

ICS 27.010
F 01

STCE

节 能 减 排 联 盟 标 准

STCE 1008—2013

基于泛在网的建筑节能监测控制 共性技术方案应用指南

Application Guide for Common Technology Solutions of Monitor Control on
Building Energy Conservation based on Ubiquitous Network

2013-11-15 发布

2014-1-1 实施

全国节能减排标准化技术联盟

发布

前 言

本标准按照GB/T 1.1—2009给出的规则起草。

本标准由全国节能减排标准化技术联盟提出并归口。

本标准起草单位：北京天地互连信息技术有限公司、中国电信股份有限公司北京研究院、中国标准化研究院、中关村国标节能低碳技术研究院。

本标准主要起草人：江连山、谷晨、宋阳、李文杰、刘丽、谭秀颖、潘崇超、龚向敏、吕秋生、王磊。

本标准为首次发布。

基于泛在网的建筑节能监测控制共性技术方案应用指南

1 范围

本标准基于泛在网技术，规定了建筑节能监测控制平台的系统架构。
本标准适用于建筑节能监测控制系统的设计、运营和实施部署。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

IEEE Std 1888 TM	Ubiquitous Green Community Control Network Protocol
YDB 062	泛在网术语
YDB 063	通信网支撑泛在物联应用绿色社区总体业务能力要求

3 术语和定义

3.1

泛在网 ubiquitous network

基于个人和社会的需求，实现人与人、人与物、物与物之间按需进行的信息获取、信息传递、信息存储、信息处理，具有环境感知、内容感知能力和智能性，为个人和社会提供泛在的、无所不备的信息服务和应用的网络。

3.2

智能联动 intelligent linkage

用电设施之间，根据联动机制，进行智能化自操作的一种节电业务。

3.3

远程测控 remote measurement and control

通过通信网络实现对用电设施远程操作的一种节电业务，包括获取数据、发送指令等。

4 建筑节能监测控制系统架构

基于泛在网的建筑节能监测控制系统如图1所示，系统包括现场设施域、网络传输域、分析处理域与应用域四大部分。

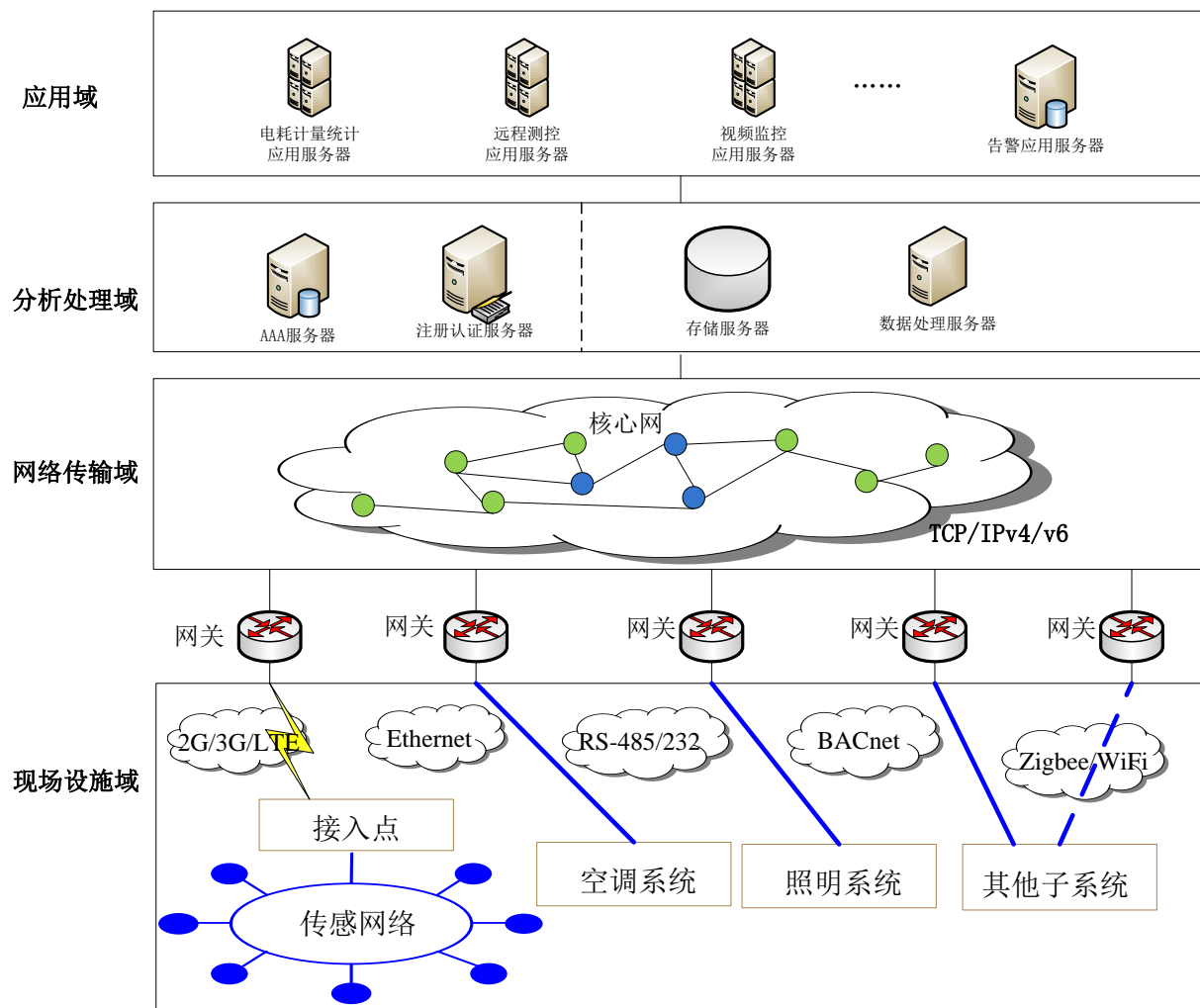


图1 基于泛在网的建筑节能监测控制系统架构图

5 现场设施域

5.1 网络架构

现场设施域主要由建筑内的用电设施（如空调系统、照明系统、暖通系统、电梯等其他子系统）和传感网络组成，传感网络包括环境监测感知节点，安防节点和智能控制节点。

5.2 用电设施的现场部署

各种不同的用电设施需要配备 I/O 通信接口，通过任意类型的协议实现通信和控制，通过相应的网关完成协议转换，统一接入到建筑节能监测控制系统中。用电设施支持的通信协议包括标准的 BACNet、RS485、RS232、Zigbee、Lontalk、Modbus、KNX 等，也可以包括其他未开放的私有协议。

5.2.1 网关通信接口要求

网关的北向通信接口要求包括但不限于：

- a) 可选的支持Ethernet接口，符合IEEE 802.3标准；
- b) 可选的支持光纤接口；
- c) 可选的支持2G、3G、LTE接口；
- d) 可选的支持WLAN接口，符合IEEE 802.11b、802.11g或802.11n协议，推荐使用IEEE 802.11n。

网关的南向通信接口与用电设施的I/O接口所支持的通信接口匹配，包括但不限于Ethernet接口，RS232/RS485，RF433，PLC，RF485，Zigbee，WiFi，6LowPAN，红外。

5.2.2 网关控制接口要求

网关的北向控制接口要求包括但不限于：

- a) 支持附录A.2.1.1中定义的信息读取协议；
- b) 支持附录A.2.1.1中定义的信息发送协议；
- c) 支持附录A.2.1.1中定义的信息触发协议；
- d) 支持附录A.2.1.2中定义的注册协议；

e) 支持附录A.2.1.2中定义的查找协议。

网关的南向控制接口与用电设施的I/O接口所支持的控制协议匹配，包括但不限于BACNet，RS232C，Zigbee，LonTalk，Modbus，KNX，也支持用电设施厂商所定义的私有控制协议。

5.2.3 用电设施通信接口要求

用电设施通信接口与网关南向通信接口相匹配，包括但不限于Ethernet接口，RS232/RS485，RF433，PLC，RF485，Zigbee，WiFi，6LowPAN，红外。

5.2.4 用电设施控制接口要求

用电设施控制接口与网关南向控制接口相匹配，包括但不限于BACNet，RS232C，Zigbee，LonTalk，Modbus，KNX，也可支持厂商自行设计的私有协议。

6 网络传输域

在网络传输域，以TCP/IP协议为基础，同时支持IPv4与IPv6协议。核心网子层包括现有的电信网、互联网、广电网，以及未来下一代网络的各个行业自建的专用网络。

7 分析处理域

7.1 注册认证服务器

注册认证服务器负责管理现场设备、网关设备的注册信息，支持附录A.2.2.4中规定的标准接口及协议，实现的功能包括但不限于：

- a) 创建并维护建筑节能监测控制系统各组成部分的绑定关系；
- b) 提供绑定关系的查询功能。

7.2 AAA服务器

AAA服务器的功能为实现安全服务，支持鉴权、授权及计费，实现数据传输的机密性以及完整性。

7.3 存储服务器

存储服务器支持统一的标准接口及协议，实现的功能包括但不限于：

- a) 存储来自现场设施系统的原始数据；
- b) 存储来自应用域处理过的数据结果；
- c) 提供对存储数据的查询功能；
- d) 在数据交互过程中，支持附录A.2.1.1中定义的信息读取协议及信息发送协议。

7.4 数据处理服务器

数据处理服务器支持统一的标准接口及协议，实现的功能包括但不限于：

- a) 对来自现场设施系统及应用域的数据进行缓存；
- b) 数据处理功能，包括预处理、过滤、分析、整合、变换、评估等；
- c) 在数据交互过程中，支持附录A.2.1.1中定义的信息读取协议及信息发送协议。

8 应用域

建筑节能监测控制系统架构的应用域为管理人员和用户提供针对性的应用服务，包括但不限于电耗计量统计、远程测控应用、视频监控应用及告警应用服务。

应用域中的服务器遵循附录A.1.1中定义的应用单元规范，支持附录A.2中定义的标准接口及协议，实现的功能包括但不限于：

- a) 节电业务生成功能，包括电耗计量统计、远程测控、视频监控、告警应用等；
- b) 支持附录A.2.1.1中定义的信息读取协议及信息发送协议。

附录 A
(规范性附录)

IEEE 1888 标准 - 泛在绿色社区控制网络协议

A.1 泛在绿色社区控制网络体系架构

A.1.1 组网架构

泛在绿色社区控制网络采用标准化的通信接口、数据格式以及通信协议，构成了开放的、标准的网络系统，其组网架构如图A.1所示。

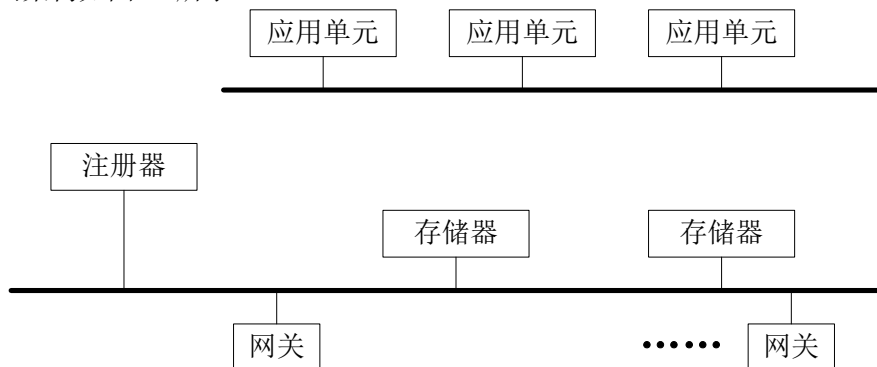


图 A.1 泛在绿色社区控制网络架构图

如图A.1所示，网关、存储器、应用单元和注册器组成了泛在绿色社区控制网络。泛在绿色社区控制网络协议规定的技术规范适用于基于TCP / IP协议的网络架构。

泛在绿色社区控制网络协议的主要目标之一是实现网络实体之间的互操作性。因此，网关，存储器和应用单元，统一抽象为“组件”，具有相同的通信接口。注册器与这些组件有不同通信接口。组件属于数据平面，注册器则属于控制平面。在泛在绿色社区控制网络中，注册器是组件的代理，管理元信息，例如：组件的具体实现形式以及相应管控对象的点标识等，从而实现组件之间的自主绑定和管理。泛在绿色社区控制网络协议详细描述了组件之间如何相互协作，以及组件与注册器之间如何绑定相应信息。

网关组件向下连接传感器、执行器等组成的现场网络，通过统一的抽象，封装不同现场网络的数据模型和访问方法。网关根据其他组件写入的值对执行器进行相应操作，并向其他组件提供传感器读取的数值。

存储器组件存储历史数据。其他组件写入的数据被永久保存在后端磁盘。当其他组件请求获取数据时，存储器还负责提供相应的数据值。

应用单元组件针对传感器读取的数据和对执行器的指令提供相关的应用。应用单元可以提供用户接口以显示当前环境状况，支持用户输入对执行器的设定策略，还可以提供各种类型的应用业务。

注册器作为组件的代理，主要负责实现组件之间的自动绑定。注册器不属于数据平面，不会对传感器和执行器进行直接操作。

A.1.2 系统模型和部署

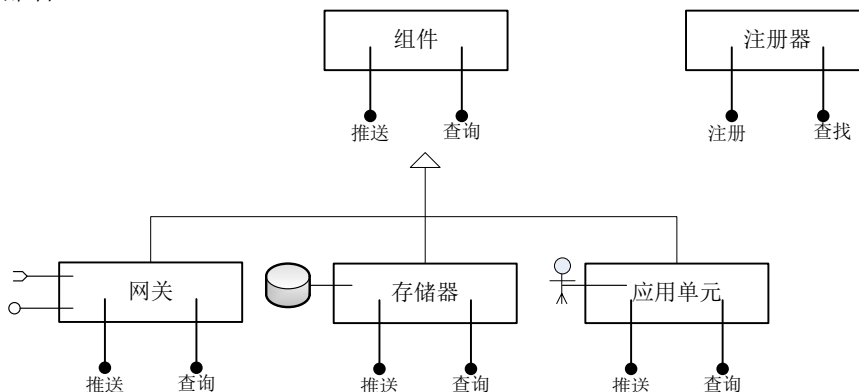


图 A.2 社区智能节电控制网络的系统模型

图A.2描述了泛在绿色社区控制网络的系统模型。作为组成系统的基本单元，组件是对网关、存储器、应用单元的统一抽象，其接口提供两种方法：推送和查询。由于网关、存储器和应用单元都是组件的继承类，因此，它们具有相同的接口，提供同样的推送和查询方法，并使用相同的协议通信。

[1]. 查询：一种从组件中获取数据（包括基于事件驱动的数据传输）的方法；

[2]. 推送：一种向组件中写入数据的方法。

注册器则与组件具有不同的接口，提供注册和查找方法。

[1]. 注册：一种绑定组件以及相应管控对象的方法；

[2]. 查找：一种搜索特定组件或管控对象的方法。

在网关、存储器、应用单元和注册器的实际部署实现中：

[1]. 网关通过推送和查询方法实现对现场网络数据的封装，提供对现场网络的输入/输出访问；

[2]. 存储器将通过推送方法对获取的数据进行保存，并通过查询方法实现历史数据的查询；

[3]. 应用单元实现其他功能。例如，提供用户门户，实现数据处理等；

[4]. 注册器管理组件的具体实现形式以及相应管控对象的点标识，通过注册方法提供组件和管控对象之间的绑定，通过查找方法提供针对组件和管控对象的查询。

A.1.3 管控对象及点标识

管控对象是社区智能节电系统管理和控制的对象实体，由基于统一资源标识符的点标识作为全局唯一标识符进行标识。管控对象描述了用于在组件之间传输某特定数据序列的消息通道，可对应于一系列的传感器读数，执行器命令等。在泛在绿色社区控制网络中，管控对象用“值”来描述，“值”可以是任何的对象类型。由于管控对象属于不同的类型，具备多种多样的特性，因此除了规定点标识来进行全局唯一标识外，还采用属性文件来描述管控对象的具体属性。

通过调用其他组件的接入接口上的方法(查询和推送)，可以在组件之间传递管控对象的“值”。通过使用这些方法，一个组件可以从另一个组件得到指定管控对象的“值”，同时也可以将某管控对象相关的“值”传输到另一个组件。

一个管控对象对应于一个可全局唯一标识的数据序列。数据可能是来自某特定传感器的读取数据，或是发往某特定执行器的指令数据。因此，为了在全局范围内区分数据序列，每个管控对象都应该有一个全局唯一的标识符。对于局部范围的私有应用，无需保证全局唯一性（不推荐使用该方式）。通过使用统一资源标识符作为点标识，可以实现管控对象的全局访问。点标识的一个范例是X（=<http://gw.foo.org/sensor1>），标识了某网关管理下的一个传感器节点。

在泛在绿色社区控制网络中，支持对管控对象的分级管理，从而引入了管控对象集的概念。一个管控对象集聚集多个管控对象和多个子管控对象集，采用树状结构作为管控对象的数据模型。通过该方法允许将管控对象分组，并进行层次化的常规操作，如A.3所示。

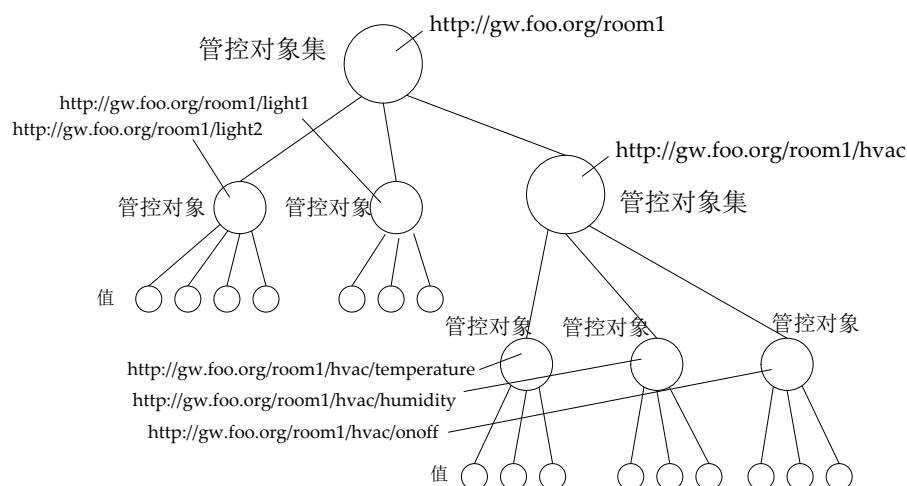


图 A.3 管控对象的树状数据模型

图A.3中，管控对象集聚集了多个管控对象和管控对象集，这种结构实现了分级管理。每个管控对象通过一系列的“值”元素来描述，传感器的一个读数或执行器的一条指令都是一个“值”元素。“值”元素本身不需要全局唯一的标识符，但具有时间属性，从而标识该“值”何时从传感器提取，或者应该在何时将该“值”发往执行器上。

在泛在绿色社区控制网络中，仅对管控对象和管控对象集定义了id属性，对“值”定义了时间属性。其他的属性可以根据具体的节点应用和现场网络进行灵活的定义和扩展。

A.2 泛在绿色社区控制网络协议架构

A.2.1 通信协议

A.2.1.1 组件之间的通信协议

在泛在绿色社区控制网络中，组件包括了网关、存储器、应用单元三类。组件之间支持三种类型的通信协议，分别是信息读取协议、信息发送协议和信息触发协议。

信息读取协议用于从远程组件获取数据。信息发送协议用于向远程组件传输数据。信息触发协议用于基于事件查询的注册和基于事件的数据传输。

下面针对这三种协议进行详细定义和描述。

[1]. 信息读取协议

信息读取协议用于从远程组件获取数据。请求获取数据的组件为“信息请求方”，负责应答提供数据的组件为“信息提供方”。图A.4是信息获取协议的示意图。

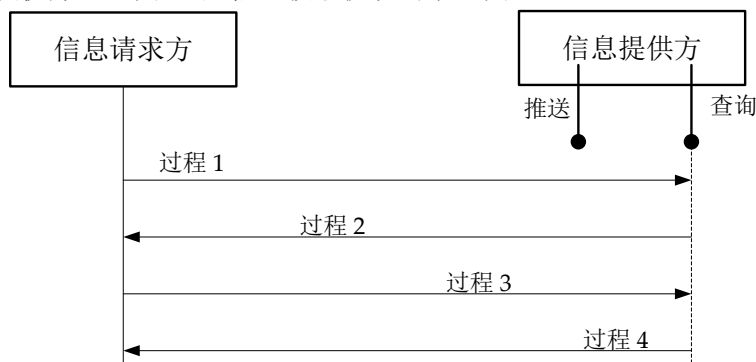


图 A.4 信息读取协议示意图

如图A.4所示，信息请求方向信息提供方发起请求，信息提供方返回响应。

过程 1：信息请求方调用信息提供方的查询方法。信息请求方发送查询请求，携带了查询条件，例如限定数据集范围的相关信息等等。除此之外，查询请求中还需要指明在远程过程调用（RPC）响应时可接受的最大数据集大小，即可同时接收管控对象“值”的最多数目，缺省值为 100。

过程 2：信息提供方通过 RPC 响应返回数据集。如果返回数据集的大小超过了信息请求方最大可接受的数据的大小，或是需要消耗信息提供方过多的计算资源，则信息提供方仅返回整个数据集的一个子集，并提供一个指针用于后续的数据获取。

过程 3：如果信息请求方接收到的响应中有一个指针，则表明还有后续的数据需要传输，此时信息请求方再次调用信息提供方的查询方法，用于获取剩余的数据，信息提供方重复过程 2 过程。

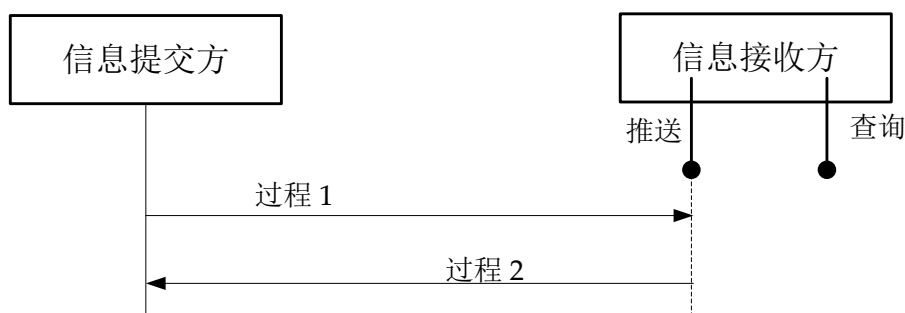
过程 4：如果接收到的响应中没有指针，表示所有的数据已经传递完毕，信息读取协议的过程也完成了。

需要注意的是，指针应具有有效时间。如果信息提供方接收到的请求消息中指针已经过期失效，则返回报错信息。指针有效期缺省推荐值为60秒。但是如果网络带宽不够，导致RPC耗时可能变长，则需要适当增加指针的有效期时间。

除了指针失效外，信息提供方遇到任何错误时都会主动返回错误信息，如访问控制策略错误，或其他系统错误等。

[2]. 信息发送协议

信息发送协议是用于向远程组件传输数据。提交数据的组件为“信息提交方”，接收数据的组件为“信息接收方”。信息发送协议示意图如A.5所示。



A.5 信息发送协议示意图

如图 A.5 所示，信息提交方向信息接收方发起数据传输请求，信息接收方返回响应。

过程 1：信息提交方调用信息接收方的推送方法，携带待发送数据的内容。

过程 2：信息接收方向信息提交方返回结果，是成功或失败。

[3]. 信息触发协议

信息触发协议支持基于事件的查询注册和数据传输。信息触发协议涉及的组件包括了“信息请求方”、“信息提供方”、“数据信息回送方”、“控制信息回送方”四种。信息请求方是用于向信息提供方设置基于事件的查询条件的组件；信息提供方负责在收到与触发查询条件匹配的更新时，发送数据；数据信息回送方是用于从信息提供方接收数据的组件；控制信息回送方则是从信息提供方接收控制信息的组件。

图A.6是信息触发协议的示意图。在实际的应用实现中，“数据信息回送方”、“控制信息回送方”和信息请求方通常是同一个组件，此时的信息触发协议实际部署方式如图A.7所示。

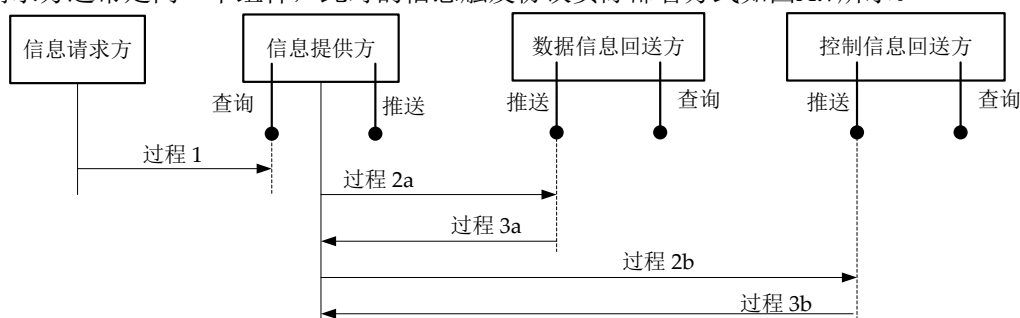


图 A.6 信息触发协议示意图

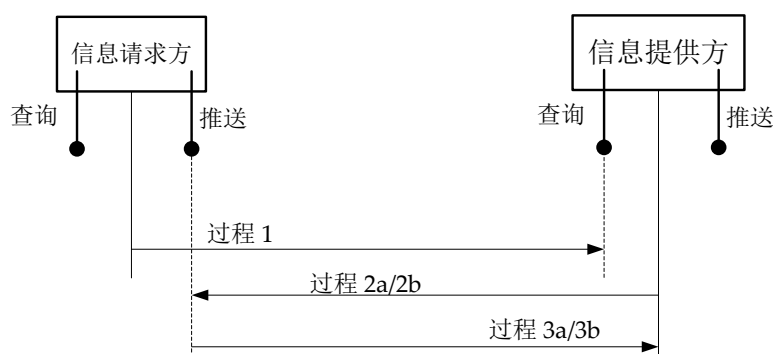


图 A.7 信息触发协议实际部署实现图

过程 1：信息请求方调用信息提供方的查询方法。除了查询表达式之外，同时传输的参数还包括查询的有效时间 TTL，数据信息回送方组件的 URI 和控制信息回送方组件的 URI。

在该过程中，需要周期性的触发该过程，以更新信息提供方记录的 TTL，从而使信息提供方能够继续接收更新通知。为了使信息提供方可以区分接收到的是更新请求还是新的设置请求，需要为更新的触发请求在查询中设置相同的 ID 以标识是同一个信息触发消息。

如果信息提供方无法接受该查询，则返回错误消息。

过程 2a：信息提供方调用数据信息回送方组件的推送方法，推送与查询表达式匹配的更新数据。

过程 3a：数据信息回送方组件回送成功或错误信息。

过程 2b: 信息提供方调用控制信息回送方的推送方法, 推送与通信错误相关的控制信息 (例如过程 3a 中遇到的错误)。

过程 3b: 控制信息回送方回送成功或错误信息。

在信息触发协议所涉及的过程中, 每过1秒, 信息提供方就将查询表达式的TTL减1。如果TTL达到“0”, 信息提供方拒绝该查询, 并将其从查询表项中删除。如果信息请求方想显式的删除某查询请求, 也可以通过采用指定TTL=0的方法实现。

A.2.1.2 组件与注册器之间的通信协议

在泛在绿色社区智控制网络中, 注册器的接口提供两种方法: 注册和查找。组件与注册器之间支持两种类型的通信协议, 分别是注册协议和查找协议。其中注册协议调用注册方法, 查找协议调用查找方法。

注册协议用于组件向注册器的注册, 其注册内容包括了组件的具体实现形式以及相应管控对象的点标识等。查找协议用于搜索查找相应的组件和管控对象。

下面针对这两种协议进行详细定义和描述。

[1]. 注册协议

注册协议使组件可以注册自己的实现角色以及对应的管控对象信息等。提交注册申请的组件为注册发起方, 注册协议的示意图如图A.8所示。

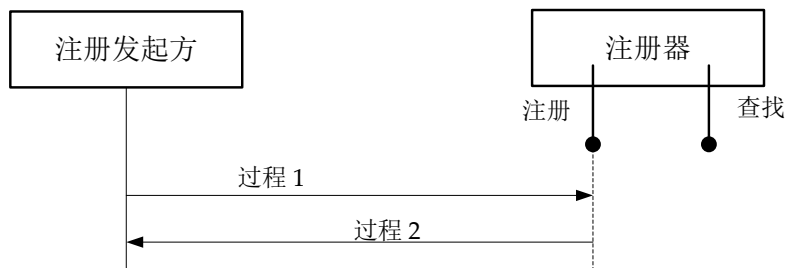


图 A.8 注册协议示意图

如图A.8所示, 注册发起方向注册器发起注册请求, 注册器返回响应。

过程 1: 注册发起方调用注册器的注册方法, 提供组件信息和角色信息 (管理的管控对象点标识列表)。

过程 2: 注册器返回本次绑定的成功或失败结果。

[2]. 查找协议

通过查找协议, 组件可以查找需要接入的组件 (对于组件之间的通信), 或是通过语义查询 semantic-query 查找指定的管控对象。发起查找请求的组件为查找请求方, 由注册器提供查找响应。查找协议的示意图如图A.9所示。

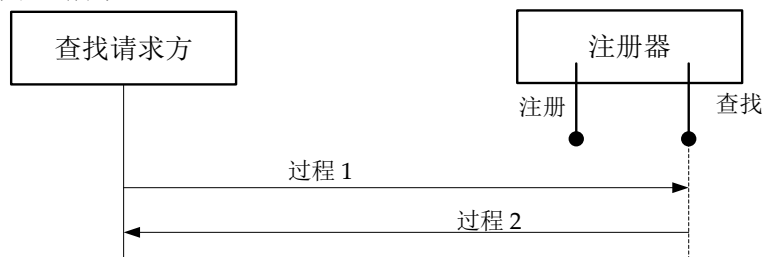


图 A.9 查找协议示意图

如图A.9所示, 查找请求方向注册器发起查找请求, 注册器返回查找结果。

过程 1: 查找请求方调用注册器的查找方法, 其中的查找表达式指明查询对象的类型, 如是组件还是管控对象。

过程 2: 注册器返回解析的组件或管控对象的接入统一资源标识符。

A.2.2 应用程序接口

A.2.2.1 接口定义

本部分定义了两种类型的应用程序编程接口 (API), 包括组件访问接口和注册器访问接口。组件访问接口用于组件到组件的通信过程, 注册器访问接口用于组件和注册器之间的通信过程。

组件（网关，存储器和应用单元）的访问接口定义了查询和推送方法，用于其他组件的访问。查询用于从组件获取或订阅数据，推送用于向组件推送数据。通过查询和推送两种方法，可以实现信息读取协议、信息发送协议和信息触发协议三个通信协议。

注册器的访问接口定义了注册和查找方法，从而支持组件（网关、存储器和应用单元）接入注册器，获得注册器提供的服务。通过注册和查找两种方法，可以实现注册和查找两个协议。

A.2.2.2 传输消息的数据结构

在泛在绿色社区控制网络中，无论是组件之间的通信还是组件与注册器之间的通信，都由RPC方法实现对组件或注册器的接入。如图A.10所示，调用方调用被调用方的接口方法，发起请求消息，而被调用方返回响应消息。

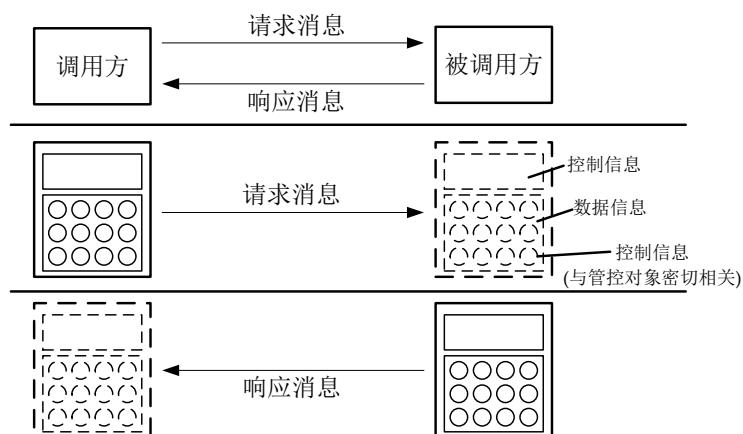


图 A.10 传输消息的数据结构

在请求方与响应方之间的通信过程中，请求消息和响应消息具有相同的数据结构，包括消息头和消息体。消息头包含控制信息，如查询表达式（用于组件之间的通信），查找表达式（用于组件和注册器之间的通信），确认信息，失败信息等等。消息体包含携带数值的管控对象或管控对象集，例如来自传感器的读数和对执行器的指令（用于组件之间的通信），或是组件的实现角色和对对应管控对象的点标识（用于组件和注册器之间的通信）。若控制信息与管控对象或管控对象的特定值密切相关，则该控制对象也是消息体的一部分。

A.2.2.3 组件的访问接口

[1]. 查询方法

调用查询方法的格式为：**Transport query (Transport t)**。

当信息请求方通过调用信息提供方的查询方法请求获取数据时，发送一个Transport类型的消息实例t，在t的消息头中指定查询表达式。查询表达式的一个重要属性是查询的类型，用type表示，有两种不同的取值，storage或是stream。当type属性取值为storage时，表明应用信息读取协议。当type属性取值为stream时，表明应用信息触发协议。不同的协议会触发信息提供方不同的处理行为。

当type="storage"时，查询请求消息t的消息头中携带有查询表达式，消息体被信息提供方忽略。查询表达式具有如下属性：

- id 用于标识该查询
- type 设置为“storage”，表明执行信息读取协议
- acceptableSize 查找请求方在一个RPC响应中，一次所能接收的“值”元素的最大数目
- cursor 用于获取未传完的后续数据。cursor属性不会出现在查找请求方发送的第一个查询请求消息中，其值由信息提供方根据响应消息的发送情况设定

当type="storage"时，在查询的响应消息中，消息头携带有查询表达式以及响应结果，通常以OK或Error表示。在查询响应携带的查询表达式中，如果存在cursor属性，则意味着还有后续数据要继续传送。如果没有cursor则意味着所有数据已经传送完毕。查询响应消息头中的OK表明信息提供方成功的完成了查询方法对应的数据传送过程，Error表示信息提供方在执行查询方法调用时出现了错误。Error也可提供具体的错误描述信息，供信息请求方检查。如果查询响应携带OK，则其消息体中携带相应的管控对象或是管控对象集的相关信息，否则该消息体被接收方忽略。

当type="stream"时，查询请求消息t的消息头中携带有查询表达式，消息体被信息提供方忽略。查询表达式具有如下属性：

- id 用于标识该查询。当周期性更新查询时，该id值不变

type	设置为“stream”，表明执行信息触发协议
ttl	设置该查询在信息提供方处的生存时间，单位为秒
callbackData	当数据与查询匹配时，数据发送的目标组件的统一资源标识符
callbackControl	当需要发送控制类信息时，接收该类信息的目标组件的统一资源标识符
acceptableSize	可选,数据信息回送方标识的组件在一次数据传输过程中，一次所能接收的“值”元素的最大数目

当type=”stream”时，在查询的响应消息中，消息头携带有查询表达式以及以OK或Error表示的响应结果。查询响应消息头中的OK表明信息提供方成功的完成了查询方法对应的数据传送过程，Error表示信息提供方在执行查询方法调用时出现了错误。Error也可提供具体的错误描述信息，供信息请求方检查。接收方忽略响应消息的消息体。当有匹配查询条件的数据更新时，信息提供方调用数据信息回送方对应组件的推送方法。信息提供方发送的请求消息t的消息头需要复制查询表达式，供数据信息回送组件获取足够的信息了解消息体中的数据内容。

[2]. 推送方法

调用推送方法的格式为：**Transport data (Transport t)**。

信息请求方将向目标组件传送的管控对象或管控对象集的相关信息封装在推送请求消息t的消息体中。如果在信息触发协议中调用推送方法，从信息提供方数据信息回送组件传送数据，则t的消息头中复制有相应的查询方法消息头中携带的查询表达式。

在推送的响应消息中，消息头携带以OK或Error表示的响应结果，OK表明目标组件成功的完成了推送方法对应的数据写入过程，Error表示目标组件在执行推送方法调用时出现了错误。Error也可提供具体的错误描述信息，供信息请求方检查。响应消息的消息体被接收方忽略。

A.2.2.4 注册器的访问接口

[1]. 注册器访问接口的数据结构

对于注册器支持的协议注册和查找而言，具有相同的数据结构，都包括消息头Header和消息体Body，如图A.11所示。其中消息头中包含有查找表达式s，以及成功或失败的状态信息。消息体部分则描述了组件的实现角色，以及对应的管控对象信息。

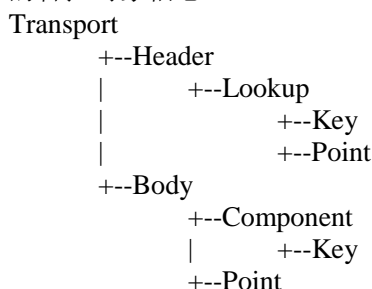


图 A. 11 注册器访问接口的数据结构

[2]. 注册方法

注册方法的格式为**Transport registration (Transport t)**。

当注册发起方调用注册器的注册方法时，完成自身的实现角色以及对应的管控对象信息的注册。基于该方法可以实现注册协议。注册器可以处理两种类型的注册信息，分别是描述组件实现角色的信息，以及描述管控对象的信息。

当处理组件相关的注册信息时，注册请求消息t的消息体中携带关于组件的信息。在注册响应消息中，消息头携带以OK或Error表示的响应结果，OK表明注册器成功的完成了注册方法对应的注册过程，Error表示注册器在执行注册方法调用时出现了错误。Error也可提供具体的错误描述信息，供注册发起方检查。

当处理描述管控对象的注册信息时，注册请求消息t的消息体中携带管控对象语义信息相关的信息。对于管控对象，除了点标识外，其他属性取决于具体的业务应用以及现场网络的实际部署。

[3]. 查找方法

查找方法的格式为**Transport lookup (Transport t)**。

查找方法用于组件查找某特定管控对象的归属组件，或者查找符合一定约束条件的匹配管控对象。基于该方法可以实现查找协议。当信息请求方通过调用注册器的查找方法时，在消息实例t的消息头中指明查找表达式。查找表达式的一个重要属性是查找的对象，用type表示，有两种不同的取值，分别为component和point。根据不同的类型，注册器会采用不同的处理方法。

当type="component"时，信息请求方请求查找正确的接入组件，注册器解析组件后返回接入统一资源标识符。查找请求消息t的消息头中携带有查找表达式，消息体被注册器忽略。查找表达式具有如下属性：

id 用于标识该查找

type 设置为“component”，表明需要查找正确的接入组件

当type="component"时，在查找的响应消息中，消息头携带有查找表达式以及以OK或Error表示的响应结果。OK表明注册器成功的完成了查找方法对应的组件查找过程，Error表示注册器在执行查找方法调用时出现了错误。Error也可提供具体的错误描述信息，供信息请求方检查。如果查找响应携带OK，则其消息体中携带解析得到的接入组件的统一资源标识符，否则该消息体被接收方忽略。

当type="point"时，信息请求方通过语义查询查找符合一定约束条件的管控对象，注册器返回匹配管控对象的信息，包括点标识、属性文件等等。查找请求消息t的消息头中携带有查找表达式，消息体被注册器忽略。查找表达式具有如下属性：

id 用于标识该查询。当周期性更新查询时，该id值不变

type 设置为“point”，表明需要查找管控对象

当type="point"时，在查找的响应消息中，消息头携带有查找表达式以及以OK或Error表示的响应结果。OK表明注册器成功的完成了查找方法对应的管控对象查找过程，Error表示注册器在执行查找方法调用时出现了错误。Error也可提供具体的错误描述信息，供信息请求方检查。如果查找响应携带OK，则其消息体中携带与查找匹配的管控对象的属性文件，否则该消息体被接收方忽略。

A.2.3 数据结构

A.2.3.1 总则

本部分针对组件之间，以及组件与注册器之间的通信协议，定义了相应的数据结构。所有的消息遵从可扩展标示语言（XML）格式。本标准中的“类”对应XML中的“元素”。为了方便理解，本标准规定，类的名称以大写字母开始，XML元素的名称以小写字母开始。

A.2.3.2 组件之间通信协议的数据结构

A.2.3.2.1 类

针对泛在绿色社区控制网络组件之间的通信协议，定义了如下十种类：

Transport

Header

Body

PointSet

Point

Value

Query

Key

OK

Error

A.2.3.2.2 Transport 类

Transport类用于在一个消息中同时传输数据平面信息和控制平面信息，类成员包括了一个Header对象和一个Body对象。Transport类没有属性。

Trnasport类的实例如下：

```
<transport>
  <header>.....</header>
  <body>.....</body>
</transport>
```

A.2.3.2.3 Header 类

Header类用于承载控制平面信息,包括查询表达式、失败消息等，类成员包括了Query对象、OK对象以及Error对象。Header类没有属性。

Header类的实例如下：

实例一：

```
<header>
```

```

    <query>.....</query>
</header>

```

实例二:

```

<header>
    <OK/>
</header>

```

A.2.3.2.4 Body 类

Body类用于承载管控对象或管控对象集的相关数据信息，以及关系紧密的控制信息，类成员包括PointSet对象和Point对象。Body类没有属性。

Body类的实例如下:

实例一:

```

<body>
    <pointSet id="http://gw.foo.org/tv1">...</pointSet>
    <pointSet id="http://gw.foo.org/refrigerator1">...</pointSet>
    <pointSet id="http://gw.foo.org/pot1">...</pointSet>
</body>

```

实例二:

```

<body>
    <point id="http://gw.foo.org/tv1/power">...</point>
    <point id="http://gw.foo.org/tv1/switch">...</point>
    <point id="http://gw.foo.org/pot1/power">...</point>
</body>

```

实例三

```

<body>
    <pointSet id="http://gw.foo.org/room1/light">...</pointSet>
    <pointSet id="http://gw.foo.org/room1/hvac">...</pointSet>
    <point id="http://gw.foo.org/tv1/power">...</point>
    <point id="http://gw.foo.org/pot1/power">...</point>
</body>

```

A.2.3.2.5 PointSet 类

PointSet类聚合相关的Point对象和PointSet对象，类成员包括PointSet对象和Point对象。PointSet类的属性为id，是基于URI的标识符，用来唯一标识PointSet对象。

PointSet类的实例如下:

实例一:

```

<pointSet id="http://gw.foo.org/room1">
    <pointSet id="http://gw.foo.org/room1/light">...</pointSet>
    <pointSet id="http://gw.foo.org/room1/hvac">...</pointSet>
</pointSet>

```

实例二:

```

<pointSet id="http://gw.foo.org/tv1">
    <point id="http://gw.foo.org/tv1/power">...</point>

```

```

    <point id="http://gw.foo.org/tv1/switch">...</point>
    <point id="http://gw.foo.org/tv1/channel">...</point>
</pointSet>

```

实例三：

```

<pointSet id="http://gw.foo.org/room1">
  <pointSet id="http://gw.foo.org/room1/light">...</pointSet>
  <pointSet id="http://gw.foo.org/room1/hvac">...</pointSet>
  <point id="http://gw.foo.org/room1/temperature">...</point>
  <point id="http://gw.foo.org/room1/humidity">...</point>
</pointSet>

```

A.2.3.2.6 Point 类

Point类用以描述管控对象，类成员包括Value对象。Point类的属性为id，用来唯一标识Point对象，如本标准定义的点标识pointID。

Point类的实例如下：

```

<point id="http://gw.foo.org/room1/temperature">
  <value time="2009-09-01T00:00:00.0000000+09:00">25.5</value>
  <value time="2009-09-01T00:01:00.0000000+09:00">25.6</value>
  <value time="2009-09-01T00:02:00.0000000+09:00">25.6</value>
  <value time="2009-09-01T00:03:00.0000000+09:00">25.7</value>
</point>

```

A.2.3.2.7 Value 类

Value类包括了具体的数据值。来自传感器的输入数据或是发往执行器的指令数据都可以封装在Value对象中。Value类没有类成员，其属性为time，按照W3C定义的时间戳标准格式，记录输入数据的产生事件以及指令数据的执行时间。

Value类的实例如下：

```

<value time="2009-09-01T00:00:00.0000000+09:00">true</value>
<value time="2009-09-01T00:01:00.0000000+09:00">>false</value>
<value time="2009-09-01T00:02:00.0000000+09:00">10</value>
<value time="2009-09-01T00:03:00.0000000+09:00">0</value>
<value time="2009-09-01T00:03:00.0000000+09:00">3.4</value>
<value time="2009-09-01T00:03:00.0000000+09:00">0.5323</value>
<value time="2009-09-01T00:03:00.0000000+09:00">HIGH</value>
<value time="2009-09-01T00:03:00.0000000+09:00">MID</value>
<value time="2009-09-01T00:03:00.0000000+09:00">LOW</value>

```

A.2.3.2.8 Query 类

Query类针对类型为storage以及stream的查询方法管理相应的查询表达式，类成员为Key对象。Query对象的属性包括：

id	用于标识该查询。
type	标识查询方法的类型，取值为“storage”或“stream”
acceptableSize	接收方在一个RPC响应中，一次所能接收的Value对象的最大数目
cursor	用于获取未传完的后续数据。仅在type=“storage”时才存在
ttl	设置该查询在信息提供方处的生存时间，单位为秒

callbackData 当数据与查询匹配时，数据发送的目标组件的统一资源标识符
callbackControl 当需要发送控制类信息时，接收该类信息的目标组件的统一资源标识符
Query类的实例如下：

实例一：

```
<query id="6229c37f-970d-9292-83e4-7c0e54733f8a"
      type="storage"
      acceptableSize="20"
      cursor="dab751ed-0133-4ce4-8b7d-ba5c54ce4fb5" >
  <key>.....</key>
  <key>.....</key>
</query>
```

实例二：

```
<query id="9eed9de4-1c48-4b08-a41d-dac067fc1c0d"
      type="stream"
      ttl="60"
      callbackData="http:// foo.org/axis/services/GUTAPI"
      callbackControl="http:// foo.org/axis/services/GUTAPI"
  <key>.....</key>
  <key>.....</key>
</query>
```

A.2.3.2.9 Key 类

Key类用于描述查询表达式，无类成员。**Key**对象的属性包括：

id	对应管控对象或管控对象集的目标标识符
attrName	随后定义的属性的名称
eq	若 key 对象的属性值等于某给定值，则为真，否则为假
neq	若 key 对象的属性值不等于某给定值，则为真，否则为假
lt	若 key 对象的属性值小于某给定值，则为真，否则为假
gt	若 key 对象的属性值大于某给定值，则为真，否则为假
lteq	若 key 对象的属性值小于或等于某给定值，则为真，否则为假
gteq	若 key 对象的属性值大于或等于某给定值，则为真，否则为假
select	枚举类型，可能具有{ maximum , minimum }两个属性值
trap	若检测到了发生某事件，则取值为{ changed }（仅在 query type="stream" 时有效）
stream	若组件接收数据，则取值为“in”；若组件发送数据，则取值为“out”

Key类的实例如下：

实例一：

```
<key id="http:// gw.foo.org/room1/temperature"
     attrName="time"
     select="maximum"/>
```

实例二：

```
<key id="http:// gw.foo.org/room1/temperature"
     attrName="time"
     lteq="2009-10-01T00:00:00.0000000+08:00"
     gteq="2009-09-01T00:00:00.0000000+08:00" />
```

实例三:

```
<key id="http:// gw.foo.org/room1/temperature"
    attrName="time"
    trap="changed"
```

实例四:

```
<key id="http:// gw.foo.org/room1/temperature"
    attrName="value"
    trap="changed" />
```

A.2.3.2.10 OK 类

OK类表明请求被成功的接受, OK类没有类成员, 也没有属性。

OK类的实例如下:

```
<OK/>
```

A.2.3.2.11 Error 类

Error类承载出错信息, 没有类成员。Error类的属性为type, 用于描述错误类型等具体错误信息。

Error类的实例如下:

实例一:

```
<error type="syntax">
    Malformed XML Error at line X.
</error>
```

实例二:

```
<error type="authorization">
    You are not permitted to access the requested resource.
</error>
```

A.2.3.3 组件与注册器之间的通信协议的数据结构

A.2.3.3.1 类

针对社区智能节电控制网络组件与注册器之间的通信协议, 定义了如下九种类:

Transport
Header
Body
Component
Point
Lookup
Key
OK
Error

A.2.3.3.2 Transport 类

Transport类用于在一个消息中同时传输数据平面信息和控制平面信息,类成员包括了一个Header对象和一个Body对象。Transport类没有属性。

Transport类的实例如下:

实例一:

```
<transport>
    <header>.....</header>
    <body>.....</body>
</transport>
```

实例二:

当向注册器注册管控对象的属性时,需要创建管控对象的属性文件,并在向注册器提交时指明命名空间,如:

```
<transport xmlns:s="..."
  <body>.....</body>
</transport>
```

A.2.3.3.3 Header 类

Header类用于承载控制平面信息,包括查找表达式、失败消息等,类成员包括了Lookup对象、OK对象以及Error对象。Header类没有属性。

Header类的实例如下:

实例一:

```
<header>
  <lookup>.....</lookup>
</header>
```

实例二:

```
<header>
  <OK/>
</header>
```

A.2.3.3.4 Body 类

Body类用于承载解析得到的组件(网关、存储器、应用单元)或管控对象属性信息,类成员包括Component对象和Point对象。Body类没有属性。

Body类的实例如下:

实例一:

```
<body>
  <component... > ...</component>
</body>
```

实例二:

```
<body>
  <point id="..." ... />
  <point id="..." ... />
  <point id="..." ... />
</body>
```

A.2.3.3.5 Component 类

Component类是对网关、存储器和应用单元的抽象,类成员包括Key对象。Component类的属性包括:

name	描述组件的名称
uri	标识某组件的接入统一资源标识符
priority	冗余数据集的接入优先级(可选属性)。当存在多于一个的候选组件可以接入时(例如存在组件的备份),该属性决定了优先接入哪个接入组件。应选择Priority的值大的组件优先接入
support	组件支持的协议类型(信息读取协议,信息发送协议,信息触发协议等)
expires	注册的有效期限(可选属性)。通过该属性,注册器可以掌握某注册信息的有效期限,并且每过1秒就将该属性值减去1.当该属性值为0时,注册器将拒绝该注册消息。如果注册申请方并未提供该属性,则由注册器自行决定对应注册条目的有效期限

Component类的实例如下:

实例一: 一个典型的网关类实例

```

<component      name="myGW"
                uri="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/FIAPBACnetWSGN"
                support="FETCH|TRAP"
    <key          id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/DB1"
                stream="in"
                limit="1" />
    <key          id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/RH1"
                stream="in"
                limit="1" />
</component>

```

其中limit="1"指的是myGW只有一个数据的缓存。

实例二：一个典型的存储器类实例

```

<component      name="myStorage"
                uri="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/FIAPStorage"
                support="FETCH|WRITE"
    <key          id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/DB1"/>
    <key          id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/RH1"/>
    .....
</component>

```

A.2.3.3.6 Point 类

Point类描述了管控对象，但不包括管控对象的Value值，没有类成员。Point类的属性在本标准中仅定义为id，用于标识该管控对象，如pointID。除了id外，支持定义任何类型的Point类属性，如类型，可读写性，物理位置等。

Point类的实例如下：

实例一：

```

<point id="X"
       s:type= "BINARY_INPUT"
       s:writable="false"
       s:location="Building2F221MeetingRoom1"/>

```

实例二：

```

<point id="Z"
       s:type= "ANALOG_INPUT"
       s:writable="false"
       s:location="Building2F221MeetingRoom1"/>

```

其中，S是配置文件的命名空间的命名空间前缀，例如

XMLNS: S="http://application-dependent.org/AttributeSet/"

A.2.3.3.7 Lookup 类

Lookup类管理查找表达式，类成员为Key对象和Point对象。Lookup类的属性包括：

id 用于标识该查找。

type 标识查找方法的类型，取值为“component”或“point”

Lookup类的实例如下：

实例一：查找组件

```
<lookup      id="6e5a0e85-b4a0-485f-be54-a758115317e1"
             type="component"
             <key id="http://fiap-gw.gutp.ic.i.utokyo.ac.jp/EngBldg2/10F/102A2/DB1" .../>
</lookup>
```

实例二：查找管控对象

```
<lookup      id="3f2504e0-4f89-11d3-9a0c-0305e82c3301"
             type="point"
             <point s:type="BINARY_INPUT" s:location="Building2F221MeetingRoom1" .../>
</lookup>
```

其中，S是配置文件的目标命名空间的命名空间前缀，例如

XMLNS: S = "http://application-dependent.org/AttributeSet/"

A.2.3.3.8 Key 类

Key类用于描述查找表达式，以及管控对象的属性文件。Key类没有类成员，属性包括：

id	用于标识管控对象的标识符，如pointID
attrName	对象搜索属性的名称
limit	数据缓存区的大小
eq	若key对象的属性值等于某给定值，则为真，否则为假
neq	若key对象的属性值不等于某给定值，则为真，否则为假
lt	若key对象的属性值小于某给定值，则为真，否则为假
gt	若key对象的属性值大于某给定值，则为真，否则为假
lteq	若key对象的属性值小于或等于某给定值，则为真，否则为假
gteq	若key对象的属性值大于或等于某给定值，则为真，否则为假
stream	若组件接收数据，则取值为“in”；若组件发送数据，则取值为“out”

Key类的实例如下：

如果一个组件保存标识符为“X”的管控对象从2010年1月1号早上9点之后的数据，则对应的key对象为：

```
<key      id="X"
          attrName="time"
          gteq="2010-01-01T00:00:00+09:00"/>
```

如果一个组件从传感器X中获取数据，则对应的key为：

```
<key      id="X"
          attrName="time"
          stream="in"/>
```

如果一个组件向执行器X发送指令，则对应的key是：

```
<key      id="X"
          attrName="time"
          stream="out"/>
```

如果一个组件可以提供输入和输出数据流，则对应的key是：

```
<key      id="X"
          attrName="time"
          stream="in|out"/>
```

A.2.3.3.9 OK 类

OK类表明请求被成功的接受，OK类没有类成员，也没有属性。

OK类的实例如下：

```
<OK/>
```

A.2.3.3.10 Error 类

Error类承载出错信息，没有类成员。Error类的属性为type，用于描述错误类型等具体错误信息。

Error类的实例如下：

实例一：

```
<error          type="syntax">
```

```
    Malformed XML Error at line X.
```

```
</error>
```

实例二：

```
<error          type="priority">
```

```
    No redundant Storage existing according to the specified point ID.
```

```
</error>
```

A.2.4 协议绑定

泛在绿色社区控制网络支持几乎所有的现有数据传输协议，如SMTP，SIP，FTP，HTTP等，并且支持新兴或未来出现的新型数据传输协议，如CoAP协议等等。但考虑到应用以及系统的具体部署实现，本标准推荐采用SOAP，HTTP或SIP作为缺省的绑定协议。如果强调系统的可运营和可管理性，则推荐使用SIP协议。

附录 B
(参考性附录)
中英文对照

AAA	Authentication, Authorization, and Accounting	鉴权、授权、计费
API	Application Programming Interface	应用程序接口
BACNet	Building Automation and Control networks	楼宇自动化与控制网络
CoAP	Constrained Application Protocol	约束应用协议
Ethernet		以太网
FTP	File Transfer Protocol	文件传输协议
HTTP	Hypertext Transfer Protocol	超文本传输协议
IP	Internet Protocol	因特网协议
IPv6	Internet Protocol Version 6	因特网协议第6版
I/O	Input/Output	输入输出端口
KNX	Konnex	住宅和楼宇控制标准
Lontalk		控制网络标准协议
Modbus		工业现场总线协议
PLC	Power Line Communication	电力线通信
RF433		一种无线收发方法
RF485		一种无线收发方法
RPC	Remote Procedure Call	远程过程调用
RS232		串行通信标准接口
RS232C		一种计算机串行接口
RS485		串行通信标准接口
SIP	Session Initiation Protocol	会话发起协议
SMTP	Simple Mail Transfer Protocol	简单邮件传输协议
SOAP	Simple Object Access Protocol	简单对象访问协议
TCP	Transmission Control Protocol	传输控制协议
TTL	time to live	生存时间
URI	Uniform Resource Identifier	统一资源标识符
W3C	World Wide Web Consortium	万维网联盟
WLAN	Wireless Local Area Network	无线局域网
XML	eXtensible Markup Language	可扩展标示语言
Zigbee		低功耗个域网协议
6LowPAN	IPv6 over Low power WPAN	引入IPv6的无线个域网标准