptographic Message Syntax)

CMS	呼叫管理服务器 (Call Management Server)
CMTS	电缆调制解调器终端系统 (Cable Modem Termination System)
Codec	编解码器 (COder-DECoder)
CSR	客户业务代表 (Customer Service Representative)
DOCSIS™	电缆数据传输业务接口规范 (Data-Over-Cable Service Interface Specifications)
DQoS	动态服务质量 (Dynamic Quality of Service)
ESP	IPsec 封装安全性有效负荷 (IPsec Encapsulation Security Payload)
FQDN	完全限定域名 (Fully qualified Domain Name)
HTTP	超文本传输协议 (HyperText Transfer Protocol)
IC	长途电信公司 (Inter-exchange Carrier)
IETF	互联网工程任务组 (Internet Engineering Task Force)
IKE	互联网密钥交换 (Internet Key Exchange)
IP	互联网协议 (Internet Protocol)
IPSec	互联网协议安全性 (Internet Protocol Security)
ITU	国际电信联盟 (International Telecommunication Union)
ITU-T	国际电信联盟—电信标准化部门 (International Telecommunication Union – Telecommunication Standardization Sector)
LNP	本地号码可携带性 (Local Number Portability)
MC	多点控制器 (Multipoint Controller)
MG	媒体网关 (Media Gateway)
MGC	媒体网关控制器 (Media Gateway Controller)
MGCP	媒体网关控制器协议 (Media Gateway Control Protocol)
MIB	管理信息库 (Management Information Base)
MTA	媒体终端适配器 (Media Terminal Adapter)
NCS	网络呼叫信令 (Network Call Signalling)
OSS	操作支持系统 (Operations Support System)
PCM	脉码调制 (Pulse Code Modulation)
PCSP	IPCablecom CMS用户配置 (IPCablecom CMS Subscriber Provisioning)
PS	配置服务器 (Provisioning Server)
PSTN	公共交换电话网 (Public Switched Telephone Network)
QCIF	四分之一通用中间格式 (Quarter Common Intermediate Format)
RAS	注册、准入和状态 (Registration, Admissions and Status)
RFC	请求评论 (Request for Comments)

SDP	会话描述协议 (Session Description Protocol)
SNMP	简单网络管理协议 (Simple Network Management Protocol)
SOAP	简单对象接入协议 (Simple Object Access Protocol)
STP	信令传输点 (Signalling Transfer Point)
TD	断开超时 (Timeout for Disconnect)
TFTP	小文件传输协议 (Trivial File Transfer Protocol)
TLV	类型长度值 (Type-Length-Value)
UDP	用户数据报协议 (User Datagram Protocol)
VoIP	IP语音 (Voice Over IP)

4.2 用词说明

在本建议书中，用于定义某需求重要性的单词大写。这些词汇如下：

“务必 (MUST)”	这个词或形容词“必需的 (REQUIRED)”意指：该条款是本建议书的绝对要求。
“绝不 (MUST NOT)”	这个词组意指：该条款是本建议书的绝对禁令。
“应 (SHOULD)”	这个词或形容词“建议的 (RECOMMENDED)”意指：在实际环境中有可能存在正当的理由对这一条款不予理会，但是，在选择不同的做法之前应充分理解全部含义和小心权衡理由。
“应不 (SHOULD NOT)”	这个词组意指：在实际环境中有可能存在正当的理由，考虑到所列举的行为是可接受的或甚至是可用的。但是，在实际用这个标记描述的任何行为之前，应充分理解全部含义和小心权衡理由。
“可 (MAY)”	这个词或形容词“可选的 (OPTIONAL)”意指：这一条款是真正可选的。例如，某个供货商可以选择含有该条款，因为实际市场需要它或因为它能提高产品价值；而另外的供货商可以忽略同样的条款。

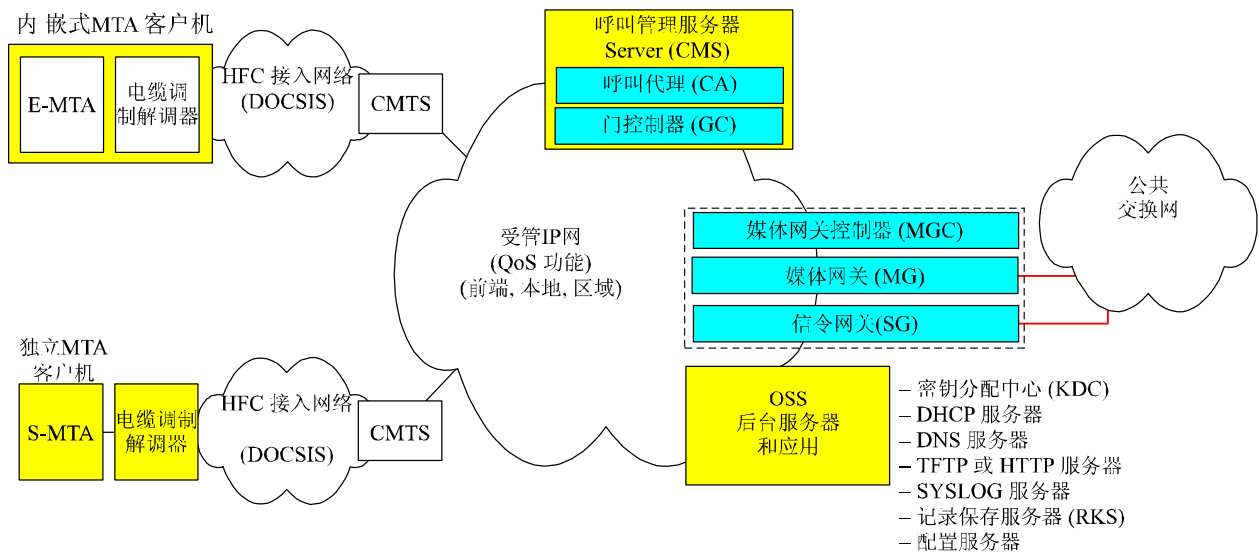
5 背景

5.1 业务目标

内容已删除。

5.2 IPCablecom参考结构

图1所示为IPCablecom网络的参考结构。有关此参考结构的详细信息，请参阅IPCablecom结构建议书，即ITU-T J.160建议书。



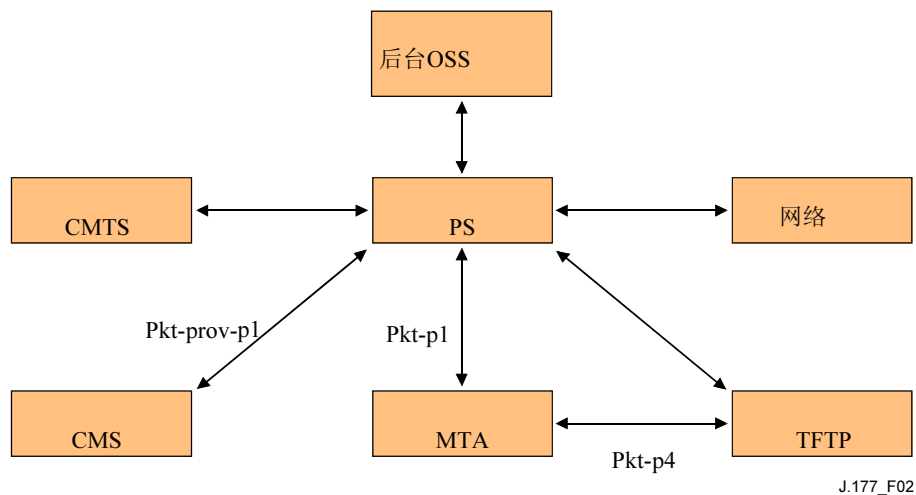
J.177_F01

图 1/J.177 – IP-Cablecom 1.0网络组件参考模型（部分）

5.3 组件和接口

配置是为客户提供特定业务所需的操作。IP-Cablecom业务配置可视为两类不同操作：MTA配置和CMS用户配置。图2所示为由配置服务器（PS）维护的与配置相关的接口，以及为各类IP-Cablecom网元授权的后台组件。没有明确标出的接口还未定义且不在IP-Cablecom的范围之内。

此建议书旨在为CMS和PS之间的配置接口或（可选）其它授权的后台组件（Pkt-prov-p1）制定要求。



J.177_F02

图2/J.177 – 配置组件接口

5.4 组件

5.4.1 后台组件（业务提供商业务和服务管理系统）

这些是业务提供商用来管理客户和其它业务的组件的后台组件。这些系统为IP-Cablecom配置流程提供业务命令或提供独立的工作流程任务来激活客户业务。这些系统还可以用于接收计费或使用数据，以创建客户计费事件。

5.4.2 配置服务器

此系统是提供商后台系统组件与某些或全部IP-Cablecom元素之间的接口。IP-Cablecom不负责此系统的实施，也不负责处理该系统与业务提供商可能使用的其它OSS的关系。

根据ITU-T J.167建议书对配置服务器的定义，该服务器的组成包括配置逻辑和能够提供有效组件接入的SNMP配置实体。在此，我们在提及配置服务器时，不对这两类实体做区分。

5.4.3 CMS

ITU-T J.160建议书描述了呼叫管理服务器组件。此组件为IP-Cablecom网络中的MTA和CMTS组件提供所有与控制及信令相关的服务。

5.4.4 MTA

媒体终端适配器是一种IP-Cablecom客户端设备，其中包括与客户CPE（例如，电话）之间的用户侧接口和与网络中呼叫控制网元间的网络侧信令接口。ITU-T J.160建议书描述了这一组件。

5.4.5 TFTP

一种作为IP-Cablecom网络中所有设备配置基础的配置文件服务。它可以是一种为设备提供数据定义文件的独立的TFTP业务，也可以是根据其它数据源随时创建配置的动态业务。

5.5 接口说明

5.5.1 Pkt-p1

ITU-T J.167建议书定义了此接口。

5.5.2 Pkt-p4

ITU-T J.167建议书定义了此接口。

5.5.3 Pkt-prov-p1

本建议书定义的接口。

6 假设

- 后台组件负责与受影响网络实体（MTA、CMTS等）和CMS协调端点的更新。
- 在CMS配置过程中，CMS将不会扮演管理器的角色，也不会规范SNMP与MTA的通信。

- CMS和PS位于相同的安全配置域内。与安全性相关的信息将由IPCablecom安全性建议书，即ITU-T J.170建议书做概要介绍。

7 用户配置

用户配置包括：

- 客户记录/计费支持；
- 设备安装/配置。

7.1 客户记录（计费）

建立包含向客户提供业务和帐单以及从客户处收款所需信息的客户记录。客户记录创建/计费被认为是后台OSS应用的一部分，且目前不包括在IPCablecom的范围之内。

7.2 设备安装与配置

这一步骤可能包括设备的物理安装和/或连接以及向客户实际提供业务所需的任何软件和/或数据库更新。设备的安装会影响IPCablecom环境中的两大组件。

- 客户所在地设备。对于IPCablecom，这就是MTA。ITU-T J.167建议书定义了MTA的配置，但未在本建议书中讨论。
- 呼叫管理服务器。CMS自身的配置可以细分为两个主要领域：基本POTS配置和呼叫功能配置。

7.2.1 CMS基本POTS配置（BPP）

BPP为CMS提供在IPCablecom网络中路由简单电话业务（POTS）所需的最小数量的数据。这一最小数量的数据集包括映射到相关MTA和FQDN的电话号码和NCS端点标识符。此数据将用于建立转换表，使CMS可将呼叫路由至某特定电话号码的相应设备/端口。在客户能够在IPCablecom网络中接收任何呼叫之前，有必要实施BPP。

7.2.2 CMS呼叫功能配置（CFP）

除BPP之外，通过CFP为客户提供呼叫功能。CFP比BPP更加复杂，因为通过的传输参数可能会根据功能的不同而不同，也可能因厂商特定的实施而异。

7.3 静态与动态用户配置数据

CMS所需的用户配置数据分为两类：

- 1) 静态、已计费、永久指配的业务状态。这一数据不会随呼叫改变。实例包括DQoS设置、已订阅/未订阅状态的呼叫功能，呼叫方的ID信息等。
- 2) 动态、未计费、半永久业务状态。通常此信息是可变的，其改变可以在端点通过*XX密钥码进行，或者通过与CMS相连的网络接口实施。其实例包括可由服务设置的呼叫特性参数，例如线路忙时呼叫前转（CFBL）。CFBL的前转号码处于动态、未计费业务状态。CFBL的已订阅/未订阅状态是由PS维护的静态数据。

在IPCablecom CMS/PS范围内，PS拥有全部静态配置状态，而CMS拥有全部动态配置状态。

8 要求

8.1 一般性要求

- 该接口绝不能对PS和CMS的实施技术做出假设。
多家合作厂商无疑将使用不同硬件、软件和开发语言平台进行CMS和PS实施。需要使用中立的平台和语言接口。
- 接口务必支持基本POTS配置。
接口的数据模型务必包括支持基本POTS业务所需的最低数量的信息。
- 接口务必支持呼叫功能配置。
接口的数据模型务必支持订阅任何IP Cablecom呼叫功能。
- 接口的数据模式务必可以扩展。
该接口目前的焦点是电话数据。但该接口应尽可能地为将来的IP Cablecom多媒体业务留下扩展的空间。宜使用一种单一、可扩展的配置数据模型和传输方式，以支持所有IP Cablecom功能和能力，但其中有些还未确定。
- 该接口绝不能影响任何正在进行的MTA操作。
特定的端点数据可在不影响其它MTA端点或正在进行的会话的情况下增加、删除或修改。必须仔细记录下可能会导致端点/MTA退出业务的CMS端点配置情境。
- 该接口务必能够接纳现有（NCS）和将来的信令协议。

8.2 传输要求

- 传输绝不能对PS与CMS之间的物理网络基础设施做出假设。
预计多家业务提供商将在同一接入网上进行互操作。因此，多家企业将有可能使用来自多个厂商的CMS和PS进行实施，在不同网络基础设施（防火墙、代理服务器等）上进行通信。CMS/PS传输协议应有助于提高进入任意网络基础设施的能力。
- 传输务必支持单一数据模型对象从PS向CMS的单向传输。
- 传输务必支持多数据模型对象从PS向CMS的高效流式传输。
- 传输可支持单一数据模型对象从CMS向PS的单向传输。
- 传输可支持多数据模型对象从CMS向PS的高效流式传输。
- 传输务必包括支持新的、经更新的和删除的数据模型对象的语义。
- 传输务必支持PS与CMS间的信息请求。
- 传输务必能够处理CMS繁忙、错误等情况。
- 传输务必对收到的操作的予以肯定/否定的确认。
- 传输务必使用“至少一次”类型的消息语义。在接收方确认之前，发送方绝不放弃这一请求（确认消息还未被确认）。传输功能务必能在传输过程中检测出数据损坏等问题，并将此类情况通知发送方。

- 传输务必对已处理的操作予以肯定/否定的确认。
- PS务必能够启动数据模型对象的传输（“推送”）。
- 传输务必安全。

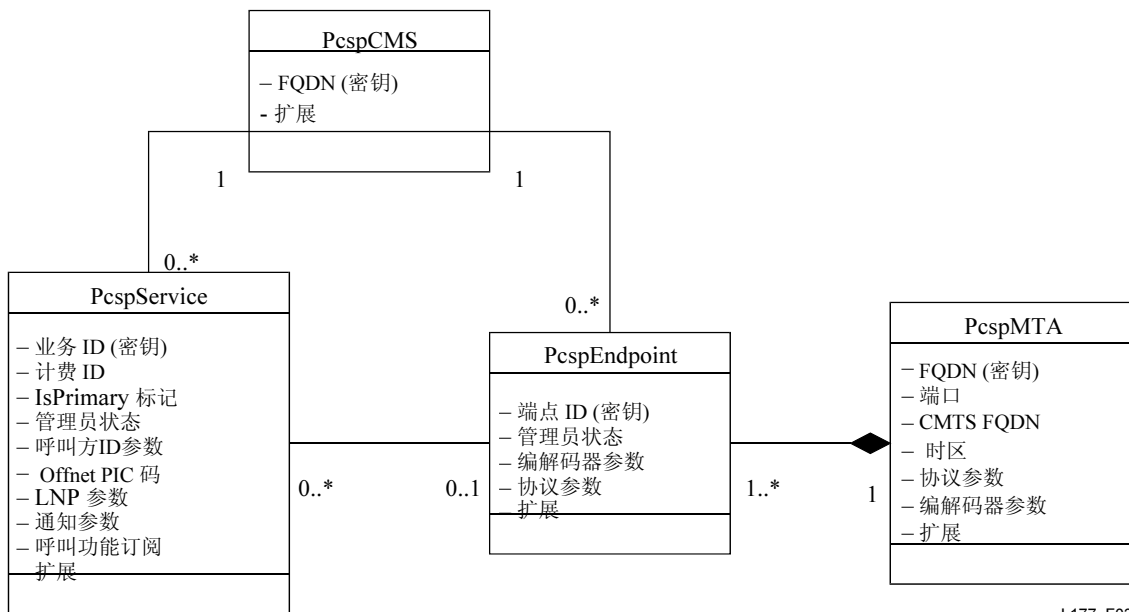
9 数据模型

本节提供了对PCSP数据模型及其XML编码的高层说明，其目的在于说明而非指令。在附件A的PCSP XML方案中可以找到数据模型及其编码的规范性权威定义。

9.1 概述

图3中显示IPCablecom CMS配置的数据模型，其中包括两类实体：

- 对象；
- 对象间的关系。



J.177_F03

图 3/J.177 – CMS配置数据模型

务必支持下述实体：

- PcpService对象是一种IPCablecom 1.0客户订阅的实体。它代表电话号码和所有相关的功能（呼叫功能等）。
- PcpMTA对象代表将MTA包含的一个或多个物理端点聚合的媒体终端适配器。
- PcpEndpoint对象代表MTA/网关的物理端点。

- PcpService对象维持端点/CMS和业务/CMS之间的关联。
- PcpRelations表示对象之间的关系。在图3中，它们显示为对象间的连接。

PcpService和PcpEndpoint是两个支持各端点多业务（电话号码）的明确对象。明确的PcpMta和PcpEndpoint对象MTA的端点由不同业务提供商管理。PcpCms对象维持着一批端点和业务。

所有对象均可扩展。

9.1.1 PcpService对象

此业务对象是IPCablecom 1.0客户订阅的实体。它代表一个电话号码及所有相关的功能。此数据模型允许为单一端点配置一个以上的业务。

PcpService对象包含下述一般性信息（完整的详细信息，请参见PCSP XML方案）：

- ServiceId – 该业务使用的唯一标识符；
- BillingId – 另一业务使用的标识符，该业务在此业务中进行的活动需要计费；
- IsPrimary标识 – 在端点配置有多种业务时，其中一种业务必须设置这一标记，以说明用于出局呼叫的缺省业务。
- PrimaryRingPattern – MTA振铃节奏表的索引，负责为此业务选择一种振铃模式；
- 此业务的管理状态（暂停、启动、号码已改变等）；
- DisplayName – 显示使用呼叫名称交付功能（CNAM）的信息；
- DisplayNumber – 显示使用呼叫号码交付功能（CND）的信息；
- 通知设置（启动、语言、时区等）；
- 运营商代码（长途电信公司的代码、LATA内电信公司的代码、国际运营商代码）；
- 本地号码可携带性控制（移植的状态、STP查询标记等）；
- 呼叫功能 – 业务包括一系列订阅呼叫功能对象；
- 扩展 – 此对象可在两处扩展：对象的主体和呼叫功能清单。

9.1.2 PcpMta对象

媒体终端适配器聚合一个或多个端点（设备本身置于MTA内）。它包括下述一般性信息（完整的详细信息，请参见PCSP XML方案）：

- MTA的FQDN，作为此MTA的唯一标识；
- MTA的NCS收听端口（缺省值：2427）；
- 控制CMTS的FQDN；
- MTA物理位置所在地的时区；
- 信令协议的指定 – 除非被一独立端点取代，此指定是为所有包含在内的端点选择的缺省协议；
- 编解码器的指定 – 除非被一独立端点取代，此指定是为所有包含在内的端点选拔的缺省编解码器；
- IPsec控制标记 – IPsec控制标记指出IPsec是否被用于CMS与MTA之间的NCS信令。缺省状态下，IPsec在所有端点都处于开启状态，但也可以对各端点分别配置。

- MTA特性名称– 可选。可由CMS识别的MTA特性指示符；
- 单一的扩展点。

9.1.3 PfspEndpoint对象

端点是MTA/网关的物理端口。其中包含下述一般性信息（完整的详细信息，请参见PCSP XML方案）：

- EndpointId – 此端点的唯一标识；
- 信令协议选择 – 可以取代MTA设置；
- 端点的管理状态 – （断开、普通服务、测试模式等）；
- 编解码器的选择 – 可以取代MTA设置；
- IPsec控制标记 – 可以取代MTA设置；
- 单一的扩展点。

9.1.4 PfspCms对象

此对象保持端点/CMS和业务/CMS之间的关联。其中包含下述一般性信息（完整的详细信息，请参见PCSP XML方案）：

- 作为此CMS唯一标识的FQDN；
- 单一的扩展点。

9.1.5 对象间关系

在图3中，连接类别的线代表对象之间的“关系”（有时称为关联）。图3中描述的关系必须支持：

- 业务/CMS – 典型的CMS将拥有一组电话号码；
- 端点/CMS – 端点需要一个用于信令的CMS；
- 业务/端点 – 物理端点必须附有一个电话号码；
- 端点/MTA – MTA实际包含的端点。

9.2 利用PfspRelation entity.XML编码方式对关系进行编码

将使用XML对数据模型对象进行编码。

9.2.1 PCSP XML方案

附件A中包含的PCSP XML方案，为下述实体定义了XML编码的语法（这些实体务必要遵守此方案的规定）：

- PfspService、PfspEndpoint、PfspMta和PfspCms对象 – 这些是主要的数据模型对象；
- PfspRelation – 用于建立或取消对象之间的关系；
- PfspImportExport – 一种可以包括大量对象或关系的通用文件格式，通常用于从PS向CMS输出完整的数据集。

此方案应使用负责验证工作的XML分析程序来确定编码实体的语法是否正确。

9.2.2 PCSP实体编码的示例

附录I包括所有PCSP数据模型实体的XML编码范例。

9.2.3 对象扩展

PCSP XML方案允许对所有对象（PcspService、PcspEndpoint、PcspMta和PcspCms）进行扩展。扩展是通过各对象中的<扩展>元素实现的。大部分对象在此对象的主体末端确定这一元素。PcspService在呼叫功能清单的末尾包括一个额外的<扩展>元素。

<扩展>元素存在若干简单规则。

- 所有<扩展>元素务必指定一个名称空间（namespace）定义。
- 所有<扩展>子元素务必通过namespace的验证。

这两条规则允许XML对系统进行分析，以便根据厂商提供的XML方案文件验证<扩展>内容。附录II中包括一个扩展的示例。

10 消息

10.1 概述

PCSP接口遵循网络业务范例中使用的方法。该接口用SOAP 1.1消息在客户机与服务器之间传输XML编码的实体（从PCSP数据模型1）。使用HTTP 1.1协议在客户机与服务器之间传输消息。有关传输考虑的完整讨论，请参见附录IV。

此接口采用了同步请求/响应模式（或远程程序呼叫- RPC）。在客户机与服务器之间支持下述消息模式：

- PUT消息- 客户机向服务器写出一个或多个XML编码对象或关系。同时支持创建新对象和对现有对象的修改。
- DELETE消息- 客户机请求从服务器删除一个或多个对象或关系。
- GET消息- 从服务器读取一个或多个XML编码对象（只支持对象-不支持关系）。
- CMDSTATUS消息- 用于在客户机与服务器之间传输“带外（out of band）”命令和状态。客户机可以向服务器通报各种状态条件。客户机可以命令服务器执行各种操作。此消息可由厂商进行扩展。

10.2 对CMS和PS消息功能的要求

总体而言，CMS和PS都可用于为客户机和服务器的消息功能提供全面的支持。但是，在IPCablecom CMS配置的范围内，CMS和PS功能应符合表1对功能的要求。

表 1/J.177 – CMS和PS消息功能

消息名称	CMS作为客户机	CMS作为服务器	PS作为客户机	PS作为服务器
GET	可选	务必	务必	可选
PUT	可选	务必	务必	可选
DELETE	可选	务必	务必	可选
CMDSTATUS	务必	务必	务必	务必

应注意以下几点：

- CMS 务必支持服务器的GET、PUT和DELETE功能。
- PS 务必支持客户机的GET、PUT和DELETE功能。
- CMS和PS 务必支持客户机和服务器的CMDSTATUS功能。
- 所有其它操作均为可选。

这些要求强制配置数据从PS流向CMS，并确保不需要CMS将动态数据修改（可由用户调整的呼叫功能修改等）推送回PS。

PS可以从CMS读取特定的对象。支持此种使用方式旨在允许PS检索CMS的用户呼叫功能设置（“动态数据”）。此功能通过从CMS读取特定的PcspService对象实现。

图4显示了所有需要的消息功能。

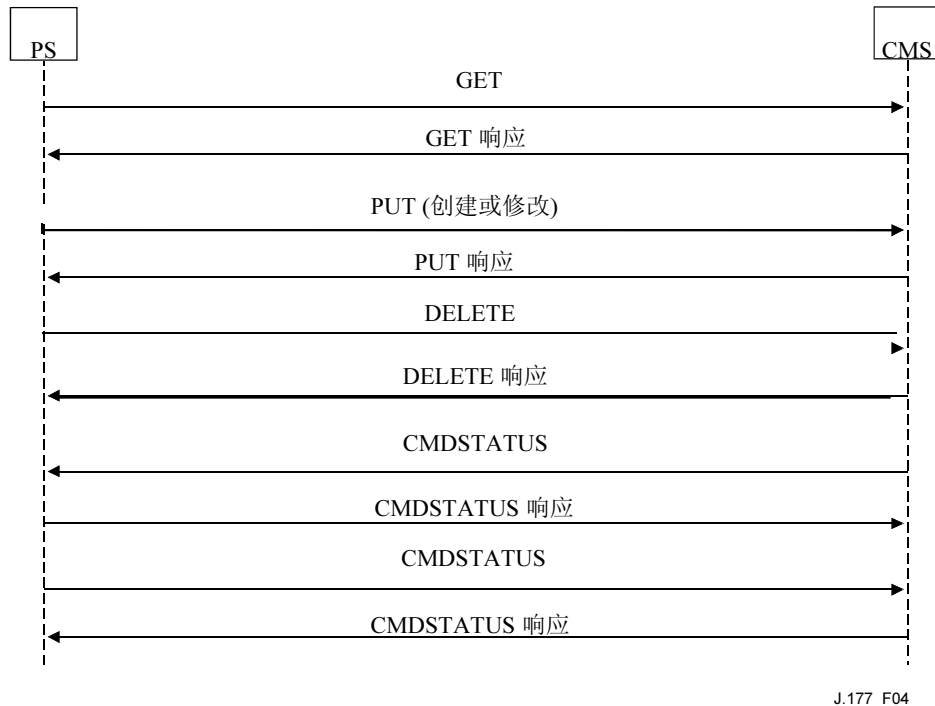


图4/J.177 – 需要的消息流

10.3 WSDL规范

PCSP接口是利用网络业务描述语言1.1版说明的。与CORBA IDL相同，WSDL接口定义规范了接口的远程操作法、这些方法接受的自变量、这些方法返回的值，以及必须定义的所有接口专用的数据类型。此外，WSDL定义还规范了消息编码格式（SOAP 1.1）和传输绑定（HTTP 1.1）。

WSDL被装入各类网络业务工具箱，供大部分操作系统和语言自动生成客户机接口桩（stub）、服务器框架（skeleton）和SOAP编组（marshalling）支持。

PCSP客户机和服务器务必符合附件B中的WSDL定义。

11 安全性

PCSP接口用传输模式的IPSec ESP协议确保安全性。密钥管理是通过具有预共享密钥的IKE实施的。这一安全性基础设施已在CMS中用于各种接口。详细内容，请参见ITU-T J.170建议书。

附件A

PCSP XML模式

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  IPCablecom CMS Subscriber/Service Provisioning (PCSP) schema.

  PCSP defines a messaging interface and an XML encoding format for objects
  transmitted over that interface. This schema defines the XML encoding syntax for the objects
  transmitted over the PCSP interface.

  Encodings for PcpsService, PcpsEndpoint, PcpsMta, and PcpsCms objects are specified.
  A PcpsRelation encoding describes associations between objects.
  A PcpsImportExport encoding is used to produce instance documents containing large numbers of
  objects.
-->
<xs:schema
targetNamespace="http://www.cablelabs.com/Pcps/I01/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.cablelabs.com/Pcps/I01/schema"
elementFormDefault="qualified">
  ===== TYPE DEFINITIONS =====
  -->
  <!--
    A non-empty string.
  -->
  <xs:simpleType name="nonEmptyString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <!--
    A Service ID.
    A non null string containing a "format" attribute (an enumeration), which defaults to 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>
  <!--
    A relation operation type.
    Used to indicate relation is being "added" or "deleted".
  -->
  <xs:simpleType name="RelationOpType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="add"/>
      <xs:enumeration value="delete"/>
    </xs:restriction>
  </xs:simpleType>
  <!--
    An enumeration of legal object "class" names.
  -->
  <xs:simpleType name="classType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="PcpsService"/>
      <xs:enumeration value="PcpsCms"/>
      <xs:enumeration value="PcpsEndpoint"/>
      <xs:enumeration value="PcpsMta"/>
    </xs:restriction>
  </xs:simpleType>
  <!--
    A list of object keys.
  -->
  <xs:complexType name="ListOfKeys">
    <xs:sequence>
```

```

        <xs:element name="Key" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<!--
Codec types.
An enumeration that matches the PktcCodecType object from the
IPCablecom™ Audio/Video Codecs Specification J.161.
This enumeration sh be kept in sync with the aforementioned specification.
For convenience, value definitions are repeated here:

    1: other.
    2: unknown.
    3: G729
    4: reserved
    5: G729E
    6: PCMU
    7: G726-32
    8: G728
    9: PCMA
    10: G726-16
    11: G726-24
    12: G726-40
    13. iLBC
    14. BV-16

In the case of the PcpEndpoint object, a codec value of "2" shall be
interpreted as "use the MTA's codec specification".

-->
<xs:simpleType name="codecType">
    <xs:restriction base="xs:integer">
        <xs:enumeration value="1"/>
        <xs:enumeration value="2"/>
        <xs:enumeration value="3"/>

        <xs:enumeration value="4"/>
        <xs:enumeration value="5"/>
        <xs:enumeration value="6"/>
        <xs:enumeration value="7"/>
        <xs:enumeration value="8"/>
        <xs:enumeration value="9"/>
        <xs:enumeration value="10"/>
        <xs:enumeration value="11"/>
        <xs:enumeration value="12"/>
        <xs:enumeration value="13"/>
        <xs:enumeration value="14"/>
    </xs:restriction>
</xs:simpleType>
<!--
Signalling protocol designations.
PcpEndpoint employs "MtaDefault" to force use of the MTA's default protocol setting.
-->
<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>
<!--
Numeric timezone designation per RFC 1123.
-->
<xs:simpleType name="TimezoneType">
    <xs:restriction base="xs:string">
        <xs:pattern value="[\+|-]\d{4}"/>
    </xs:restriction>
</xs:simpleType>
<!--
Local number PortingStatus

    0: not ported.
    1: ported in (owned by another TSP)
    2: ported out (loaned to another 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>
<!--
===== SUPPORTING ELEMENT DEFINITIONS =====
-->
<!--
Network announcement control. Contains...

Language - per XML schema language type.

Timezone - see previous definition.
-->
<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>
<!--
Interexchange carrier codes. Used to route offnet regional, long distance, and international
calls to specific carriers.

PIC - Predesignated interexchange carrier (long distance).

LPIC - Predesignated intra-LATA carrier

IPIC - Predesignated international carrier
-->
<xs:element name="InterExchange">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PIC" type="xs:string"/>
      <xs:element name="LPIC" type="xs:string"/>
      <xs:element name="IPIC" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
Local Number Portability parameters.

PortingStatus - see portingStatusType.

LNPT - LNP trigger determines if number is in transition.
false/0: no STP lookup required.
true/1: STP lookup required to determine LRN of destination switch.
-->
<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>
<!--
Vendor Extension element.

Used within the PcspService, PcspCms, PcspMta, and PcspEndpoint objects to enable vendor
extensions.
Also used to extend the call feature list within the PcspService object.
-->
<xs:element name="Extension">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" processContents="strict" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
===== CALL FEATURES =====

A service includes a list of call feature objects, each encoding one of the call features
described in PKT-TR-VOIPBRF-R01-000608 and PKT-TR-VOIPERF-R01-000831.

Each call feature includes its "static" state data (owned by the PS):
Feature name (implicitly as the element name),
Subscribed/non subscribed state,
Administrative state the feature.

```

Many call features include just this information.
Absence of a specific call feature implies the feature is not subscribed.
The subscribed state is used to indicate that an
explicitly listed call feature is not subscribed (an atypical case).

Several features extend the "static" parameter set with feature specific data.
This feature specific data is typically configured by the user (via handset or by calling a
CSR). The PCSP spec classifies the user adjustable data as "dynamic", meaning that it is owned
by the CMS. Changes to the dynamic data in the CMS are not required to be pushed back to the
PS.

```
-->
<!--
  Always -
    false/0: Subscriber may change forward-to number.
    true/1: Service provider (only) may change forward-to number
-->
<xs:element name="Always" type="xs:boolean"/>
<!--
  ForwardTo - Service Id to which call will be forwarded.
  NOTE: empty strings are allowed.
-->
<xs:element name="ForwardTo" type="xs:string"/>
<!--
  ListOfServiceId - a list of Service Ids.
-->
<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 - list of Service Ids / speed dial # pairs.
  Each pair contains a one or two digit speed dial number and its associated service id.
-->
<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>
<!--
  The definition of each of the supported call features:

  All call features can be considered in two parts.
  1. Common section containing the administrative state of the feature (the "static" data).
  2. An optional section containing feature specific parameters, typically set by the end user
(the "dynamic" data).
-->
<!--
  The "base object" for all call features, containing:

  Subscribed -
    0/false: feature is not subscribed
    1/true: feature is subscribed.
  UsageBilling -
    0/false: do not generate billing records on feature usage
    1/true: generate billing records on feature usage.

  AdminStatus -
    0: feature is suspended by service provider.
    1: feature is enabled by service provider.
```

In general, presence of a call feature implies that it is subscribed.
 The Subscribed flag is supported for the atypical case of wanting to
 indicate that an explicitly listed call feature is not subscribed.

```
-->
<xs:complexType name="CfBase">
  <xs:sequence>
    <xs:element name="Subscribed" type="xs:boolean"/>
    <xs:element name="UsageBilling" type="xs:boolean" minOccurs = "0"/>
    <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" Calling Number Delivery
-->
<xs:element name="CfCND">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CNAM": Calling Name Delivery
-->
<xs:element name="CfCNAM">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CIDCW": Calling Identity Delivery on Call Waiting
-->
<xs:element name="CfCIDCW">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CW": Call Waiting
-->
<xs:element name="CfCW">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CCW": Cancel Call Waiting (*70)
-->
<xs:element name="CfCCW">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CFV": Call Forwarding Variable and Usage- Sensitive Call Forwarding (*72/*73)

  Extends CfBase with the following:

  Active -
    0/false: user has deactivated feature (*73).
    1/true: user has activated feature (*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": Automatic Recall (*69)
-->
<xs:element name="CfAR">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "AC": Automatic Callback (*66)
-->
<xs:element name="CfAC">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "VMWI": Visual Message Waiting Indicator
  Extends CfBase with the following:

  Indicator Type -
    0: None.
    1: Stutter Dial tone Only
    2: Message Lamp Only
    3: Both Stutter Dial tone and Message Lamp
-->
<xs:element name="CfVMWI">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase">
        <xs:sequence>
          <xs:element name="UserParams" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Type">
                  <xs:simpleType>
                    <xs:restriction base="xs:int">
                      <xs:enumeration value="0"/>
                      <xs:enumeration value="1"/>
                      <xs:enumeration value="2"/>
                      <xs:enumeration value="3"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "COT": Customer Originated Trace (*57)
-->
<xs:element name="CfCOT">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

```

<!--
  "TWC": Three-Way Calling / Usage-Sensitive Three-Way Calling (*71)
-->
<xs:element name="CfTWC">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "RACF": Remote Activation of Call Forwarding
-->
<xs:element name="CfRACF">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "OCAA": Outside Calling Area Alerting
-->
<xs:element name="CfOCAA">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CIES": Calling Identity with Enhanced Screening
-->
<xs:element name="CfCIES">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "ACR": Anonymous Call Rejection (*77 / *87)

  Extends CfBase with the following:

  Active -
    0/false: user has deactivated feature (*87).
    1/true: user has activated feature (*77).
-->
<xs:element name="CfACRestrict">
  <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": Automatic Callback - Restrict
-->
<xs:element name="CfAC-R">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "ACB": Automatic Recall Blocking
-->
<xs:element name="CfACB">

```

```

<xs:complexType>
  <xs:complexContent>
    <xs:extension base="CfBase"/>
  </xs:complexContent>
</xs:complexType>
</xs:element>
<!--
  "CIDB" Calling Identity Delivery Blocking (*67 / *82).

  Extends CfBase with the following:

  Flag -
    "PUBLIC": deliver Caller ID info
    "ANONYMOUS": do not deliver Caller ID info.
-->
<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:complexType>
</xs:element>
<!--
  "CFBL" Call Forwarding Busy Line ( *68 / *40 / *88 ).

  Extends CfBase with the following "dynamic", user
  adjustable parameters (owned by the CMS).

  Active -
    0/false: user has deactivated feature (*88).
    1/true: user has activated feature (*68/*40).

  Always - see previous definition.

  ForwardTo - see previous definition.
-->
<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" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "CFDA" Call Forwarding Don't Answer (*68 / *42 / *88)

  Extends CfBase with the following "dynamic", user
  adjustable parameters (owned by the CMS):

  Active -
    0/false: user has deactivated feature (*88).

```

1/true: user has activated feature (*68/*42).

Always - see previous definition.

RingPeriod - number of ringing cycles after which forwarding is activated.

ForwardTo - see previous definition.

```
-->
<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" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

```
<!--
  "CFC" Call Forwarding Combination
```

Extends CfBase with the following "dynamic", user adjustable parameters (owned by the CMS):

Active -

0/false: user has deactivated feature (*88).

1/true: user has activated feature (*68).

Always - see previous definition.

RingPeriod - number of ringing cycles after which forwarding is activated.

ForwardTo - see previous definition.

```
-->
<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" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

```
<!--
  "SCF" Selective Call Forwarding (*63/*83).
```

Extends CfBase with the following "dynamic", user adjustable parameters (owned by the CMS):

Active -

0/false: user has deactivated feature (*83).

1/true: user has activated feature (*63).

ListOfServiceId - list of service identifiers that will be forwarded. See previous element definition.

ForwardTo - the service to which to forward. See previous element definition.

```
-->
<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" minOccurs="0"/>
              <xs:element ref="ForwardTo" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:element>
<!--
"SCA" Selective Call Acceptance (*64 / *84 ).

Extends CfBase with the following "dynamic", user
adjustable parameters (owned by the CMS):

Active -
  0/false: user has deactivated feature (*84).
  1/true: user has activated feature (*66).

ListOfServiceIds - list of service identifiers that will be accepted. See previous element
definition.
-->
<xs:element name="CfSCA">
  <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" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
"SCR" Selective Call Rejection (*60 / *80 ).

Extends CfBase with the following "dynamic", user
adjustable parameters (owned by the CMS):

Active -
  0/false: user has deactivated feature (*80).
  1/true: user has activated feature (*60).

ListOfServiceIds - list of service identifiers that will be rejected. See previous element
definition.
-->
<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" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

```

<!--
  "DRCW" Distinctive Ringing/Call Waiting (*61 / *81)

  Extends CfBase with the following "dynamic", user
  adjustable parameters (owned by the CMS):

  Active -
    0/false: user has deactivated feature (*81).
    1/true: user has activated feature (*61).

  ListOfServiceIds - list of incoming service identifiers that will receive the distinctive ring
  treatment (vs standard power ring or call waiting tone). See previous element definition.
-->
<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" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "SPCALL" Speed Calling (*74 / *75)

  Extends CfBase with the following "dynamic", user
  adjustable parameters (owned by the CMS):

  ListOfSpeedDial - see previous element definition.
-->
<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" Residence Distinctive Alerting Service.
-->
<xs:element name="CfRDA">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CfBase"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "LSR" Line Service Restriction.

  Extends CfBase with the following "dynamic", user
  adjustable parameters (owned by the CMS):

  BlkDomLongDist - block for outgoing domestic long distance calls.
    0/false: not blocked.
    1/true: blocked.
  BlkIntlLongDist - block for outgoing international long distance calls.
    0/false: not blocked.
    1/true: blocked.
  BlkPayPerCall - block for outgoing pay per calls (900/976).
    0/false: not blocked.

```

1/true: blocked.
 BlkOperatorAssist - block for outgoing operator assisted calls.
 0/false: not blocked.
 1/true: blocked.
 BlkDirAssist - block for outgoing directory assistance calls.
 0/false: not blocked.
 1/true: blocked.
 BlkTollFree - block for outgoing toll free calls.
 0/false: not blocked.
 1/true: blocked.
 Active -
 0/false: user has deactivated feature (*82).
 1/true: user has activated feature.
 PIN - code to enter to deactivate blocking
 ServiceList - list of service identifiers for domestic long distance calls that are always
 allowed.

```
-->
<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" minOccurs="0"/>
                <xs:element name="BlkIntLongDist" type="xs:boolean" minOccurs="0"/>
                <xs:element name="BlkPayPerCall" type="xs:boolean" minOccurs="0"/>
                <xs:element name="BlkOperatorAssist" type="xs:boolean" minOccurs="0"/>
                <xs:element name="BlkDirAssist" type="xs:boolean" minOccurs="0"/>
                <xs:element name="BlkTollFree" type="xs:boolean" minOccurs="0"/>
                <xs:element name="PIN" type="xs:string" minOccurs="0"/>
                <xs:element name="Active" type="xs:boolean"/>
                <xs:element ref="ListOfServiceId" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  "DND" Do Not Disturb
```

Extends CfBase with the following "dynamic", user adjustable parameters (owned by the CMS):

Active -
 0/false: user has deactivated feature.
 1/true: user has activated feature.

WeekDayStartTod1 - week day start time for DND.
 WeekDayStopTod1 - week day stop time for DND.
 WeekDayStartTod2 - week day start time for DND.
 WeekDayStopTod2 - week day stop time for DND.
 WeekEndStartTod1 - week end start time for DND.
 WeekEndStopTod1 - week end stop time for DND.
 WeekEndStartTod2 - week end start time for DND.
 WeekEndStopTod2 - week end stop time for 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" minOccurs="0"/>
                <xs:element name="WdStopTod1" type="xs:time" minOccurs="0"/>
                <xs:element name="WdStartTod2" type="xs:time" minOccurs="0"/>
                <xs:element name="WdStopTod2" type="xs:time" minOccurs="0"/>
                <xs:element name="WeStartTod1" type="xs:time" minOccurs="0"/>
                <xs:element name="WeStopTod1" type="xs:time" minOccurs="0"/>
                <xs:element name="WeStartTod2" type="xs:time" minOccurs="0"/>
                <xs:element name="WeStopTod2" type="xs:time" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

```

        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<!--
    "COC" Curfew on Calls.

    Extends CfBase with the following "dynamic", user
    adjustable parameters (owned by the CMS):

    Active -
        0/false: user has deactivated feature.
        1/true: user has activated feature.

    StartTod - start time for COC.
    StopTod - stop time for COC.
    ServiceList - list of service identifiers for incoming and outgoing services which are allowed
to bypass the 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" minOccurs="0"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<!--
    "NSA" No Solicitation Announcement

    Extends CfBase with the following "dynamic", user
    adjustable parameters (owned by the CMS):

    Active -
        0/false: user has deactivated feature.
        1/true: user has activated feature.

    StartTod - start time for COC.
    StopTod - stop time for COC.
    ServiceList - list of service identifiers for incoming and outgoing services which are allowed
to bypass the 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" minOccurs="0"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<!--
    A list of call features. The list may contain at most 1 of each of the
    features outlined above, along with any vendor extension call features.
-->
<xs:element name="ListOfCallFeatures">

```



```

<xs:complexType>
  <xs:all>
    <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="CfACRestrict" 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="CfSCA" 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:all>
</xs:complexType>
</xs:element>
<!--

```

===== MAIN OBJECT DEFINITIONS =====

There are 6 encodings defined in the PCSP schema.

The 4 main object encodings:

- PcspCms - a CMS. A collection of Services and Endpoints.
- PcspService - represents a phone number, its configuration, and call features.
- PcspMta - represents a physical MTA and its configuration. A collection of Endpoints.
- PcspEndpoint - represents an Endpoint on an MTA.

A PcspRelation object. This object encodes the associations between objects.

A PcspImportExport object. This is used to produce a bulk loading file for the CMS.

```
-->
<!--
```

PcspRelation.

The relation object specifies inter-object associations between the PcspCms, PcspService, PcspEndpoint, and PcspMta objects.

The "relOp" attribute specified if the relation is being added or deleted.

```
-->
```

```

<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>
<!--
```

The PcspCms object.

This object maintains associations between Endpoints, Services, and their managing CMSs.

Contents...

CmsFqdn - FQDN uniquely identifying this 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>
<!--
  The PcspEndpoint object.

  An endpoint is a physical port on a MTA/Gateway.

  Contents...

  EndpointId - Uniquely identifies this endpoint. Format per
    "IPCablecom Network Based Call Signalling Protocol Specification".
    Example: "aaln/1@mta01.cablelabs.com"

  AdminStatus -
    0: endpoint is disconnected
    1: normal - endpoint is in service
    2: test mode - endpoint is under test.

  Protocol - optional override for MTA protocol setting.

  Codec - optional override for MTA codec setting.

  IPSecControl - optional override for the MTA IPSecControl setting.
```

```
-->
<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" minOccurs="0"/>
      <xs:element name="Codec" type="codecType" minOccurs="0"/>
      <xs:element name="IPSecControl" type="xs:boolean" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
  The PcspMta object.
```

A Media Terminal Adapter aggregates one or more endpoints (physically contained within the MTA).

Contents...

MtaFqdn - MTA's FQDN, uniquely identifying this MTA.
MtaPort - MTA's NCS listening port (default: 2427)
CmtsFqdn - FQDN of controlling CMTS. CMS needs this to establish MTA DQoS with correct CMTS.
MtaProfile - MTA Profile Name - Optional; An MTA Profile Indicator identifiable by the CMS.
Timezone - within which this MTA is physically located. Optional; If present overrides the CMS default setting for the time zone. Per RFC 1123 numeric timezone format.
Protocol - Optional; If present it must be set to "MGCP 1.0 NCS 1.0". This is the default for all contained endpoints.
Codec - Optional: if present it is the default for all contained endpoints.
IPSecControl - Optional; NCS IPSec Control Flag (default = True; IPSec enabled)

```
-->
<xs:element name="PcspMta">
  <xs:complexType>
    <xs:sequence>
```

```

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

```

The PcspService object.

Contents...

ServiceId - unique identifier for the service.

AdminStatus -

- 0: suspended (i.e. bill not paid).
- 1: enabled (normal state).
- 2: number has changed.
- 3: out of service.
- 4: unassigned.

BillingId - A telephone number identifying another service to be billed instead of this service.

ExternalId - an arbitrary string used to carry such data as subscriber ID, etc.

IsPrimary - With multiple services provisioned upon an endpoint, one service must have this flag set to indicate the default service to use for outgoing calls.

- false/0: this service is not a primary service.
- true/1: this service is a primary service.

PrimaryRing - Primary Ringing Pattern ID. Index into MTA cadence table, selecting ring pattern for this service. Optional if the "Is Primary" flag is set to False. If not present, the CMS 务必 use its normal ring pattern

DisplayName - Used for Call Name Delivery feature (CNAM)

DisplayNumber - Used for Call Number Delivery feature (CND)

Password - various call features require a password before any alterations are permitted.

Network announcement control. See previous definition. Optional; if not present the CMS must use its default settings.

Inter-exchange codes and Local Number Portability settings. See previous definitions. Optional; if not present the CMS must not assign any inter-exchange codes to the service.

Call features. See previous definitions.

```

-->
<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:sequence>
  </xs:complexType>
</xs:element>

```

```

        </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" minOccurs="0"/>
<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" minOccurs="0"/>
<xs:element ref="InterExchange" minOccurs="0"/>
<xs:element ref="LNP"/>
<xs:element ref="ListOfCallFeatures"/>
<xs:element ref="Extension" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<!--
    Import/Export file format.
    Used to transfer one or more objects and relations between PS/CMS.

    NOTE: PccspCms is not included. There is currently no reason for a CMS to obtain its own CMS
    object from the PS.
-->
<xs:element name="PccspImportExport">
    <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="PccspService"/>
            <xs:element ref="PccspEndpoint"/>
            <xs:element ref="PccspMta"/>
            <xs:element ref="PccspRelation"/>
        </xs:choice>
    </xs:complexType>
</xs:element>
</xs:schema>

```

附件B

PCSP消息的WSDL规范

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    The IPCablecom CMS Subscriber Provisioning interface.
    Specified in Web Service Description Language 1.1.
-->
<definitions name="PccspI01Service" 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/">
<!-- NOTE: the parameter <unique_fully_qualified_namespace> must be replaced by fully qualified
unique identifiers for the actual implementation.
-->
<!--
    The <types> section defines custom datatypes required by the interface.

    PCSPi01 requires two custom datatypes:
        PccspArg (and array of)
        PccspObj (and array of).

    // PccspArg (pseudo code)
    //
    class PccspArg
    {
        // EntityName and key of a specific object.
        // Wildcard are currently not permitted.
        // Key is ignored when entity is PccspRelation.
        //
        String entityName;
        String key;

        // Reserved for future use. Set to 0 for now.
        //

```

```

    int flags;
}

// PccspObj (pseudo code).
//
class PccspObj
{
    // EntityName and key of the specific object.
    // Key is ignored when entity is PccspRelation.
    //
    String entityName;
    String key;

    // cmdStatus:
    // PccspObj as method output/result - must be set to one of the status codes specified
below.
    // PccspObj as input to Put() - must be set to one of the following:
    //     1, create new object
    //     2, modify existing object.
    // This field is ignored when entity is PccspRelation.
    //
    int cmdStatus;

    // XML encoding per PCSP Data Model Schema or 0 (null)
    //
    String xmlEncoding;
}

```

EntityNames; must be one of the following:

```

"PccspService"
"PccspMta"
"PccspEndpoint"
"PccspCms"
"PccspRelation"

```

Status codes: Used for method output or contained in the cmdStatus field of a PccspObj result (output).

```

0 , Operation succeeded
1 , Object not found
2 , Invalid Put() mode specified.
3 , Object creation failed, object already exists
4 , Read op failed
5 , Create op failed
6 , Modify op failed
7 , Delete op failed
8 , Internal problem.
9 , Server Busy
10, Unsupported operation.
11, Vendor extension.

```

...extended as needed...

```

-->
<types>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.IPCablecom.com/pcsp/i01">
    <complexType name="PccspObj">
        <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="ArrayOfPccspObj">
        <complexContent>
            <restriction base="soapenc:Array">
                <attribute ref="soapenc:arrayType" wsdl:arrayType="tns:PccspObj []" />
            </restriction>
        </complexContent>
    </complexType>
    <complexType name="PccspArg">
        <sequence>
            <element name="entityName" type="string"/>
            <element name="key" type="string"/>
            <element name="flags" type="int"/>
        </sequence>
    </complexType>
    <complexType name="ArrayOfPccspArg">

```

```

        <complexContent>
            <restriction base="soapenc:Array">
                <attribute ref="soapenc:arrayType" wsdl:arrayType="tns:PcspArg[]" />
            </restriction>
        </complexContent>
    </complexType>
</schema>
</types>
<!--
    Message section.

```

Invoking a method on the interface involves two "messages": an input message and an output message.

"In" contains the set of input args to the method call.

"Out" contains the return values.

```

-->
<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>
<!--
    Port type defines the interface.

```

Each "operation" is a method on the interface, with associated input and output messages (args and return values).

```

// The PCSP service interface (in pseudo code).
//
interface IPcspI01Service
{
    // Get (read) one or more objects from the server.
    // EntityName of 'PcspRelation' it not allowed (objects only)
    //
    PcspObj[] Get(PcspArg[] args);

    // Put (write) objects and relations to the server.
    //
    PcspObj[] Put(PcspObj[] objs);

    // Delete objects and relations from the server.
    //
    PcspObj[] Delete(PcspArg[] args);

    // Out-of-band command and status reporting.
    //
    // Predefined command codes:
    //     0 - extension command
    //
    // Predefined status codes:
    //     0 - extension status
    //
    int CmdStatust(boolean cmd, // true for CMD, false for STATUS.
                   int code, // CMD or STATUS code (see above).
                   int subCode // SubCode. Further refines code.
                   String extension);
}
-->

```

```

<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>
<!--
  Bind the interface ("portType") to transport specifics.
  Essentially, each method's input and output flow is bound as a
  remote procedure call using 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" namespace="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>
<!--
  The top level definition of the PCSP I01 Service.

  Note that the <service> element does not contain an address. It is assumed that the actual
  address of the service will be set explicitly within the client and server.
-->
<service name="PcspI01Service">
  <documentation>IPCablecom CMS Subscriber Provisioning Service I01</documentation>
  <port name="PcspI01Service" binding="tns:PcspI01Service">
    <soap:address location=""/>
  </port>
</service>

```

```
</port>
</service>
</definitions>
```

附录I

实体编码的范例

I.1 PccspService对象的范例

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Example Service object encoding.

  Default and "pcsp" namespace is set to PccspI01.
  "pcsp" namespace is a convenience, allowing vendor extensions
  to reference elements from the main PCSP schema.
-->
<PccspService xmlns="http://www.cablelabs.com/Pccsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pccsp/I01/schema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="PccspI01.xsd">
  <!--
    A sample Service object.
  -->
  <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 PcspEndpoint对象的范例

```

<?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">
  <!--
    A sample Endpoint object.
  -->
  <EndpointId>aaln/1@mta01.cablelabs.com</EndpointId>
  <AdminStatus>2</AdminStatus>
  <Protocol>MtaDefault</Protocol>
  <Codec>2</Codec>
  <IPSecControl>>true</IPSecControl>
</PcspEndpoint>

```

I.3 PcspMta 对象的范例

```

<?xml version="1.0" encoding="UTF-8"?>
<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">
  <!--
    A sample MTA object.
  -->
  <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>
  <IPSecControl>>true</IPSecControl>
</PcspMta>

```

I.4 PcspCms对象的范例

```

<?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">
  <!--
    CMS object.

    Not much defined yet...just its key.
    Serves as a collection for Services and Endpoints.
  -->
  <CmsFqdn>cma01.cablelabs.com</CmsFqdn>
</PcspCms>

```

I.5 PcspRelation 对象的范例

```

<?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">
  <!--
    A PcspRelation.

    This relation associates several Endpoints to the Service "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>

```

附录II

对象扩展的范例

II.1 扩展PcspService对象的范例

下述范例展示了PCSP方案的扩展能力。此范例中通过在对象主体中加入新的呼叫功能和若干新元素，扩展了PcspService对象。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  An example illustrating how to extend a Pcsp object.
  This example extends the PcspService object with additional
  fields and call features.

  See details below.
-->
<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">
  <!--
    The main body of the Service object is filled with sample data that will allow the object to
    validate.
  -->
  <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>
  <!--
    A Service object can be extended in two locations:
    1. The main body of the object.
    2. The call feature list.

    Here we extend the set of call features with the CfXYZ call feature.

    1. The VendorExt element must specify a valid namespace for the extension's schema. This
    allows the parsing system to locate the schema file for the extension.
    2. Any content within the VendorExt element must be namespace qualified, enabling validation
    against the extension's schema.
  -->
  <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>
  <!--
    Here, we extend the data content of main body of the Service object.
  -->
  <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 扩展方案

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  The schema for the sample PcspService extension.

  This schema defines several extensions:

  A, B, and C for the main body of the Service object.
  Call feature CfXYZ for the Service's call feature list.
-->
<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>
```

附录 III

数据编码评估

在对数据对象和消息编码时需考虑的选项（见表III.1）。

III.1 XML

XML是一种标准元语言，它允许各组织为文件出版和数据交换设计自己的标识语言。此类基于文本的标识：旨在使人和流程都能对它一目了然。XML提供：

- 开放、基于标准的、独立于平台的数据交换；
- 将数据输入内存的标准化分析程序；
- 处理数据的标准化接口（树型和流型）；
- 显示数据的标准方法；
- 查询数据的标准方法；
- 链接数据的标准方法；
- 对人员进行出版与数据处理方面的标准化培训。

代价：在某种程度上增加了编码规模和分析程序的开销。

XML规范由万维网联盟（W3C）的XML工作组监督。来自各个领域的特殊兴趣小组专家为此项工作做出了贡献。它是一种公共标准——并不是任何公司独自的研发项目。1.0版本的规范已于1998年2月10日，被W3C作为建议书接受。此规范可在<http://www.w3.org/TR/REC-xml>找到。

III.2 ASN.1/BER

ASN.1是一种用来表示任意数据结构的、基于文本且独立于平台的语法。它常被用于表示SNMP MIB。基本编码规则（BER）是一种简单递归算法，通过ASN.1说明产生紧凑型八位字节编码。BER将每个条目编写成一个标记，说明该条目的类型、说明对象规模的长度和包含对象实际内容的数值。

III.3 专用ASCII

专用编码不在此范围之内。

III.4 SDP（会话描述协议）

内容不灵活。主要用于描述流媒体能力。

III.5 RADIUS

RADIUS数据编码（TLV）过于原始且无法强制实施排序。RADIUS难以使用且限于码结构。

III.6 SQL

与特定的关系数据库实施/方案相连。有些厂商可能已经在数据库中部署了不兼容的方案。

III.7 可选方案一览表

表III.1/J.177 – 数据编码的可选方案

可选方案	优点	缺点
XML	灵活，基于ASCII标记。 用方案 / DTD进行语法检查。 易于扩张而不会影响传输。 允许厂商进行扩展。 独立于平台。 不依赖某种语言。	不安全，需要安全的传输层。 比二进制编码使用的CPU和网络容量大。分析时间等。
ASN.1/BER	数据格式化的分层结构。定义描述数据格式（或“方案”）的语言。 数据结构可以嵌套。 数据的格式采用独立于平台的方式。 如果设计中包含的格式版本确定方式可使应用知道应使用哪种格式分析数据内容，则格式可以扩展。	ASN.1不易扩展。 格式版本的反向兼容性，难于集成在设计之中，且不易实施。 对于各个方案，大部分实施使用经编辑的二进制层分析程序，而这意味着难以定义灵活的应用。 由于很小的格式错误都会造成数据“包”无法分析 / 不可读，因此调试应用及其互操作性很难。
专用ASCII	超出范围	超出范围
SDP （会话描述协议）		内容不灵活。 主要用于描述流媒体能力。

可选方案	优点	缺点
RADIUS		RADIUS数据编码（TLV）过于原始。 无法强制实施排序。 难用且限于码结构。
SQL		与关系数据库方案的实施相连—有些供应商可能不会使用。

III.8 建议：XML

XML提供平台不可知且技术中立形式结构消息和打包数据。它是在异构应用之间发送数据的上佳选择，因为各应用无需了解其它应用的专用格式。由于XML是一种结构性语言，它适用于分层类型的消息。数据可以很容易地映射到各元素，因此XML文件（作为树形结构）可以负责分层的维护。代价：增加了线路中的有用负荷，并延长了对象的编组时间（分析）。

附录IV

传输协议评估

IV.1 具有IPSec的TFTP

在IPCablecom基础设施（DOCSIS）中，小文件传输协议（TFTP）已经投入使用。此文件协议专为传输小型文件而设计。

IV.2 批处理RADIUS – 通过事件消息在单一的请求中做多个记录

RADIUS是一种主要用于处理互联网拨号鉴权、授权和计费的IETF标准。目前，RADIUS是大部分路由器制造商对此类操作采用的实际标准。有些IP电话网关设备制造商已经使用RADIUS支持厂商进行扩展，以便提供计费所需的信息。

RADIUS既定义了传输协议，也定义了消息格式的规范。作为传输协议，RADIUS依靠用户数据报协议（UDP）进行消息广播，而且是以端口为基础的。

作为一种消息格式，数据的格式化是基于标记长度值（亦称属性长度值）。标准鉴权、授权和计费标记均已预定义且仅需使用最小值。但可以在不影响现有协议实施的前提下增加新的属性。RADIUS全部消息的最小长度为20个字符，最大长度为4096个字符。独立的数据字段支持247字节的数据，例如247字节的URL或文件名。

RADIUS的可靠性很差，基本没有差错恢复功能，而且用于新标记的数量也很有限。（它总共只能定义255个标记，而与之相比，某些PSTN第5类交换机目前就有600种功能。）

有关RADIUS的详细信息，请参见：

<http://info.internet.isi.edu/in-notes/rfc/files/rfc2139.txt>

http://www.livingston.com/marketing/whitepapers/RADIUS_paper.html

IV.3 Diameter

Diameter是IETF工作组中的一项活动，设计上与RADIUS反向兼容。此功能可扩展性更好，具有更大的安全性优势，设计上寻求最大限度地减少配置。此外，通过公钥等各类安全机制，它还可以很好地支持跨域AAA。Diameter支持向备份服务器的故障切换（为低故障要求的环境设计（99.99+））。参见IETF RFC Diameter 基础协议。

RADIUS/DIAMETER不提供双向通信（仅提供确认），因此不能满足这些要求。

IV.4 分布式对象系统

IV.4.1 CORBA/IIOP

分布式对象技术受到对象管理组的支持。有多达800家OMG成员支持此项技术。

通用对象请求代理体系结构（CORBA）允许在应用之间使用对象请求代理（ORB）进行通信。ORB是一种在对象之间建立客户机—服务器关系的中间件。ORB负责注册客户机，并管理公布、订阅和收听等权利。利用ORB，客户机可以透明地为服务器对象调用一种方法，且该对象可以在同一设备上或在整个网络中。ORB监听本地呼叫，并负责查找能够实施此请求、为其发送参数、调用其方法并返回结果的对象。

接口定义语言（IDL）用于建立客户机与服务器对象之间的ORB协议契约。ORB基本会对程序员隐藏传输的细节。IDL被编辑为C++、Java等语言，实施客户机与服务器的桩（stub），处理所有客户机与服务器间使用的IIOP传输协议所需的数据编码/解码事务。

CORBA将处理查找方法调用服务器，从客户机设备向服务器设备传输自变量，以及将所有返回代码传回客户机设备的细节工作。

目前，来自多家厂商的ORB可用于三十多种硬件平台和操作系统。CORBA在Unix平台上特别流行。但实际上ORB厂商在功能上展开了竞争。在基本功能（安全性等）问题解决之后，互操作性的问题会长期存在。两个随机选择的ORB之间成功通信的可能性很低。从发展的角度看，CORBA可能会变得非常复杂。此外，CORBA是一种相对昂贵的选择（运行时间和开发许可证）。

IV.4.2 DCOM

微软的分布式组件对象模型。与CORBA共享下列特性：

- 将对象接口与实施分开。通过MIDL（微软的IDL变体）实现。
- 允许实现位置的透明。客户从远程对象调用方法，而不知道远程对象在哪台机器上运行。
- 统一的例外处理机制（DCOM方法调用会返回一个恒定的（flat）HRESULT返回状态）。

但是：

- DCOM是基于与CORBA使用的IIOP传输协议不兼容的DCE ORPC传输协议。

- DCOM基本上是一种只使用微软产品的技术。它是Win95、Win98和NT平台上的标准配置。

IV.5 HTTP

超文本传输协议（HTTP）是万维网的基本协议。HTTP的设计是为了将异构数据源连接在一起，以创建一个分布式的信息系统。其设计也考虑到了扩展性的要求。在典型的HTTP事务处理中：

- 1) 客户机建立与服务器的连接；
- 2) 客户机向服务器发出请求（带URL参数）；
- 3) 服务器发送包含状态和请求的URL的响应；
- 4) 任何一侧均可断开。

HTTP为客户机请求和服务器响应提供事务处理报头。客户机事务处理报头中可包括帮助向客户机交付其所需信息的参数（例如，数据格式、语言等的类型）。服务器报头中可包括给出响应信息的参数（例如，请求的状态（如返回代码）、发送数据的长度、内容类型、内容的语言等）。

根据网络目前的状态，HTTP无处不在；且可与防火墙共用。

IV.6 可选方案摘要

见表IV.1。

表IV.1/J.177 – 传输的可选方案

可选方案	优点	缺点
使用IPSec的TFTP	体积小。 已对DOCSIS实施。	不提供双向通信。
RADIUS	灵活：包括厂商专用和客户专用字段。IPCablecom可以定义这一空间中的字段。 为多种IP电话计费系统所用。 已在路由器等IP组件中广泛部署。	不是所有RADIUS产品都支持AAA。 应用层需要处理可靠性问题。 内置安全性不足，需要独立的可信协议或与其它边缘路由器的共享保密密钥。
基于套接字的带SSL的专用协议		专用。如果深究，我们最终写出的可能是SOAP或XMLP已有的大部分功能。
CORBA/IIOP	易于实施，因为名称解析细节，将打包参数并入消息以及传送，均由CORBA基础设施管理。	可能会是一种更昂贵的解决方案。 CORBA基础设施必须开发或购买，且在这两种情况下都必须与应用一同部署。 可能还会存在互操作或不同CORBA/ORB产品引起的问题。

可选方案	优点	缺点
DCOM	DCOM和CORBA/IIOP是类似的技术。	基本上是一种只使用Microsoft Windows的可选方案。 不会与CORBA-IIOP进行互操作。
HTTP	<p>HTTP已作为数据源（网络服务器）与数据用户（客户机浏览器）之间进行交换的下层传输协议，被广泛部署。</p> <p>基于简单事务处理的数据传输。</p> <p>简单、无状态（stateless）、基于ASCII文本的协议。</p> <p>允许通过最新部署的网络基础设施（防火墙等）进行便利的数据交换。</p> <p>在提出GET、POST、HEAD、PUT、DELETE、LINK和UNLINK等请求时，客户机可以使用简单的方法调用。</p>	<p>由于是Stateless，该协议在事务处理结束后不保留对事务处理的记录。</p> <p>它能否跟上要求的CMS事务处理速率？</p> <p>就带宽和处理需求而言，相对昂贵。</p>

IV.7 建议：HTTP 1.1

基于上述分析，建议使用HTTP 1.1作为传输协议。

参考书目

- IETF RFC 2139 (1997) , *RADIUS Accounting*.
- 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.
- Telcordia Local Exchange Routing Guide (LERG) , *Telcordia Technologies, Inc.*

ITU-T系列建议书

A系列	ITU-T工作的组织
D系列	一般资费原则
E系列	综合网络运行、电话业务、业务运行和人为因素
F系列	非话电信业务
G系列	传输系统和媒质、数字系统和网络
H系列	视听及多媒体系统
I系列	综合业务数字网
J系列	有线网络和电视、声音节目及其它多媒体信号的传输
K系列	干扰的防护
L系列	电缆和外部设备其它组件的结构、安装和保护
M系列	电信管理，包括TMN和网络维护
N系列	维护：国际声音节目和电视传输电路
O系列	测量设备的技术规范
P系列	电话传输质量、电话设施及本地线路网络
Q系列	交换和信令
R系列	电报传输
S系列	电报业务终端设备
T系列	远程信息处理业务的终端设备
U系列	电报交换
V系列	电话网上的数据通信
X系列	数据网、开放系统通信和安全性
Y系列	全球信息基础设施、互联网协议问题和下一代网络
Z系列	用于电信系统的语言和一般软件问题