TR069 修正版 1

CPE WAN Management Protocol

CPE 广域网管理协议

2006 年 12 月

编辑

Jeff Bernstein, 2Wire Heather Kirksey, Motive
Tim Spets, Westell William Lupton, 2Wire
John Blackford, 2Wire Anton Okmianski, Cisco
Mike Digdon, SupportSoft

译

陈科锦

TSV1.0

本规范只对英文协议的主体部分进行了翻译，不包括附录部分。

为了让你更好地掌握本协议，强烈建议在阅读本规范后再次阅读英文协议原本，或把本规范作为阅读英文协议时的参考。

欢迎各位给本规范翻译的不当之处提出你的修改建议或意见。

Email：hz.chenkejin.study@163.com

概要

CPE 广域网管理协议描述了 CPE 和自动配置服务器 ACS 之间通信的一个公共平台，它包括了可靠的自动配置以及一些其他的 CPE 管理功能。

1 介绍

本文档描述 CPE 广域网管理协议是用于 CPE 和自动配置服务器 ACS 之间的通信协议。本协议定义了一套机制，包括 CPE 的可靠的自动配置，同时将一些其他的 CPE 管理功能合并到同一个公共的平台框架上。

本文档描述了可以应用于所有的 TR-069 CPE 的管理协议 method 的通用需求。而其他一些文档是针对特定的管理实体或数据模型，针对特定类型设备或服务进行描述。

1.1 功能

CPE 广域网管理协议支持很多用于管理 CPE 群的功能，主要包括如下：

a) 自动配置和提供动态业务

b) Sofware/firmware 镜像文件的管理

c) 状态和性能监控

d) 诊断

1.1.1 自动配置和提供动态业务

CPE 广域网管理协议允许 ACS 给一个 CPE 或一组基于多种标准的 CPE 提供服务。

这个服务提供机制允许 CPE 既可以在刚刚连接到宽带接入网时，也可以在以后的任何时间来进行动态服务提供或重新配置。此机制包括支持在异步状况下 ACS 发起的服务。

此协议包含的鉴权机制允许为每一台指定的 CPE 或基于相同的 CPE 供应商、型号、软件版本或其他标准的 CPE 组提供服务。

本协议同时也提供了一些可选的功能来管理那些有特殊级别安全需求的应用或业务，例如付费服务等。这种可选的数字签名凭证的控制机制在附录C中描述。

此服务机制使得将来把那些还没有包括在本规范中的服务和功能能够很容易的扩充进来。

1.1.2 Software/Firmware 镜像文件的管理

CPE 广域网管理协议提供了对软硬件镜像文件下载的管理。此协议提供了版本号鉴权机制，文件下载触发机制（ACS 发起下载或可选的 CPE 发起下载），以及对 ACS 的文件下载失败或成功的通知机制。

CPE 广域网管理协议定义了一种数字签名文件格式，可以用于下载一个独立的文件，或者下载一个含有需要 CPE 执行的外部安装指令集的文件包。这种包格式保证了下载文件的完整性和跟安装指令的关联性，且能够鉴定一个文件是否只是 ACS 操作的一部分。

1.1.3 状态和性能监控

CPE 广域网管理协议为 CPE 生成有效的监控信息提供支持，ACS 可以使用这些信息来监控 CPE 的状态和性能统计。同时，本协议也提供了一套允许 CPE 动态通知 ACS 自身状态更新的机制。

1.1.4 诊断

CPE 广域网管理协议为 CPE 生成有效的诊断信息提供支持，ACS 可以使用这些信息来诊断和解决连通或业务的问题，同时提供执行预先定义好的诊断测试能力。

1.1.5 Web 应用的身份管理

为了支持从 CPE 本地网络通过 Web 浏览器访问基于 Web 的应用，CPE 广域网管理协议定义了一套允许这些 web 站点根据对应的 CPE 的一些明确信息来定制它们的内容的机制。此机制的详细描述参见附录 D。

1.2 端到端架构中的布置

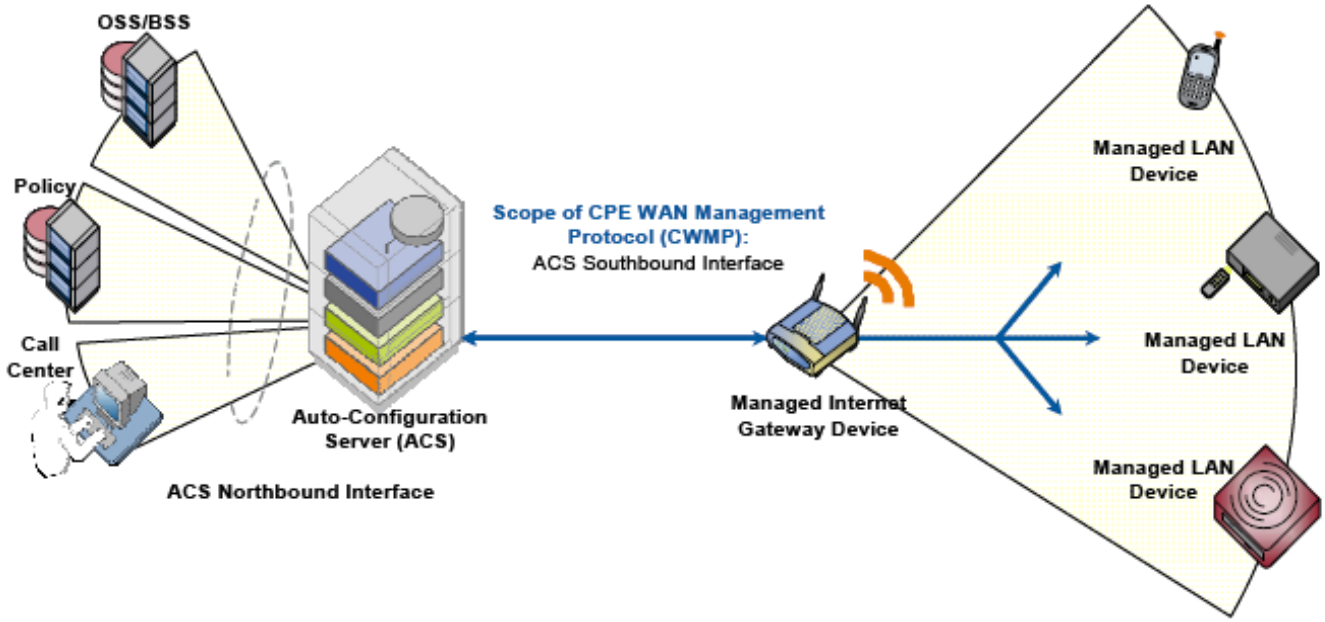
ACS 是在网络中的服务器，根据预定制来管理设备。CPE 广域网管理协议可以用于管理 DSL B-NT 以及其他类型的 CPE,包括路由器和 LAN 侧的客户端设备。它被服务供应商用于基于跟设备已经建立的 IP 层链接的特殊媒体接入上。

注 1：关于 B-NT，TR-046 描述了一个关于 B-NT 自动配置的全面的平台框架，同时 TR-062 和 TR-044 定义了 ATM 层和 IP 层的自动配置过程。其他类型的宽带 CPE 应根据自身网络架构对应的协议来建立 IP 链接。

注 2：CPE 广域网管理协议是用于管理包括 B-NT(或其他 internet 网关设备)，以及那些运行在 B-NT（或其他 internet 网关设备）背后的 LAN 侧客户设备，附录 F 描述了一个让 ACS 把这

两个关联在一起的机制以用于统一管理。

Figure 1 – Positioning in the End-to-End Architecture



1.3 安全目标

CPE 广域网管理协议提供了高级别的安全性。这个安全模型也是可升级的。在允许更高级别的安全机制的同时也为那些较低健壮性的 CPE 操作提供基本的安全性。概况地说，CPE 广域网管理协议的安全目标如下：

a) 防止对 CPE 或 ACS 管理功能，或对 CPE 和 ACS 之间事务处理的篡改。

b) 保证 CPE 和 ACS 之间事务处理的机密性。

c) 运行为每一种事务处理提供适当的鉴权。

d) 阻止对服务的窃取。

1.4 架构目标

本协议对连通模型上有很好的灵活性：

a) CPE 和 ACS 都可以发起建连，避免在 CPE 和 ACS 之间维护一个长期的连接。

b) ACS 和 CPE 之间的功能交互应当不受这个连接是由哪端发起的这个条件的约束。特别地，即使在 ACS 不支持发起建连的情况下，所有的 ACS 发起的业务应当能够在 CPE 发起的

连接上正常处理。

c) 允许一个或多个 ACS 为一个 CPE 组服务，这些 CPE 可以根据一个或多个业务供应商信息关联在一起。

本协议支持 ACS 和 CPE 的 discovery 和 association：

a) 提供机制让 CPE 去搜寻既定业务供应商对应的 ACS 的机制。

b) 提供机制允许 ACS 安全地识别一个 CPE 并把它跟用户/客户关联起来。支持这种关联的处理需要支持完全自动化的用户交互的模型。

本协议允许 ACS 可以控制和监控各种个 CPE 关联的参数。提供访问这些参数的机制设定了如下的前提：

a) 不同的 CPE 可以拥有不同的能力级别，执行可选功能的不同子集。另外，ACS 可以管理一些不同类型的设备并为他们提供一些不同的业务。ACS 必须能够找出某个特定 CPE 所拥有的能力。

b) ACS 必须能够控制和检测 CPE 的当前配置。

c) 除了 ACS 之外的一些其他的控制实体可能也能够控制一台 CPE 设置的一些参数（例如，通过 LAN 侧的自动配置）。本协议必须允许 ACS 来估算 CPE 配置的外部变化。同时，ACS 也应该能够控制这些能够通过其他方式被控制的配置参数。

d) 本协议应该允许供应商特定参数的定义和访问。

当面临复杂性和功能的选择时，本协议希望执行的复杂度能够最小化。本协议仅当特定功能被要求时才加入一些可选组件。The protocol also incorporates existing standards where

appropriate, allowing leverage of off-the-shelf implementations.

The protocol is intended to be agnostic to the underlying access network.

本协议是一个可扩展的协议，它包括支持将来对本标准的扩展的机制，同时也包括供应商特定要求的扩展的机制。

1.5 假定

以下是 CPE 广域网管理协议定义时的一些假定：

a) 所有的 CPE，不管它时什么类型的（桥接，路由或其他），需要获得一个 IP 地址用于跟 ACS 直接的通信

在同一时间，一台CPE只能跟一个ACS进行交互。在任何时候，CPE需要知道一个它可能正确连接的ACS（注：在一个载荷均匀后面的一组ACS应被认为是一个ACS)。

1.6 术语

略。

2 架构

2.1 协议组件

CPE 广域网管理协议包含一些该协议特有的组件，同时也使用了一些标准协议。图 2 是 CPE 广域网管理协议的协议栈描述。表 1 是每一层的摘要描述。注意，除非有其他的详细规定，CPE 和 ACS 必须遵循以下标准协议的需求定义。

图 2——协议栈

CPE/ACS Management Application
RPC Methods
SOAP
HTTP
SSL/TLS
TCP/IP

表 1——协议层摘要

CPE/ACS Application	该应用程序分别用于 CPE 广域网管理协议的 CPE 和 ACS。不属于 CPE 广域网管理协议的一部分。
RPC Methods	CPE 广域网管理协议定义的特殊的远程过程调用方法，详见附录 A
SOAP	标准的基于 XML 的语法，这里用于对 RPC 编码，要求支持 SOAP 1.1
HTTP	要求支持 HTTP 1.1
SSL/TLS	标准的 internet 传输层安全协议。特指 SSL3.0，或 TLS1.0
TCP/IP	标准的 TCP/IP

2.2 安全机制

CPE 广域网管理协议在交互过程中拥有高度机密性。它防止了对 CPE 和 ACS 之间事务处理的篡改，保证这些事务处理的机密性，同时允许各种级别的鉴权。

本协议有如下的安全机制：

a) 该协议支持 CPE 和 ACS 之间通信时使用 SSL/TLS。它保证了事务处理的机密性，数据的完整性，以及 CPE 跟 ACS 之间的基于证书的鉴权。

b) HTTP 层提供了一种基于共享密钥的 CPE 鉴权或 ACS 鉴权。本协议不描述 CPE 和 ACS 如何学习此共享密钥。

本协议包括一些其他的安全机制，associated with the optional signed Voucher mechanism and the Signed Package Format,在附录A和附录E中分别作了描述。

2.3 架构组件

2.3.1 参数

RPC 方法规范（附录 A）定义了一种通用机制，ACS 可以根据它来读写参数以达到配置 CPE 和监控 CPE 的状态和统计信息。不同类 CPE 的参数定义在不同的文档中。目前为止有如下的 TR-069 数据模型标准：

a) TR-106: Data Model Template for TR-069-Enabled Devices

b) TR-098: Internet Gateway Device Data Model for TR-069

c) TR-104: Provisioning Parameters for VoIP CPE

每一个参数包括一个“名-值”对。名字标识了特定的参数，并且拥有一个类似于文件目录的分级结构，每一级之间用一个点‘.’分隔开。参数的值可能是多种已定义的数据类型中的一种。

参数可以被设置成只读或只写。只读参数可以让ACS用于确定特定CPE的特征，观察CPE的当前状态，或者收集统计信息。可写参数可以让ACS去定制各种CPE操作。所有可写参数必须是可读的，虽然那些包含用于机密性信息，例如密码，可能会在读时返回一个空值（在数据响应规范里面有描述）。一些可写的参数能够被那些不同于本规范定义的接口所独立的更改（例如，一些参数也可以通过LAN侧的自动配置协议所修改）。

可能因为一些其他协议（也或者用户行为）独立地改变了设备的配置，ACS不能假定自己是唯一一个可以更改设备配置的实体。另外的，某个LAN侧的机制可能会改变设备的配置，它可以以这样的方式来违反ACS提供的配置。需要特别小心的处理WAN和LAN侧的自动配置机制以及用户接口，从而来限制这种情况的发生。

本协议提供一种discovery机制来使得ACS确定一台特定的CPE能够支持那些参数，允许对可选参数的定义，也支持对将来的标准参数的扩展。

本协议也包括一种扩展机制，它允许除了使用那些定义在本规范内的参数之外，还可以使用供应商特定的参数。

2.3.2 文件传输

RPC 方法规范（附录 A）定义了一种使得文件下载或为了各种目的的上载（例如 firmware 上载或供应商特定的配置文件的上载）变的很容易的机制。

当由 ACS 发起时，CPE 得到了需要传输的文件的地址，使用 HTTP/HTTPS，或者，可选的使用 FTP/SFTP/TFTP 作为它的传输协议，然后 CPE 就开始执行本次传输，同时通知 ACS 传输失败或成功。

下载也有可能由 CPE 发起。在这种情况下，CPE 首先向 ACS 请求一个特殊类型文件的下载，然后 ACS 跟 ACS 发起下载时的操作一样开始响应这次下载。

CPE 广域网管理协议同时也定义了一种可选的用于下载的数字签名文件格式。此签名报格式的定义详见服务 E。

2.3.3 CPE 发起会话

RPC 方法规范（附录 A）定义了一种允许 CPE 向对应的 ACS 通报各种状态的机制，同时保证了 CPE-ACS 通信发生的最小频率。

它包括 CPE 初始安装时发起的建立通信的机制，用于‘bootstrap’（引导）初始的客户参数到 CPE 中去。同时它也包括在正在进行的基础上建立和 ACS 周期通信的机制，或者事件发生时必须通知 ACS（例如当 CPE 的宽带 IP 地址发生改变时）。

在任何一种情况下，当通讯建立后，CPE 用工厂信息和序列号信息（以及可选的产品识别符）来唯一地标识自己，这样，ACS 就可以知道是那个 CPE 跟他通讯并用一种适当的方法来响应。

2.3.4 ACS 发起的异步会话

服务自动配置的一个重要方面是 ACS 拥有异步地通知 ACS 改变配置的能力。这将允许使用自动配置机制的服务能够用一种近似实时的方式重新配置 CPE。例如，这可能被用来提供给一个终端用户来立即接入或使用一个定制的服务或特征，而无需等待下一个通知间隔周期。

CPE 广域网管理协议包含一种可以让 ACS 在任何时候向 CPE 发起连接请求的机制，通知 CPE 向 ACS 建立一个通讯会话。

同时 CPE 广域网管理协议也允许用 CPE polling 来代替 ACS 发起连接，CPE 广域网管理协议不会依靠 polling 或建立 CPE 的永久连接来提供异步通知。

CPE 广域网管理协议定义的用于激活异步 ACS 发起的通讯的基本机制假定 ACS 能够直接寻址到 CPE 的 IP 地址。另一个可选的在附录 G 中描述的机制用于 CPE 工作于一个 NAT 网关后面使得 ACS 无法直接寻址它的情况。

3 规程和必要的条件

这一节，包括它所提到的参考附录，定义了 CPE 广域网管理协议的标准化需求。

这一节也参考了一些其他的标准和规范，作为 CPE 广域网管理协议的组成部分。除非有另外的规范说明，CPE 和 ACS 必须遵循这些参考规范的需求。

3.1 ACS Discovery

CPE 广域网管理协议定义了以下机制可以让 CPE 用于搜寻它对应的 ACS 的地址：

a) CPE 可能在本地配置好 ACS 的 URL。例如，它可以通过 LAN 侧的 CPE 自动配置协议来实现。CPE 通过 DNS 用 URL 中的 host name 组成部分找到 ACS 的 IP 地址。

b) 作为 IP 层自动配置的一部分，接入网的 DHCP 服务器可以把 ACS URL 作为 DHCP option[14]来配置给 CPE。然后 CPE 通过 DNS 用 URL 中的 host name 组成部分找到 ACS 的 IP 地址。在这种情况下，另一个 DHCP option 可能被用于设置 ProvisioningCode，用于指示 ACS 的服务供应商信息和其他 provisioning 信息。

   一个 CPE 通过在 Vendor Class Identifier（DHCP option 60）的任何地方包含“dslforum.org”（小写）字符串来向 DHCP 服务器标识自己能够支持这种 method。

   CPE 可以用从 DHCP 服务器上收到的 Vendor Specific Information(DHCP option 43)中的值来设置表 2 中列举的相应的参数。这个 DHCP option 是按照一列“一个或多个 Encapsulated Vendor-Specific Options”的格式（定义[14]）进行编码。这一列里面还可能包括其他的没有在这里列举出来的供应商特有选项（vendor-specific options）。

   如果 CPE 通过 DHCP 获取到一个无法到达的 ACS URL，CPE 必须发起一个 DHCP Inform 来重新搜寻 ACS URL。CPE 如果无法跟 ACS URL 对应的每个 IP 地址在 300 秒之内建立 TCP 连接，则 CPE 必须认为此 ACS 不可达。如果 CPE 没有收到 DHCP Inform 的响应，它必须按照 RFC2131 来尝试重新获取。

   当 CPE 需要联系 ACS，它必须在下面的情况下来使用 DHCP discovery 机制：

   i) 如果 CPE 拥有一个空值的 ManagementServer.URL 参数值，或者

   ii) 如果 CPE 不能联系到 ACS 并且 CPE 最早（最近出厂缺省值后的第一次成功获取）是通过 DHCP 获取到它的 ACS URL。

   此行为使得 CPE 在没有预先配置 ACS URL 的情况下回去使用 DHCP 来获取 ACS。例如，它可以用于处理当在 CPE 里面设置了错误的 ACS URL 时。这个行为不能认为是一个 ACS 错误恢复机制。

CPE 必须记住它第一次成功地同 ACS 建立联系的机制。如果 CPE 没有用 DHCP 去搜寻 ACS URL，那么它将不应该退回去用 DHCP 来搜寻 ACS。如果 CPE 原先用 DHCP 进行 ACS 搜寻，那么一旦它跟 ACS 失去联系，它必须通过 DHCP 来重新搜寻。这个 last requirement 一直保留着即使 ACS URL 在后面已经被非 DHCP 机制所重新设置。

Table 2 – Encapsulated Vendor Specific Options

Encapsulated Option	Encapsulated Vendor-Specific Option number	Parameter[2]
URL of the ACS	1	...ManagementServer.URL
Provisioning code	2	...DeviceInfo.ProvisioningCode

这个指定的 URL 必须是一个完整的 URL。URL 以及 ProvisioningCode 都必须不能以 null 来结束。如果 CPE 收到一个 URL 或 ProvisioningCode 值时是以 null 为结束的，CPE 必须要接受这个值，但是必须不得将 null 字符解析为 URL 或 ProvisioningCode 的值的一部分。

c) CPE 有一个默认的 ACS URL 在没有其他的 URL 提供时使用。

这个 ACS URL 必须是一个有效的 HTTP 或 HTTPS URL 的格式。使用 HTTPS URL 表示 CPE 必须跟 ACS 建立一个 SSL 或 TLS 链接。

一旦 CPE 跟 ACS 建立了连接，ACS 可以在任何时候修改存储在 CPE 中的 ACS 地址参数（…ManagementServer.URL，见定义[13]）。一旦修改成功，在以后的跟 ACS 的连接中，CPE 必须使用这个修改过的地址。

当使用证书鉴权机制时，ACS URL 的“host”被 CPE 用于跟 ACS 进行证书认证。因为这个依赖于 ACS URL 的正确性，整个协议的安全性就由 ACS URL 的安全性而定了。

CPE 应该通过严格的安全验证机制来限制本地配置 ACS URL 的能力。CPE 应把本地设置 ACS URL 的能力更加严格地限制在初始建连时，一旦跟 ACS 首次成功建立连接后，只允许 ACS 来修改这个 URL 而不允许本地配置。

DHCP 配置 ACS URL 的使用应该受限于 DHCP 服务器和 CPE 之间的能够被服务供应商所确保的连接安全性状况所决定。由于 DHCP 自身没有安全机制，其他的安全机制将被用于确保此操作的安全性。

ACS URL 可能包含一个 DNS 主机名或一个 IP 地址。在寻址 ACS 主机名时，DNS 服务器可能返回多个 IP 地址。在这种情况下，CPE 应该从 IP 地址列表中随机地选择一个 IP 地址。当 ACS 不可达，CPE 需要随机地从列表里面选择另一个 IP 地址并尝试用这个新 IP 地址跟 ACS 建立连接。此过程能够确保当多个 IP 地址对应多个 ACS 时，CPE 能够匀称地把请求发送给这些不同的 ACS。

CPE 必须不能存储来自 DNS 服务器的超过生存时间周期（TTL）的响应，并在一个刷新周期内不得向 DHCP 服务器发起联系。此过程为 DNS RFC1034 的需求，它给 DNS 服务器提供了一个更新已经失效的信息的机会。

本协议进一步建议 CPE 的操作应该绑定于一个特定的 ACS IP 地址。绑定于一个给定的 IP 地址是指 CPE 只要能通过这个地址跟 ACS 建立连接，CPE 将会一直使用这个 IP 地址。这样形成了一个更稳定的体系，同时能使 CPE 更好地运行于一个更好的缓存中。为了实现这个绑定，CPE 需要在永久性存储器中存贮它最后一次成功建连的 IP 地址以及整个它用于挑选的 IP 地址列表。CPE 应该一直用同一个能够跟 ACS 建立连接的 IP 地址正常地继续执行 DNS 询问机制，直到 DNS 返回的 IP 地址列表发生了变化。当 IP 地址列表发生了变化或用此 IP 地址无法跟 ACS 建立连接时，CPE 应该选择一个新的 IP 地址。这个过程给服务供应商提供了一个重配他们网络的机会。

端口 7547 已经被 IANA 指定给 CPE 广域网管理协议（见[17]），ACS 应该在 URL 中使用这个端口。

3.2 建立连接

3.2.1 CPE 发起连接

CPE 可以在任何时候使用预定的 ACS 地址向 ACS 发起建连（见 3.1）。在下面的情况下，CPE 必须向 ACS 发起建连并发送 Inform RPC method（下面的规程描述见 3.7）：

a) CPE 初始安装时第一次向接入网发起建连

b) 上电或重启

c) 在每一个通知周期间隔（PeriodicInformInterval，例如，每 24 小时）

d) 当被 optional ScheduleInform method 所指示需要如此执行时

e) 一旦 CPE 接收 ACS 的有效链接请求时

f) 一旦 ACS 的 URL 发生改变时

g) 一旦一个在发生改变时需要发起一个 Inform 的参数发生了改变时

h) 一旦那个被 ACS 通过 SetParameterAttributes method 所标识为“active notification”的参数由于一个外部原因（非 ACS 自己）发生改变时。由 ACS 自己通过 SetParameterValues 使得参数发生的改变必须不能引起一个新的会话的发起。如果一个参数在 CPE 能够成功的发起会话执行这个通知之前发生了多次修改，CPE 只需上报一个修改通知。

   如果一个参数在会话运行过程中由于内部因素发生了改变，一个新的会话将会在本会话结束后发起并建连（它必须不能影响当前会话）。

   为了避免对 ACS 有过多的业务量，CPE 可以在参数修改通报的频率上做一些本地特殊限

制。定义了此限制后只有在异常环境下才会超出这个限制。一旦此限制被超出，CPE 应该延迟一个用于通知 ACS 的会话的本地指定的会话开始期限。在这个延迟后，CPE 必须发起一个到 ACS 的会话并把从上一次通报以后发生的所有相关参数的修改（那些已经被标注为需要通知的参数）通报上去。

i) 一旦一个不正常终结的会话根据会话重新建立规程（见 3.2.1.1）发起重建时。

当 CPE 或 ACS 没有存在等待处理的消息时，CPE 必须不能继续维护这个到 ACS 的链接。CPE 会话终止规范详见 3.7.1.4。

3.2.1.1 会话重建规程

CPE 必须尝试重建失败的会话来重新递交先前递交失败的事件，同时允许 ACS 用一种适当的方式来发起其他的请求。3.7.1.5 详细描述了成功递交事件、重新递交事件、并在递交失败时丢弃事件的规则。CPE 必须记录它尝试重建一个失败会话的次数。

CPE 必须在等待一个表 3 中指定的时间间隔或当一个新的事件在此时间间隔内发生时尝试重建那个异常会话。CPE 必须从 post-reboot session retry count 给定的一个范围中随机地选择一个值（秒级）来作为等待间隔时间。当在一个重启间隔后对这个失败的会话发起重建尝试时，CPE 必须清除那个它选取的用于第一次会话重建尝试的等待间隔时间。换句话说，如果因为发生一个 BOOT 之外的新事件使得会话开始重建尝试的话，它将不清除这个等待间隔时间，虽然这个连续发生的新事件可能比表项中定义的时间更加频繁地导致会话的发起。不管原先这个会话是由于什么原因导致失败或者尝试重建的状态如何，CPE 必须把会话尝试重建次数传递给 ACS。

从第十个 post-reboot 会话重试开始，CPE 必须从 2560 到 5120 秒内选择一个值。CPE 必须尝试重建一个会话直到此会话能够成功终止。一旦一个会话成功终止，CPE 必须清除会话重建尝试次数，同时将不再启动会话重建规程来决定何时启动下一个会话。

Table 3 – Session Retry Wait Intervals

Post reboot session retry count	Wait interval range (min-max seconds)
#1	5-10
#2	10-20
#3	20-40
#4	40-80
#5	80-160
#6	160-320
#7	320-640
#8	640-1280
#9	1280-2560
#10 and subsequent	2560-5120

3.2.2 ACS 发起连接

ACS 可以在任何时候使用连接请求机制请求 CPE 发起一个到 ACS 的连接。CPE 需要支持此机制，同时在 ACS 侧建议支持此机制。

此机制依赖于 CPE 要有一个可被 ACS 路由到的 IP 地址。在 ACS 和 CPE 之间，如果 CPE 是在一个防火墙或 NAT 欺骗设备后面，ACS 将完全无法连接到这个 CPE。附录 G 描述了一个使得 ACS 可以通过一个 NAT 设备跟 CPE 相连接的机制。

连接请求机制定义如下：

a) 连接请求必须是一个到 CPE 指定的 URL 的 HTTP1.1 GET。这个 URL 值在 CPE 上面是一个有效的只读参数。这个 URL 的通道值应该是有 CPE 随机生成的，所以它是唯一地对应某个 CPE。

b) 连接请求必须使用 HTTP，而不是 HTTPS。对应的 URL 必须是一个 HTTP URL。

c) 连接请求的 HTTP GET 里面不应有数据。CPE 应该忽略里面所有可能存在的数据。

d) CPE 在执行之前必须使用摘要鉴全（digest-authentication）来对 ACS 进行认证——CPE 必须不能因为一个不成功的认证请求而发起建连。

e) CPE 必须允许来自任何源地址的拥有针对这个目标 CPE 的正确的认证参数的连接请求。

f) CPE 通过 HTTP 状态码“200（OK）”或“204（Not Content）”来响应一个经过正确认证的连接请求。CPE 必须在认证成功后并在开始发起这个会话之前立即发送这个响应。同时 HTTP 响应中的消息体的长度必须是 0。

g) 为了拒绝服务攻击，CPE 应该限制在某段时间内它能接受的连接请求数。如果由于这个原因 CPE 拒绝了一个连接请求，CPE 必须给这个连接请求响应一个 HTTP503 的状态码（无效的服务）。在这种情况下，CPE 不应该在响应报文中包括 HTTP 的 Retry-After 头。

h) 如果 CPE 对一个连接请求鉴权成功并跟上面的描述一样做了响应，并且它并不已经在会话过程中，则，它必须在发送响应后的 30 秒内根据 inform 里面的“6 CONNECTION REQUEST” EventCode 中包含的 ACS 地址来尝试建立一个会话。

   注：实际上在很小的几率上可能存在异常环境使得 CPE 不能满足这个需求。

i) 如果 ACS 收到了一个连接请求的成功响应，但是在 30 秒内 CPE 还无法根据 inform 里面的“6 CONNECTION REQUEST” EventCode 中包含的 ACS 地址成功建连，则 ACS 可以给 CPE 重新发送这个连接请求。

j) 如果 CPE 在成功鉴权并响应了一个连接请求后，且在跟这个 ACS 建立一个会话之前收到

了一个或多个其他的成功鉴权的连接请求，CPE 必须给每个连接请求返回一个成功的响应，但是必须不能为这些其他的连接请求建立其他的会话，不管在这段时间内收到了多少个连接请求。

k) 如果 CPE 已经跟 ACS 建立会话时收到一个或多个的连接请求，它必须不能过早地终止当前的会话，而必须选择一个下面的处理：

1) 用 HTTP503 状态响应码（无效的服务）来响应每个连接请求。在这种情况下，CPE 应给此响应包包括 HTTP Retry-After 头。

2) 在当前会话结束后，跟 Inform 的“6 CONNECTION REQUEST” EventCode 中包含地址正确地发起一个新的会话（不管在这之前收到了多少个连接请求）。在这种情况下，CPE 必须在现有会话结束并且所有修改已经有效的情况下立即发起这个会话。

此规程需要 CPE 保留在任何时候收到的连接请求，包括 CPE 处于会话时，也包括 CPE 在建立会话的过程中时。

l) CPE 必须不能因为由于上面描述的原因之外的其他原因来拒绝一个被完全鉴权过的连接请求。如果 CPE 由于某个上面描述的原因而拒绝这个连接请求，它必须不能给这个连接请求发起一个到 ACS 的会话。

此机制依赖于 ACS 已经至少一次地跟 CPE 通过 CPE 发起的交互中进行了通信。在这个交互中，如果 ACS 期望能够允许将来 ACS 发起的事务处理，它需要使用参数…ManagementServer.ConnectionRequestURL 的值（见[13]）。如果这个用于管理接入的 URL 改变了，CPE 必须通过发送一个 Inform 消息来通知 ACS 并指示新的管理 IP 地址，以此来保证 ACS 的更新。

端口 7547 已经被 IANA 分配给 CPE 广域网管理协议（见[17]），CPE 可以在连接请求 URL 中使用这个端口。

3.3 SSL/TLS 和 TCP 的使用

建议 CPE 广域网管理协议使用 SSL/TLS 进行传输，虽然此协议也可以直接在 TCP 连接上使用。如果不使用 SSL/TLS，就需要提供一些其他的安全方面的保证。特别地，SSL/TLS 提供了机密性和数据完整性，同时允许基于证书的认证代替共享秘密的认证。

使用 SSL/TLS 和 TCP 的某些限制定义如下：

a) CPE 必须支持 SSL3.0[10]或 TLS1.0[11]的一种或全部。

b) 如果 CPE 两种皆支持，它应该按照 RFC2246 的附录 E 中描述的把两种能力都通知给 ACS，由 ACS 来选择使用哪个协议。

c) 如果 ACS URL 已经被指示为一个 HTTPS URL，则 CPE 必须使用 SSL/TLS 跟 ACS 建立连接。

d) CPE 必须支持下面 SSL/TLS 密码套件：

1) RSA_WITH_3DES_EDE_CBC_SHA

2) RSA_WITH_RC4_128_SHA

e) CPE 必须能够发起到 ACS 的外向连接。

f) ACS 必须能够接收 CPE 发起的连接。

g) 如果使用了 SSL/TLS，CPE 必须使用 ACS 提供的证书来鉴权这个 ACS。鉴权 ACS 需要 CPE 必须根据根证书来验证这个证书，CPE 必须确保这个证书中的 subject 域的 CN(Common Name)组成部分的值跟 CPE 所清楚的 ACS URL 的 host 部分完全匹配（即使 ACS URL 中的 host 部分是一个 IP 地址）。此过程必须是直接对 CN 和 ACS URL 的 host 部分的字符串进行比较来实现。如果他们中只要有一方是 hostname 的格式（而不是 IP 地址），这个比较就必须不能包括 hostname 所对应的 IP 地址。

为了能够基于根证书进行验证，CPE 必须包含预先上载或者按照本规范之外的安全方式提供的一个或多个正确的根证书。

如果由于 HTTP 的重定向，CPE 需要用一个跟原先配置 ACS URL 不同的 URL 来跟 ACS 建立连接，CPE 必须使用重定向的 ACS URL 的 host 部分来进行 CN 部分的校验，而不是使用原先配置的 ACS URL。

CPE 应该等到有绝对正确的时间时才跟 ACS 连接。如果 CPE 选择了在没有绝对正确的时间（或它不支持绝对正确的时间）之前跟 ACS 进行连接，它必须忽略 ACS 证书中包含绝对时间的那一部分内容，例如，证书限制中的 not-valid-before 和 not-valid-after。

h) 支持使用客户端证书来进行 CPE 认证对于 CPE 和 ACS 都是可选的。这种客户端证书必须用适当的链来标记。当客户端证书被 ACS 用来认证 CPE 时，CPE 证书中的 CN(Common Name)域必须是下面两种类型中的一种：

1) 唯一的 CPE 客户端证书。在这种情况下，CN 域的值必须是在全范围内唯一地标识一个 CPE。特别地，CN 域必须符合按照 3.4.4 中建议的 username/userid 格式。

例如：

00D09E-0123456789

012345-STB-0123456789

01234-Set%2DTop%2DBox-0123456789

2) 通用 CPE 客户端证书。在这种情况下，CN 域的值可以对于一组 CPE 是一样的，比如所有供应商提供的特定 model 的所有 CPE。CN 域的内容在这种情况下并不是专用的。

如果使用了通用 CPE 客户端证书，ACS 应该另外用 HTTP 基本认证的或摘要（digest）认证来认证这个 CPE 以建立这个具体 CPE 的标识符。

3.4 HTTP 的使用

CPE 和 ACS 之间的 SOAP 消息承载于 HTTP1.1[5]之上，CPE 作为 HTTP 客户端而 ACS 作为 HTTP 服务器端。

注：CPE 广域网管理协议也用 HTTP 来实现连接请求，在那个使用，ACS 作为一个 HTTP 客户端而 CPE 作为一个 HTTP 服务器端。HTTP 的这个用法详见 3.2.2。

3.4.1 SOAP over HTTP 的打包

SOAP over HTTP 的打包是 HTTP 捆绑对 SOAP 协议的扩充（详见[8]中的第 6 章中描述），描述如下：

a) ACS 到 CPE 的 SOAP 请求通过 HTTP response 来发送，CPE 对这个 ACS 的请求所做出的 SOAP 响应通过紧接着的 HTTP POST 来传送。

b) 当在一个 HTTP 请求报文中存在一个 SOAP 响应包，或者当在 HTTP 请求报文中存在一个 SOAP 错误响应包，HTTP 请求报文的 SOAPAction 头必须不得包含任何值（没有应用）来表明这个头不能提供关于这个消息的任何信息。也就是说，它必须显示如下：

SOAPAction:

c) 如果一个 HTTP 请求报文或响应报文里面包含一个 SOAP 封装（envelope），这个 HTTP 的 content-type 头必须包含一个“text/xml”的类型/子类型。

d) 一个空的 HTTP POST 必须不得包含 SOAPAction 头。

e) 一个空的 HTTP POST 必须不得包含 content-type 头。

f) 一个包含某种 CPE 广域网管理协议承载（一个到 CPE 的 SOAP 请求，一个到 CPE 的成功的 SOAP 响应，或者一个包含了如 3.5 中描述的错误元素的 SOAP 错误响应包）的 HTTP 响应必须使用 HTTP 状态码 200（OK）。

下面是一个来自 ACS 的包含一个 SOAP 请求的 HTTP 响应的例子：

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=”utf-8”
Content-Length: xyz

<soap:Envelope
  xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:cwmp=”urn:dslforum-org:cwmp-1-0”>
    <soap:Body>
        <cwmp:Request>
            <argument>value</argument>
        </cwmp:Request>
    </soap:Body>
</soap:Envelope>
```

注：在上面的例子中，XML 命名空间前缀只是使用了一个例子。正真的命名前缀值是任意的值，它只是用于指示一个命名空间描述。

3.4.2 事务处理会话

为了把一系列有序的事务处理形成一个单独的会话，CPE 应该在整个会话过程中始终维护着一个 TCP 连接。然而，如果一个 TCP 链接在一个 HTTP 请求/响应循环后正常关闭了，并且如果此会话并没有在最后一个 HTTP 响应时终止（不管是成功的或者不成功的终止），CPE 必须在一个新的 TCP 连接上继续发送下一个 HTTP 请求。

在接收到一个认证 challenge，除非 ACS 通过 HTTP 头“Connection:close”特别请求关闭这个 TCP 连接，CPE 必须在同一个 TCP 连接中发送下一个 HTTP 请求（包含 HTTP 头“Authorization”）。否则，CPE 必须准循 ACS 请求关闭这个 TCP 连接，并在一个新的 TCP 连接上发生下一个 HTTP 请求（包含 HTTP 头“Authorization”）。

如果 CPE 由于某种原因建立 TCP 连接失败，发送 HTTP 消息失败，或者接收 HTTP 响应失败，CPE 必须认为这个会话已经被非正常关闭了。CPE 必须等待至少 30 秒后再宣告建立一个 TCP 链接失败，或者接收一个 HTTP 响应失败。

ACS 应该如[7]中描述的使用会话 cookie 来维护这个会话。ACS 也可以使用 old-style 的“Netscape”，或者用 new-style 的 cookie[7]。ACS 应该只有在这个 cookie 标注为 discard 时使用它，并且不应该假设 CPE 会维护一个超出会话期限的 cookie。

为了确保 ACS 使用一个会话 cookie，CPE 必须支持规范[7]一样使用 cookie，并在每个以后的 HTTP POST 中包含这个 cookie 的返回值，另外，CPE 没必要支持存储已经超过会话期限的 cookie。特别地，因为 ACS 可能发送一个 old-style，new-stpe，或是一个 old-style 和 new-style cookie 的混合，CPE 必须支持规范[7]中 9.1 的兼容性需求。CPE 必须支持 ACS 对应多个 cookie

的使用，必须至少需要 512 字节用于存储 cookie。

当一个会话成功地或非成功地终止后，CPE 必须关闭关联的跟 ACS 的 TCP 连接并且丢弃所有的 cookie 并标记为 discard。

CPE 必须支持使用 ACS 对 HTTP 的重定向。下面是 CPE 和 ACS 关于使用 HTTP 重定向的需求：

1) CPE 必须支持 HTTP 状态码 302（Found）和 307（Temporary Redirect）。

2) CPE 同时也可以支持用于重定向的 HTTP 状态码 301（Moved Permanently）。

3) CPE 必须允许在会话的任何时候发生重定向，ACS 可以在会话的任何时候发起重定向。

4) 如果发生了 CPE 重定向，它必须使用 HTTP 重定向响应中的 URL 来尝试继续这个会话。特别地，CPE 必须把重定向响应的 HTTP POST 重新发送到 ACS 的重定向后的 URL 的地址，并且 CPE 必须尝试继续这个会话，就好像这个重定向就根本没有发生。

5) 如果发生了 CPE 重定向，这个重定向的 URL 必须只是应用于当前会话的接下来部分或者直到本次会话中下一次重定向的发生。CPE 必须不能保存这个重定向的 URL（例如，作为…ManagementServer.URL，规范[13]），不能把它用于以后的会话或以后的针对本会话的 retry 会话。这个需求必须被执行，即使在重定向时使用了 HTTP 状态码 301（Moved Permanently）。

6) CPE 必须允许至多 5 个连续的重定向。如果 CPE 被多于 5 次连续的重定向，它可以认为此会话不正常地终止了。

7) HTTP 重定向中提供的 URL 可以是 HTTP 或 HTTPS URL。对应的传输层机制（TCP 或 SSL/TLS）必须使用这个新的目的地而不是重定向以前传输层使用的那个目的地。

8) 如果这个重定向会话中使用的是 SSL/TLS，要求 CPE 对这个 ACS 进行鉴权，这个鉴权必须基于重定向后的 URL 而不是原先配置的 ACS URL(参见 3.3)。

9) 在 ACS 发送的一个包含重定向状态码的 HTTP 响应中，HTTP 消息体的长度必须是 0。如果 CPE 收到了一个包含非空消息体的 HTTP 重定向响应消息，它必须忽略这个消息体内容。

10) 当发生了重定向后，CPE 必须把所有的相关的 cookie 包含在这个会话的接下来的 HTTP 请求中发给重定向后的 ACS。CPE 必须认为一个来至 ACS 的重定向是一个规范[7]中描述的“verifiable transaction”，因此它必须把 cookie 发送给重定向后的 ACS 而无需对这些 cookie 执行域验证。

3.4.3 文件传输

如果 CPE 被 ACS 通过下载或上载请求来要求执行一个文件传输，并且如果这个文件所在的位置是由一个跟 ACS 拥有相同 host name 的 HTTP URL，则 CPE 必须选择下面的一个途径来执行这个文件传输：

a) CPE 可以在已经建立好的连接上发送一个 HTTP GET/PUT。一旦开始传输文件，CPE 可以在继续维护这个连接的基础上发送另外的消息给 ACS。

b) CPE 可以打开另一个连接用于文件传输，而同时维护跟 ACS 的那个会话用于继续发送消息。

c) CPE 可以终止跟 ACS 的会话然后开始执行文件传输。

如果文件的位置不是一个 HTTP URL 或不跟 ACS 处于同一个 domain 或需要使用一个不同的端口，那么只有后面两种选项对它有效。

CPE 必须支持在 3.3 中描述的对 SSL/TLS 的使用来建立一个单独的 TCP 连接用于在 HTTP 上传输文件。如果文件位置使用由 HTTPS URL 来描述时，CPE 必须使用 SSL/TLS。

CPE必须在文件传输时支持HTTP基本（basic）和摘要（digest）认证两种认证。这个特定的认证由文件服务器根据virtue of providing a basic or digest authentication challenge来选择。如果文件服务器使用了认证，ACS必须用发起一个传输（例如下载，上载）的特定的RPC方法来指定信任书（specify credentials）。

3.4.4 认证

如果 CPE 没有经过 SSL/TLS 认证，ACS 必须使用 HTTP 对 CPE 进行认证。如果 SSL/TLS 已经被用于了加密认证，ACS 可以任意使用基本认证或摘要认证[6]。如果没有使用 SSL/TLS，ACS 必须使用摘要认证。

CPE 必须支持 HTTP 基本认证和摘要认证两种认证。由 ACS 通过 virtue of providing a basic or digest authentication challenge 来选择这个认证方案。

如果 CPE 已经从 ACS 收到了一个 authentication challenge（基本认证或者摘要认证），CPE 应该在这个 TCP 连接的所有以后发送的 HTTP 请求都带上一个认证头。无论 CPE 是否已经这样处理，ACS 可以在一个或多个 TCP 连接里面发起多个连续的 authentication challenges。

如果只要有一种形式的 HTTP 认证被用于认证 CPE，CPE 就应该使用一个在所有 CPE 厂商全范围内唯一的一个 username/userid。特别地，CPE 的 username/userid 应该是下面的某一种格式：

<OUI>”-“<ProductClass>”-“<SerialNumber>
<OUI>”-“<SerialNumber>

如果使用了上面的 username/userid 格式，这个<OUI>，<ProductClass>，和<SerialNumber>域必须跟 Inform 消息中的 DeviceIdStruct 里面的对应的参数完全匹配，另外，根据附录 A 中的

描述，为了保证可以从 username/userid 中取出这个参数的值，<ProductClass>和<SerialNumber>中的每一个字符不能是文字数字（alphanumeric：字母和数字混合）的也不能是下划线（“_”），而作为替换必须使用 RFC3986[12]中定义的 URI percent encoding。

如果 username/userid 使用了上面的格式，当且仅当 ProductClass 参数的值为空时必须使用第二种格式。

例如：

012345-0123456789
012345-STB-0123456789
012345-Set%2DTop%2DBox-0123456789

在任何一种 HTTP 认证中使用的密码应该对于每个 CPE 都是唯一的。换句话说，多个 CPE 不应该共享同一个密码。此密码是共享秘密，因此必须让 CPE 和 ACS 都要知道。在初始 CPE 安装时如何使得两个实体都知道这个共享密码的方法不属于本规范的介绍范围。CPE 和 ACS 都应该采取适当的步骤来防止未经认可的密码的接入，或者在一个 ACS 的情况下使用多个密码。

3.4.5 摘要（digest）认证

本章简单地描述了 CPE 广域网管理协议中使用摘要认证的需求。这个需求应用于 RPC 交互的连接和文件传输中。注意 ACS 和 CPE 在不同的连接类型中它们将会替换使用 HTTP 客户端和服务器端的角色。ACS 在发起连接请求时作为一个 HTTP 客户端。而 CPE 在发起到 ACS 的连接时作为一个 HTTP 客户端。

CPE 和 ACS 必须支持 RFC2617 在“qop”选项中包含值“auth”。依照 RFC2617，它表示如果 HTTP 服务器端给它提供了这个选项，HTTP 客户端必须使用一个 new style 摘要机制。

当使用摘要认证时，对于每个开启的新的 TCP 连接，ACS 应该用一个 new nonce value 而 CPE 应该用一个 new cnonce value。

注：如果 CPE 广域网管理协议的会话中不使用 SSL/TLS，ACS 对用于 HTTP 认证的 nonce values 的重新使用的政策可能对会话的机密性有很严重的影响。特别地，当 ACS 交叉地重新认证多个 TCP 连接时如果 ACS 重用了一个 nonce value，ACS 就会很容易收到 replay attacks。然而，如果会话中使用了 SSL/TLS，那么这个风险就会大大地减轻了。

CPE 和 ACS 必须支持 MD5 摘要运算法则（MD5 digest algorithm）。CPE 必须另外地支持 MD5-sess 摘要运算法则。

3.4.6 其他的 HTTP 需求

下面规定了一些其他的 HTTP 相关的需求：

a) 一旦 ACS 发送一个空 HTTP 响应，它必须使用 HTTP 状态码“204（无内容）”。

b) 一旦 CPE 发送一个空 HTTP 请求，HTTP 消息体的长度必须为 0。

c) CPE 必须不能向 HTTP1.1[5]中描述的那样使用流水线方式操作。

3.5 SOAP 的使用

CPE 广域网管理协议规定 SOAP1.1[8]为传输附录 A 中定义的 RPC 方法呼叫和响应的编码语法。

下面时 RPC 方法到 SOAP 编码的对应描述：

a) 编码必须采用标准的 SOAP 1.1envelope 和 serialization 命名空间：

   1) envelope 命名空间识别符“http://schemas.xmlsoap.org/soap/envelope/”

   2) serialization 命名空间识别符“http://schemas.xmlsoap.org/soap/encoding/”

b) 作为这个 CPE 广域网管理协议版本的一部分的所有的 element 和 attribute 都跟下面的命名空间识别符相关联：

   - “urn:dslforum-org:cwmp-1-0”

c) 附录 A 中使用的数据类型直接对应于 SOAP 1.1 serializaiton 命名空间中定义的数据类型。（通常来说，附录 A 中使用的类型是对应 SOAP 类型的限定子集。）

d) 根据 SOAP 规范[8]，指定为“anySimpleType”类型的 element 必须包括一个类型属性用于指示这个真正的 element 类型。

e) 除了“anySimpleType”之外其它类型的 element 可以在当且仅当被定义成附录 A 的 RPC 方式 XML schema（XML Schema 是用来描述和约束 XML 文档的一种 XML 语言，从功能上看，它和早期的 DTD 是非常类似的，但是它比 DTD 的更加强大）中的一个指定的数据类型时才包括一个类型属性。如果包括了一个类型属性，这个类型属性的值必须跟在 schema 中定义的数据类型完全匹配。

f) 对于一个数组变量（argument），这个在数组表项中指定的变量命名必须被用作整个数组元素（element）名字使用。一个数组中的成员（member element）命名必须是数组定义表项中规定的数组数据类型（除了括号和括弧中的长度限制之外），且必须不能是 namespace qualified。例如，一个叫做 ParameterList 的变量，是一个 ParameterValueStruct 结构的数组，应该 encoded 为：

```
<ParameterList soap-enc:arrayType=”cwmp:ParameterValueStruct[2]”>
    <ParameterValueStruct>
```

```
            <name>Parameter1</name>
            <value xsi:type=”sometype”>1234</value>
        </ParameterValueStruct>
        <ParameterValueStruct>
            <name>Parameter2</name>
            <value xsi:type=”someType”>5678</value>
        </ParameterValueStruct>
    </ParameterList>
```

注：argument 参数；
在函数调用中使用的数据项。一个参数可以是常量、变量或表达式

g) 另一个例子，GetRPCMethodsResponse 中的 MethodList 数组应该 encoded 如下：

```
<MethodList soap-enc:arrayType=”xsd:string[3]”>
    <string>GetRPCMethods</string>
    <string>Inform</string>
    <string>TransferComplete</string>
</MethodList>
```

注：在上面的例子中，XML 命名空间前缀的使用只是一个例子。真正的命名空间前缀值是任意的，只用于对应一个命名空间声明。

注：通常需要为 arrayType 属性规定一个 XML 命名空间前缀。对于 CWMP-specific 类型的数组通常是 CWMP 命名空间前缀，对于其它类型的数组通常是 XML Schema 命名空间前缀或者 SOAP encoding 命名空间前缀。

h) 参考 RPC 方式（附录 A 中定义的每种方式）编码的 SOAP 规范([8]中第七章)，方式调用中列出的每个变量代表一个[入]参数，同时方式响应中列出的每一个变量也代表一个[出]参数。这里将不使用[入/出]参数。

i) 使用标准的 SOAP 命名协定来定义 RPC 方式：通过在方式名中增加“Response”后缀来使得响应消息跟一个方式进行对应。

j) 一个 SOAP 封装必须正确地包含一个 body 元素。

k) CPE 必须能够接收一个至少 32KB(32768 字节)封装大小的 SOAP 请求而不生成一个“Resources Exceeded”响应。

l) CPE 必须能够生成一个任意需求的大小的 SOAP 响应而不产生一个“Resources Exceeded”响应，也就是说，CPE SOAP 响应长度没有最大值。

m) ACS 必须能够接收一个至少 32KB(32768 字节)封装大小的 SOAP 请求而不生成一个“Resources Exceeded”响应。

n) ACS 必须能够生成一个任意需求的大小的 SOAP 响应而不产生一个“Resources Exceeded”响应，也就是说，ACS SOAP 响应长度没有最大值。

o) 一个错误响应必须按照如下协定使用 SOAP 错误元素：

    1) SOAP 的错误码元素必须指出这个对应的特定错误的来源为客户端或服务器端。在这个用法中，客户端代表 SOAP 请求的发起端，而服务器端代表 SOAP 回应者。接收了错误响应后不需要使用这个元素的值，并可以完全地忽略整个 SOAP

错误码元素。

2) SOAP faultstring 子元素必须包含字符串“CWMP fault”。
3) SOAP detail 元素必须包含一个在“urm:dslforum-org:cwmp-1-0”命名空间定义的 fault 结构。附录 A 中的 RPC method XML schema 规范地定义了这个结构。这个结构包含下面的元素：
   一) 一个 FaultCode 元素，包含附录 A 中定义的单一数字的错误码。
   二) 一个 FaultString 元素，包含可阅读的错误描述。
   三) 一个 SetParameterValuesFault 元素，只用于对 SetParameterValues 方式的错误响应，它包含了一列一个或多个用于指示关于每个错误参数的错误细节的结构。这个结构包含了下面的元素：
      A) 一个 ParameterName 元素，包含错误参数的全路径名称。
      B) 一个 FaultCode 元素，包含附录 A 中定义的单一数字的错误码用于指示跟特定的错误参数相关的错误。
      C) 一个 FaultString 元素，包含一个针对特定错误参数的可阅读的错误描述。

   下面是一个包含错误响应的 envelop 例子：

```
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
        <soap:Header>
            <cwmp:ID soap:mustUnderstand="1">1234</cwmp:ID>
        </soap:Header>
    <soap:Body>
        <soap:Fault>
            <faultcode>Client</faultcode>
            <faultstring>CWMP fault</faultstring>
            <detail>
                <cwmp:Fault>
                    <FaultCode>9000</FaultCode>
                    <FaultString>Upload method not supported</FaultString>
                </cwmp:Fault>
            </detail>
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

   下面是一个包含一个对 SetParameterValues 方式呼叫的错误响应的 envelope 例子：

```
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
    <soap:Header>
        <cwmp:ID soap:mustUnderstand="1">1234</cwmp:ID>
    </soap:Header>
    <soap:Body>
```

```
            <soap:Fault>
                <faultcode>Client</faultcode>
                <faultstring>CWMP fault</faultstring>
                <detail>
                    <cwmp:Fault>
                        <FaultCode>9003</FaultCode>
                        <FaultString>Invalid arguments</FaultString>
                        <SetParameterValuesFault>
                            <ParameterName>
                                InternetGatewayDevice.Time.LocalTimeZone
                            </ParameterName>
                            <FaultCode>9012</FaultCode>
                            <FaultString>Not a valid time zone value</FaultString>
                        </SetParameterValuesFault>
                        <SetParameterValuesFault>
                            <ParameterName>
                                InternetGatewayDevice.Time.LocalTimeZoneName
                            </ParameterName>
                            <FaultCode>9012</FaultCode>
                            <FaultString>String too long</FaultString>
                        </SetParameterValuesFault>
                    </cwmp:Fault>
                </detail>
            </soap:Fault>
        </soap:Body>
    </soap:Envelope>
```

注：在上面的例子中，XML 的命名空间前缀只是一个例子。真正的命名空间前缀值是任意的，只用于对应一个命名空间声明。

一个错误响应只能用于对一个 SOAP 请求的响应。一个错误响应必须不能是针对一个 SOAP 响应或其他的错误响应的响应消息。

如果一个错误响应不能符合上面的需求，这个 SOAP 消息对于接收方必须要被认为无效。CPE 广域网管理协议会话上的无效 SOAP 的后果在 3.7 章中定义。

4) 当在处理一个收到的封装包（envelope）时，ACS 和 CPE 两者可以忽略：（a）所有不认识的 XML 元素和他们的子元素或内容--注 1，（b）所有不认识的 XML 属性和他们的值，（c）所有嵌入其中的 XML 注释，以及（d）所有 XML 处理指令。或者，ACS 和 CPE 对接收到的 XML 包进行完全验证，并且拒绝一个包含了不认识元素的封装包（envelope）。注意：这样会阻止（preclude）通过包含另外的元素而不修改消息名的方法来对已有消息的扩充。

注 1：除了在收到一个不认识的 SOAP action 时必须要返回一个错误响应来指示方式不被支持（见附录 A）之外，其它可以参考本协定。

5) 如果一个 RPC 方式要求参考 XML Schema 命名空间（for example for the “type” attribute, or for references to XML Schema data types），这些参考必须是针对这些命名空间定义的 2001 版本，特别指，http://www.w3.org/2001/XMLSchema-instance 和 http://www.w3.org/2001/XMLSchema。接收者可以拒绝一个参考跟他们命名空间不同的其它版本的 RPC 方式。

一个如上描述的 RPC 方式编码的例子，一个 GetParameterNames 请求应该编码如下：

```
<soap-env:Envelope xmlns:soap-enc="http://schemas.xmlsoap.org/soap/encoding/"
                   xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
  <soap-env:Header>
    <cwmp:ID soap-env:mustUnderstand="1">0</cwmp:ID>
  </soap-env:Header>
  <soap-env:Body>
    <cwmp:GetParameterNames>
      <ParameterPath>Object.</ParameterPath>
      <NextLevel>0</NextLevel>
    </cwmp:GetParameterNames>
  </soap-env:Body>
</soap-env:Envelope>
```

注：在上面的例子中，XML 的命名空间前缀只是一个例子。真正的命名空间前缀值是任意的，只用于对应一个命名空间声明。

注：这个 CWMP 命名空间前缀只指定给在 CWMP schema 的最高层次（at the top level of）的元素（上面例子中的 ID 和 GetParameterNames）。如果指定一个命名空间到这些元素之间元素上时错误的（上面例子中的 ParameterPath 和 NextLevel）。This is because the CWMP schema specifies an elementFormDefault value of “unqualified”。

CPE 广域网管理协议定义了一系列的 SOAP 头元素，详见表 4.

表 4——SOAP 头元素

Tag Name	Description
ID	这个头元素通过对每一个请求使用一个唯一识别符来用于关联 SOAP 请求和响应，对应的响应消息需要包含相匹配的标识符。识别符的值是一个任意的字符串，是由请求者任意设置的。 如果在 SOAP 请求中使用，这个 ID 头必须在对应的响应消息中出现（不管是一个成功的还是失败的响应）。

	因为对这个头元素的支持是被要求的，这个头的 mustUnderstand 属性必须被置为“1”（true）。
HoldRequests	这个的头元素可以包含在 ACS 发送到 CPE 的用于控制来自 CPE 请求的传输的封装包（envelope）里面。这个头元素必须不能出现在 CPE 发送到 ACS 的封装包（envelope）中。 它是一个布尔类型的值“0”（false）或“1”（true）。不存在则认为是等于“0”（false）。 CPE 收到这个头元素后的行为在 3.7.1.3 中定义。CPE 对于这个头元素的支持是被要求的。 因为对这个头元素的支持是被要求的，所以这个头元素的 mustUnderstand 属性必须设置为“1”（true）。

下面是一个所有头元素的使用的消息例子：

```
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
        <soap:Header>
            <cwmp:ID soap:mustUnderstand="1">1234</cwmp:ID>
            <cwmp:HoldRequests soap:mustUnderstand="1">0</cwmp:HoldRequests>
        </soap:Header>
        <soap:Body>
            <cwmp:Action>
                <argument>value</argument>
            </cwmp:Action>
        </soap:Body>
</soap:Envelope>
```

注：在上面的例子中，XML 的命名空间前缀只是一个例子。真正的命名空间前缀值是任意的，只用于对应一个命名空间声明。

3.6 RPC 支持需求

表 5 是对所有方式的一个概要，并且指示了附录 A 中每个 RPC 方式执行的环境是必须的还是可选的。

表 5——RPC 消息需求

方式名（Method name）	CPE 需求	ACS 需求
CPE methods	Responding	Calling
GetRPCMethods	必需的	可选的
SetParameterValues	必需的	必需的
GetParameterValues	必需的	必需的
GetParameterNames	必需的	必需的
SetParameterAttributes	必需的	可选的
GetParameterAttributes	必需的	可选的
AddObject	必需的	可选的
DeleteObject	必需的	可选的
Reboot	必需的	可选的
Download	必需的(注 5)	必需的（注 5）
Upload	可选的	可选的
FactoryReset	可选的	可选的
GetQueuedTransfers	可选的	可选的
ScheduleInform	可选的	可选的
SetVouchers	可选的（注 6）	可选的（注 6）
GetOptions	可选的（注 6）	可选的（注 6）
ACS methods	Calling	Responding
GetRPCMethods	可选的	必需的
Inform	必需的	必需的
TransferComplete	必需的（注 7）	必需的（注 8）

RequestDownload	可选的	可选的
Kicked	可选的	可选的

注 5：只有在支持任何类型的文件下载时是必须的
注 6：如果支持 voucher mechanism，SetVouchers 和 GetOptions 方式都是必须的。
注 7：只有在支持任何类型的文件下载或上载时是必须的。
注 8：只有在 ACS 支持发起文件上载或下载时是必须的。

3.7 事务处理会话规程

所有的事务处理会话必须由 CPE 的一个包含在初始 HTTP POST 中的 Inform 消息开始。This serves to initiate the set of transactions and communicate the limitations of the CPE with regard to message encoding。在一个会话中 Inform 消息必须不能出现多次（这个限制不应用于一个因为作为 HTTP 认证处理的一部分的 HTTP 状态码“401 未经授权的”的接收导致的 Inform 请求重传的可能需求，或者因为作为 HTTP 重定向的一部分的 HTTP 状态码 3xx 的接收导致的重传需求）。

这个会话在当 ACS 和 CPE 都已经没有请求需要发送且都已经没有需要等待接收的响应时终止。在这种情况下，CPE 必须关闭这个连接。

在同一时间，CPE 和它对应的 ACS 之间只能有一个会话存在。

> 注：事务处理会话只有在某个方向有消息需要传送时保持。当一个特定的信息交互完成后，这个对话和对应的 TCP 连接就没必要继续保持开启状态了。

3.7.1 CPE 操作

3.7.1.1 会话发起

CPE 会在 3.2.1 章节描述的情况下发起到 ACS 的一个事务处理会话。一旦这个跟 ACS 的连接成功建立，CPE 通过发生一个 Inform 请求到 ACS 来发起这个会话。它向 ACS 指示了 CPE 的当前状态同时表示 CPE 已经准备好接收来自 ACS 的请求。

但且仅当接收到了一个成功的 Inform 响应时，CPE 才能认为这个会话已经成功建立。

从会话发起到会话终止，CPE 必须确保所有 CPE 广域网管理协议所允许参数的事务处理的安全性。在一个会话期间，CPE 所有可配置的参数必须面向 ACS 作为只能通过 ACS 进行修改的一致性设置。在整个会话过程中，CPE 必须防护 ACS 对参数的更新操作被其它的实体所看到。它包括可配置参数的值，那些 presence 或 absence 的可配置参数和对象。如何达到事务处

理的安全性是 CPE 侧的问题。

CPE 必须采用一些必需的步骤来确保会话事务处理的安全性。例如，在某些异常的情况下，为了满足 CWMP 会话连接需求，可能需要 CPE 来终止一个 LAN 侧的管理会话。

3.7.1.2 接收请求

在一个会话中（会话已经成功发起，且在碰到如 3.7.1.4 描述的规定终止会话之前），收到一个来自 ACS 的 SOAP 请求，CPE 必须在下一个发送给 ACS 的 HTTP POST 中响应这个请求。

3.7.1.3 发送请求

在一个会话中（会话已经成功发起，且在碰到如 3.7.1.4 描述的规定终止会话之前），如果 CPE 需要发生一个或多个请求到 ACS， CPE 必须在当且仅当满足以下所有条件时于下一个 HTTP POST 中发送一个请求（不是多个）：

1) 最近从 ACS 收到的 HTTP 响应不包含 SOAP 请求。

2) ACS 已经指示 HoldRequests 为 false（参见 3.5）。此条件当且仅当在从 ACS 收到的最近的 HTTP 响应中包含下面情况时会满足：

   a) 一个 HoldRequests 头的值设置为 false 的 SOAP 封装包。

   b) 一个没有 HoldRequests 头的 SOAP 封装包。

   c) 没有 SOAP 封装的空 HTTP 响应。

3) 在当前会话已运行的这段时间，在 ACS 已经指示 HoldRequests 为 false（如上描述）后 CPE 没有发送一个空 HTTP POST。

   如果当碰到上面的规范时 CPE 已经 pengding 了一个或多个请求，选择发送哪个请求由 CPE 来判断决定，除非有其他特别的规定。

   在一个会话中，如果没有碰到上面的任何一个情况或者 CPE 没有需要发送给 ACS 的请求，并且来自 ACS 的最近的 HTTP 响应没有包含一个 SOAP 请求，CPE 必须发生一个空 HTTP POST。

   一旦在最近的 HoldRequest 为 false（见 3.5）且 CPE 已经发送了一个空 HTTP POST，CPE 必须不能在这个会话的接下来的时间里发送任何更多的请求。在这种情况下，如果 CPE 还有其他的请求需要发送到 ACS，CPE 必须等到接下来的会话中发送这些请求。

   表 6 概括了在会话过程中（会话已经成功发起，且在碰到如 3.7.1.4 描述的规定终止会话

之前）CPE 必须要发送给 ACS 的内容。

表 6——CPE 消息传输限制

	HoldRequests	有未完成的 ACS 请求	无未完成的 ACS 请求
CPE 请求 pending （注 9）	False	响应	请求
	True	响应	空 HTTP POST
无 CPE 请求 pending		响应	空 HTTP POST

注 9：只有在最近的 HoldRequests 为 false，且 CPE 还没有发送一个空 HTTP POST 时，CPE 才可以有请求 pending。否则，CPE 被认为是没有 pending 的请求。

3.7.1.4 会话终止

如果满足了下面所有的条件时，CPE 必须终止这个事务处理会话：

1) ACS 没有更多的要发送给 CPE 的请求。当且仅当来自 ACS 的最近的 HTTP 响应是空时 CPE 才可以做出这个推断。

2) CPE 没有更多的需要发送 ACS 的请求，且 CPE 已经向 ACS 发送了一个空 HTTP POST 并包括 HoldRequests 为 false（用于指示 ACS，表示 CPE 在这个会话中已经没有更多的需要发送的请求）。如表 6 所定义，如果这个条件并没有满足，但是 CPE 已经没有更多的请求或响应需要发送，它必须发送一个空 HTTP POST，这样就可以达到这个状态。

3) CPE 已经从 ACS 接收了所有的响应消息。

4) CPE 已经发送了所有的响应消息给 ACS 作为已经收到的请求的回应。

如果在一个本地规定的不小于 30 秒的周期内没有收到来自 ACS 的 HTTP 响应，CPE 必须认为一个会话被异常终止了。如果 CPE 接收 HTTP 响应出错，CPE 必须不能在同一个会话中尝试重传这个对应的 HTTP 请求。

如果 CPE 在响应一个 Inform 请求时收到了一个错误码不是“Retry request”（错误码 8005）的 SOAP 层错误，CPE 必须认为这个会话已经被异常终止了。

如果 CPE 从 ACS 收到一个 HTTP 响应来指示不合适（not well-formed）的 XML，认为无效的 SOAP 结构，包含一个 3.5 中没有描述的 SOAP 错误，或者 CPE 认为这个协议已经收到了侵犯，CPE 必须认为这个会话已经被异常终止了。

如果 CPE 从 ACS 收到一个带有不是由 CPE 来处理的错误状态码（一个 4xx 或 5xx 状态码）的 HTTP 响应时，CPE 必须认为这个会话已经异常终止了。注：CPE 应该接收一个带有“401 未经认可的”状态码的 HTTP 响应作为正常认证处理的一部分，当 CPE 随后尝试去进行重新认证时，如果返回的 HTTP 响应包含一个“401 未经认证的”状态码，CPE 必须认为这个会话已经被异常终止了。

如果没有碰到上面的情况，CPE 必须继续这个会话。

如果 CPE 在响应除了 Inform 之外的其他的方式时收到一个 SOAP 层的错误码不是“Retry request”（错误码 8005）的错误响应（见 3.5），CPE 必须继续这个会话。也就是说，这个类型的错误响应必须不能导致会话的不正常终止。

> 注：在一个错误的状态下，完全由 ACS 来判定这个错误响应是应该让这个会话继续的 SOAP 层错误，或者是应该导致一个会话异常终止的 HTTP 层错误。

如果在会话中执行了一个或多个需要 CPE 重启来完成这个请求操作的消息交互，CPE 必须等到这个会话按照上面的标准完全地终止了以后才开始执行这个复位操作。

如果会话被意外地终止了，CPE 必须按照 3.2.1.1 来尝试重建这个会话。在这种情况下，CPE 可以设置一个限制尝试重建会话的次数的本地限制。

3.7.1.5 事件

一个事件即一个指示，即已经发生的我们关心的某件事情需要 CPE 通过 3.3.1 中描述的 Inform 请求来通知 ACS。CPE 必须至少一次的尝试递交每个事件。如果 CPE 目前不跟 ACS 处于一个会话中，它必须立即尝试递交这个事件；否则，它必须等到这个会话终止后尝试递交他们。CPE 必须接收一个来自 ACS 的证实来确认一个事件的成功递交。一旦 CPE 成功地递交了一个事件，CPE 必须不能再一次发送同样的事件。另一方面，ACS 必须准备多次接收同一个事件，因为 ACS 可能已经发送了一个响应，但是 CPE 却一直收不到。某些类型的事件（例如 PERIODIC，VALUE CHANGE）可以合法地出现在随后的会话中即使已经在以前的会话中成功地递交。在这种情况下，同一会话中以后的相同的事件指示了相同类型的一个事件再一次发生而不是对一个事件递交失败的重传。

每一种类型的事件都有一个规范用于规定如果上次递交失败是否以及何时 CPE 必须重新尝试事件递交。当尝试重新递交事件，它必须在会话后立即执行；在一个会话中递交失败的事件不能被紧接着的会话忽略而在更后的会话中重新递交。

对于大部分的事件，递交在 CPE 收到一个成功的 InformResponse 时得到证实。三种标准的事件类型（KICKED,TRANSFER COMPLETE,REQUEST DOWNLOAD）指示一个或多个方式（Kicked[A.4.2.1],TransferComplete[A.3.3.2],RequestDownload[A.4.2.2]，分别地）会在这个会话的接下来的时间里调用，它是对这些指示事件递交的方式的一个成功响应。CPE 也可以发送一个 vendor-specific 事件（使用表 7 中的语法描述），它的成功递交，重试，以及丢弃的规定服从供应商（vendor）的定义。

如果在 CPE 需要重新递交一些事件时没有发生新的事件，CPE 必须按照 3.2.1.1 的会话重试规范中预先定义的那样去尝试重新递交这些事件。

下面是一张事件类型的表，他们在 Inform 请求里面的编码，他们的 cumulative behavior，CPE 必须要接收的用于确认成功递交的响应，以及在递交不成功时重试及/或丢弃的策略规定。

表 7——事件类型

Event Code	Cumulative Behavior	Explanation	ACS Response for Successful Delivery	Retry/Discard Policy
“0 BOOTSTRAP”	Single	指示此次会话由第一次 CPE 安装或 ACS URL 发生了变化而建立的。 必须导致 BOOTSTRAP EventCode 的特定条件： a) CPE 在出厂后第一次到 ACS 的连接 b) CPE 在恢复为出厂配置后第一次到 ACS 的连接 c) CPE 在 ACS URL 被修改后（通过任何方法）第一次到 ACS 的连接 跟其他所有的 EventCode 值一样，BOOTSTRAP EventCode 可能跟其他的 EventCode 值一起包含在事件数组（Event array）中。这是可能发生的，例如，CPE 出厂后首次 boot 时，CPE 会同时包含 BOOTSTRAP 和 BOOT EventCode。	InformResponse	CPE 必须不得丢弃且不递交 BOOTSTRAP 事件。 所有其他没有递交的事件在 BOOTSTRAP 后必须被丢弃。
“1 BOOT”	Single	指示此会话是由 CPE 上电起来或重启所建立的。它包括通过任何途径，包括 Reboot 方式的使用，导致的系统 boot 的发起。	InformResponse	CPE 必须尝试重新递交直到发生 reboot 才丢弃它
“2 PERIODIC”	Single	指示此次会话的建立是基于周期性的 Inform 间隔	InformResponse	CPE 必须不得丢弃和不递交 PERIODIC 事件
“3 SCHEDULED”	Single	指示此次会话的建立是由于 ScheduleInform 方式呼叫引起的。 这个事件码必须只跟”M ScheduleInform”事件码一起用。	InformResponse	CPE 必须不得丢弃和不递交 SCHEDULED 事件
“4 VALUE	Single	指示从最后一次成功的 Inform	InformResponse	CPE 必须重新尝

CHANGE”		（A.3.2.4 中定义的条件下），一个或多个带有使能的 Passive 或 Active Notification 的参数发生了修改（即使它的值已经被改回上次成功 Inform 的值）。 如果这个 EventCode 包含在了这个事件组（event array）中，所有这类修改的参数必须包含在这个 Inform 的参数列表（ParameterList）中。如果这个事件一旦被丢弃，整列被修改的参数必须同时被丢弃。		试递交直到 reboot 或者 ACS URL 发生了修改才丢弃它。
“5 KICKED”	Single	指示此会话是为了 web identity 管理（见附录 D）而建立的，并且在此会话过程中，kicked 方式（见 A.4.2.1）会被一次或多次的调用。	KickedResponse	CPE 可以根据自己的判断来尝试重新递交
“6 CONNECTION REQUEST”	Single	指示了此会话是因为一个来自 ACS 的连接请求（3.2）而建立的	InformResponse	CPE 必须不得尝试重新递交
“7 TRANSFER COMPLETE”	Single	指示此会话是为了指示一个早先请求的上载或下载的完成（不管是成功的或不成功的），TransferComplete 方式会在这个会话期间被调用一次或多次。这个事件码必须只跟“M Download”和/或“M Upload”事件码（“M Download”和“M Upload”见下面）。	TransferCompleteResponse	CPE 必须不得丢弃和不递交 TRANSFER COMPLETE 事件
“8 DIAGNOSTICS COMPLETE”	Single	在完成了一个或多个 ACS 发起的诊断测试后跟 ACS 重新建立连接时使用。	InformResponse	CPE 必须尝试递交该事件直到 reboot 才丢弃
“9 REQUEST DOWNLOAD”	Single	指示此会话的建立是为了让 CPE 一次或多次的调用 RequestDownload 方式（A.4.2.2）	RequestDownloadResponse	CPE 可以根据自己的判断来尝试重新递交
“M Reboot”	Multiple	ACS 通过使用 Reboot RPC 来让 CPE 发生重启。相当于一个可以生成一个“1 BOOT”事件码的事因。	InformResponse	CPE 必须不得丢弃和不递交“M Reboot”事件
“M ScheduleInform”	Multiple	ACS 请求一个预制 Inform	InformResponse	CPE 必须不得丢弃和不递交“M ScheduleInform”事件
“M Download”	Multiple	一个早先 ACS 通过 Download	TransferCompleteRes	CPE 必须不得丢

		方式（A.3.2.8）请求的下载内容完成。相当于“7 TRANSFER COMPLETE”。	ponse	弃和不递交“M Download”事件
“M Upload”	Multiple	一个早先 ACS 通过 Upload 方式（A.4.1.5）请求的上载内容完成。相当于“7 TRANSFER COMPLETE”。	TransferCompleteResponse	CPE 必须不得丢弃和不递交“M Upload”事件
“M” <供应商特有方式>	Not specified	供应商特有方式的请求操作的完成。The action taken by the CPE and response by the ACS is vendor-specific.一个供应商特有方式的名字必须是一个 A.3.1.1 章中描述的格式。 例如： “MX_012345_MyMethod”	Not specified	Not specified
“X “<OUI>” ” <事件>	Not specified	供应商特有事件。“X”后面的 OUI 以及空格是一个组织的唯一识别符，由大写字母的 6 位十六进制数字的值和头部的任意个 0 组成，这个值必须是一个有效的同[9]中定义的 OUI，并且必须是那个定义了这个供应商特有事件的组织。<事件>的值和解析是供应商特有的。 “X 012345 MyEvent”	Not specified	Not specified

上表中的累积行为把事件类型区分为非累积（“Single”）和累积（“Multiple”）两种。例如，如果 CPE 在早先的“1 BOOT”事件还没有递交时重启了，没有必要在下一个 Inform 中包含两个“1 BOOT”事件组实体。相反的，如果在早先的“M Download”还没有完成递交的情况下一个下载完成了，在下一个 Inform 里面就需要包含两个“M Download”事件组实体，因为每一个都关联了一个不同的 ACS 请求。“Single”和“Multiple”累积行为定义如下：

1) 如果发生了一个“Single”累积行为的事件，这一列事件在下一个 Inform 中必须只能包含一个此事件码的实例，不过是否由同一类型的没有递交的事件存在。

2) 如果发生了一个“Multiple”累积行为的事件，这个新的事件码必须包含在事件列表中，不受同一类型的任何没有递交的事件的影响，它也必须不能影响那些没有递交的事件。

当一个或多个事件直接跟同一个根源相关联，那么所有这些事件必须包含在一个事件列表中。下面是这种情况下的例子（没有列出全部）：

1) 因为 Reboot RPC 方式导致的重启。在这种情况下，Inform 里面必须至少包含下列事件码的值：

“1 BOOT”

“M Reboot”

2) 由于一个早先的下载请求引起的 TransferComplete 在一个新会话中的发送，跟重启无关的成功的传输：

“7 TRANSFER COMPLETE”
“M Download”

3) 一个或多个设置了 Passive Notification 的参数值已经因为最近的 Inform 发生了变化，一个周期 Inform 产生（在这种情况下，因为对于 Passive Notifications，这些事件必须包含在同一个 Inform 中，会导致“4 VALUE CHANGE”事件发生的 Inform 不得不由一些其他的原因来产生——在这个例子中，是一个周期 inform）：

“2 PERIODIC”
“4 VALUE CHANGE”

对于那些不同原因的事件，如果他们同时发生，CPE 应该在同一个 Inform 消息中包括所有的这些事件，也可以为每一个这种事件发送单独的 Inform 消息。一个不相关事件的例子：

“2 PERIODIC”
“7 TRANSFER COMPLETE”

3.7.1.6 方式重试行为准则

如果在对一个来自 CPE 的请求的响应中，CPE 从 ACS 收到了一个“重试请求”响应（错误码 8005），CPE 必须在当前会话的下一个 HTTP POST 中重新发送同一个请求。这个行为准则应用于所有的 ACS 方式（包括 Inform）。

如果改为 CPE 收到一个用于响应非 Inform 方式的任何其他方式的，且错误码不是 8005 的而是任何其他错误码的响应，CPE 必须继续本次会话，且必须不得尝试重发这个方式（如 3.7.1.4 中描述，在 Inform 情况下一个这样的响应会导致会话的终止）。

3.7.2 ACS 操作

3.7.2.1 会话发起

当收到了来自 CPE 的初始的 Inform 请求，如果 ACS 期望允许这个会话的发起，它必须用一个 Inform 响应来回应。

3.7.2.2 接受请求

在一个会话中（此会话已经成功发起，且在碰到 3.7.2.4 描述的准则而终止会话之前），接收到一个来自 CPE 的 SOAP 请求，ACS 必须在下一个发送给 CPE 的 HTTP 响应中来回应这个请求。

如果 ACS 期望在这个会话的某段时间内阻止 CPE 发送请求消息，它可以通过在每个传送给 CPE 的封装包中设置 HoldRequests 头的值为 true 来实现，直到 ACS 又期望允许来自 CPE 的请求为止。ACS 必须允许 CPE 的请求直到此会话完成（这一点可以通过 HoldRequestes 头来明确的实现或通过发送一个空 HTTP 响应来含蓄地实现）。

3.7.2.3 发送请求

在一个会话中（此会话已经成功发起，且在碰到 3.7.2.4 描述的准则而终止会话之前），如果 ACS 有一个或多个需要发送给 CPE 的请求，且最近收到的来自 CPE 的 HTTP POST 里面没有包含 SOAP 请求，ACS 必须在下一个 HTTP 响应中发送这些请求中的一个请求。

另外，在一个会话中，如果 ACS 没有需要发送给 CPE 的请求，且最近收到的来自 CPE 的 HTTP POST 里面没有包含 SOAP 请求，ACS 必须发送一个空 HTTP 响应。

表 8 概述了在一个会话期间 ACS 必须要发送什么给 CPE（此会话已经成功发起，且在碰到 3.7.2.4 描述的准则而终止会话之前）。

表 8——ACS 消息传输限制

	CPE request outstanding	No CPE request outstanding
有 pending 的 ACS 请求	响应	请求
无 pending 的 ACS 请求	响应	空 HTTP 响应

3.7.2.4 会话终止

CPE 是对 ACS 的 HTTP 连接的操纵方，由 CPE 来负责连接的发起和终止。

ACS 必须当所有下面的条件都成立时才认为此会话终止了：

1) CPE 已经没有更多的请求要发送给 ACS。ACS 当且仅当在 HoldRequests 为 false 时收到了一个来自 CPE 的空 HTTP POST 才能认为满足此条件。

2) ACS 没有更多的请求需要发送给 CPE，并且最近的 ACS 发送给 CPE 的 HTTP 响应是空的（用于指示 CPE，ACS 已经没有更多的请求了）。

3) ACS 已经针对早前收到的所有请求都回了响应消息给 CPE。

4) ACS 已经从 CPE 收到了所有自己发出的请求的响应消息。

如果在 ACS 发送最后一个 HTTP 响应之前所有以上的准则都已经满足，ACS 的最后一个 HTTP 响应必须是空的。

如果以上的准则并没有全部的满足，当 ACS 已经在一个本地定义的不少于 30 秒的时间段内没有收到对应 CPE 的一个 HTTP POST，它可以认为此会话终止了。在这种情况下，ACS 可以通过执行一个连接请求来重建一个会话（见 3.2.2）。

如果 ACS 收到一个来自 CPE 的 HTTP POST，且对应的 XML 不是 well-formed，它的 SOAP 结构被认为是无效的，或它包含了一个不在 3.5 中描述的 SOAP 错误的格式，ACS 必须用一个 HTTP400 状态码（Bad Request）来响应 CPE，并且必须认为此会话已经异常终止了。这个错误响应必须不能包含一个 SOAP 内容，但可以包含可理解的文本内容用于更详细的描述这个错误的原因。

如果 ACS 从一个它认为已经超时的会话中收到一个请求，或 ACS 认为发生了一些其他协议的攻击，或由于 ACS 所判定的一些其他原因，ACS 可以通过用 HTTP400 状态码（Bad Request）响应 CPE 来异常终止这个会话。这个 HTTP 响应必须不能包含任何的 SOAP 内容，但可以包含可理解的文本内容用于更详细的描述这个错误的原因。

如果 ACS 从 CPE 收到了一个 SOAP 错误相应，如 3.5 中定义，ACS 可以选择下面的一个操作：

1) ACS 可以强制地异常终止这个会话。为了实现它，ACS 必须用 HTTP400 状态码（Bad Request）响应 CPE。这个 HTTP 响应必须不能包含任何的 SOAP 内容，但可以包含可理解的文本内容用于更详细的描述这个错误的原因。

2) ACS 可以尝试去正常地终止这个会话，这样 CPE 就不会来尝试重建这个会话。为了实现它，ACS 不能发送更多的请求到 CPE，且应该遵循上面定义的规则去判定此会话什么时候终止。

3) ACS 可以继续这个会话，并发送其他的请求到 CPE。

3.7.3 事务处理的例子

图 3 所示的例子中，ACS 首先读出一些参数的值，然后基于这个结果，设置一些参数的值。

图 3——事务处理会话例子

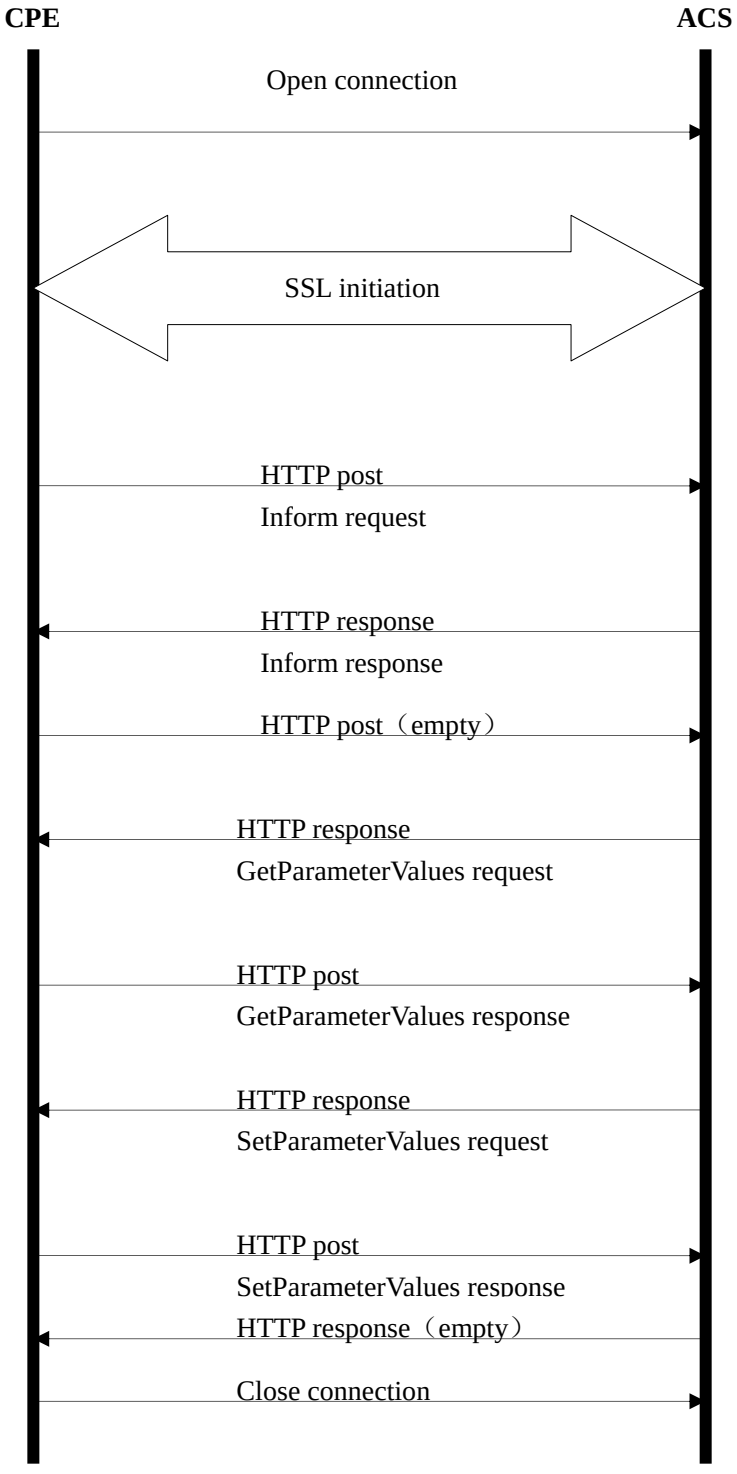


图 4 中显示的例子，ACS 首先发起一个文件下载，然后 CPE 在同一个会话中发送一个 TransferComlete。注：此情况只有在这个文件下载非常的短，且 CPE 有能力在这个 CPE 广域网管理协议会话正在进行时并列执行它（实际上 CPE 不需要如此操作）时才会发生。为了允许这个可能性，ACS 设置 HoldRequests 等于 true 直到它已经成功的把这些请求发送给 CPE。

图 4——ACS：HoldRequests 等于 true