CoRE
Internet-Draft
Intended status: Informational
Expires: September 1, 2012

P. van der Stok, Ed.
Philips Research
K. Lynn
Consultant
A. Brandt
Sigma Designs
February 29, 2012

CoRE Discovery, Naming, and Addressing
draft-vanderstok-core-dna-00

Abstract

This is a working document intended to focus discussion and refine draft language for the CoAP protocol specification (or other proposed standards) in the areas of discovery, naming, and addressing. Engineering tradeoffs become more challenging in constrained environments; therefore discovery, naming, and addressing are considered within the context of adjacent topics that may impact or be impacted by design choices in the subject areas. Special emphasis is placed on group definition and discovery. Examples show how groups can be represented in CoAP Resource Directories or DNS-SD.

Table of Contents

1. Introduction

The CoRE working group is chartered to design and standardize a Constrained Application Protocol (CoAP) for resource constrained devices and networks [I-D.ietf-core-coap]. The requirements for CoRE are documented in [I-D.shelby-core-coap-req]. This draft discusses the requirements on service discovery for M2M and interactive applications using resource constrained devices. We propose the use of DNS-SD [I-D.cheshire-dnsext-dns-sd] and Resource Directory (RD) [I-D.shelby-core-resource-directory] to satisfy the requirements. The proposal relies heavily on naming and addressing conventions. Special emphasis is placed on the definition, naming, and discovery of groups.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]. Addtional privileged words are described below.

A "device" is a physical processor connected to at least one link through a network interface. Each interface has at least one IP unicast address. The IP address is optionally bound to a host name, which may be a Fully Qualified Domain Name (FQDN).

An "end-point" corresponds to a "service" and is indentified by a unique {protocol, host, port} tuple. The identity of an endpoint may be specified by the 'scheme' and 'authority' parts of a URI [RFC3986]. 'Protocol' is a label that indicates application and transport protocol bindings as well as default port (if port is not specified) and possibly default semantics such as web-linking [RFC5988]. 'Host' corresponds to the [RFC3986] ABNF production as updated by [I-D.ietf-6man-uri-zoneid].

A "service type", e.g. _bldg-ctrl._tcp, is equivalent to the 'protocol' label described above. It identifies an application protocol, typically defined by a Standards Development Organization (SDO), and is ultimately registered with IANA [I-D.cheshire-dnsext-dns-sd]. The SDO may additionally specify service subtypes (e.g. _light, _onoff-control, _air-flow ...) to designate units of functionality, the attributes of the subtypes, and the primitives acting on the attributes.

Any attribute of an end-point that can be acted upon by REST methods will be represented as a "resource" and must be identified by a URI, that is, an end-point plus a 'path' component [RFC3986].

A "Function Set" is a service subtype with a standarized set of resources and behaviors that may be accessed through a REST interface. A Function Set will typically be decribed by a base URI plus an interface definition as described in [I-D.shelby-core-interfaces]. The interface definition may specify the naming patterns of subordinate resources and the methods that act on them as defined by the SDO.

A "Collection" is a set of two or more homogeneous subordinate resources that may be acted upon in the aggregate by sending messages to their parent resource, or individually by sending messages to the collection member.

A "Device type" describes a standardized set of function sets that satisfy a use case for a hosting device.

A "group" is a set of devices.

A "multicast group" is a group of devices that share a multicast address. The multicast address is optionally bound to a FQDN that identifies the multicast group. For the purposes of this document, (multicast) groups generally host the same Function Set.

A "scope" is a possible set of devices or groups. A scope may be realized logically, e.g. as a DNS domain, or a zone which is a unit of delegation (partition) of a domain; physically, e.g. the local link; or administratively, e.g. as a set of links.

1.2. Motivation

In this draft, we focus and expand discussions on requirements pertaining to discovery, naming, and addressing, including REQ8 of the CoRE charter:

REQ8: A definition of how to use CoAP to advertise about or query for a Device's description. This description may include the device name and a list of its Resources, each with a URL, an interface description URI (pointing e.g. to a Web Application Description Language (WADL) document) and an optional name or identifier. The name taxonomy used for this description will be consistent with other IETF work, e.g. draft-cheshire-dnsext-dns-sd. [charter]

The basis of this draft originated in [I-D.vanderstok-core-bc].

1.3. Applicability

TBD

2. Architecture

This section illustrates the aspects of naming schemes and their support by DNS-based Service Discovery (DNS-SD) [I-D.cheshire-dnsext-dns-sd] , Extended Multicast DNS (xmDNS) [I-D.lynn-dnsext-site-mdns], and the Resource Directory (RD) [I-D.shelby-core-resource-directory] on a set of network architectures.

The basic network for low-power nodes can be composed of low-resource nodes sharing the same IPv6 prefix and connected to low-power links like IEEE 802.15.4, ITU-T G.9959, or Powerline. The "lowpan" is a good example of such a network [RFC4944], [RFC6282]. The network can be either isolated or connected. This draft assumes that application profiles are defined above coap or http, for example, applications as specified by the ZigBee Smart Energy Profile 2.0 (SEP2), Obix, or BACnet IT working groups. The naming and discovery solutions presented here are applicable to multiple interconnected subnets. Example network architectures are:

- An isolated lowpan consists of at least two lowpan devices, one of which is an edge router that is not connected to a backbone. A Resource Directory may be situated on the edge router. Alternatively, xmDNS responders may execute on each device.
- A connected lowpan consists of at least two lowpan devices, one of which is an edge router that is connected to a backbone. A Resource Directory may be situated on an edge router. Alternatively, xmDNS responders may execute on each device or the DNS-SD service may be available over the backbone.
- Interconnected lowpans consist of at least two lowpans connected via edge routers to the same backbone. A Resource Directory may be situated on each edge router. Alternatively, xmDNS responders may execute on each device or the DNS-SD service may be available over the backbone.
- A site is a set of interconnected subnets that is locally administered. A site may include zero or more lowpans. Border routers may prevent some messages from passing into or out of the site.

In certain scenarios, the domain may correspond to the network topology. In the general case, the domain and network subnet structure may differ.

3. Use Cases

The use of service discovery is presented in two environments: (1) interactive service discovery, and (2) M2M service discovery. From the use cases we derive the types of queries that service discovery should support. In particular, a primary motivation is the discovery of groups that support a given Function Set.

3.1. Discovery scope

Service discovery has a scope that can be defined and realized with domain names. The authority part of a URI [RFC3986] can express the location of the device hosting the service. A common example is the naming associated with the structure of a building. A device may acquire a FQDN that relates directly to its location in the building. For example, power-strip.office4.floor1.example.com (or shorter ps.o4.f1.example.com) refers to a power-strip with device name "power-strip" (or short "ps") in office4 at floor1 in the building of the company example.com. Another naming scheme can be functional like TV1.media.IT.example.com, possibly refering to TV number 1 maintained by the media group of the IT department of the example organisation. Domain naming can be used to express that devices are situated in the same building area or belong to the same organisational units. Multiple FQDNs can identify a given device.

The DNS provides the mapping from Fully Qualified Domain Name (FQDN) to network address and vice-versa. The binding of FQDN to a physical device (for example, assigning a given FQDN to the TV in the corner) depends on the operational conditions as described below.

3.2. Interactive mapping

In the residential context, naming of the device is done by the occupant of the home. After connecting the device to the network, an IP-address is assigned, possibly based on the EUI-64 value of the network interface. The occupant can use a remote control with a graphical user interface to display all devices that provide a given service (e.g. a "lighting" service). The remote control prompts the occupant to identify the (default named) devices, possibly accompanied by on/off switching, barcode scanning, or other manual intervention. The occupant can provide a meaningful name that will be bound to that device. The installation steps can be as follows: Service discovery returns all interfaces on which the specified service is available. The occupant, with or without additional physical support, establishes the binding between an IP address (based on MAC address or other unique identifier) and the name of the device providing the service, plus its relationship with other devices or function sets in the system. In some cases a device has

multiple names, and an additional mapping between user specified name and an automatically generated DNS name is supported.

3.3. M2M mapping

In the commercial context (e.g. office buildings) it is usual to employ a Comissioning Tool (CT) to provide the mapping from physical device to IP network address or FQDN. The professional context is more rigid than the home because the absence of devices and also the unwanted presence of devices needs to be detected. Devices are named by an architect or installation contractor. Names can be generated automatically and need not be human-friendly. The CT contains the names and the physical locations of the devices. At commissioning time the interfaces have acquired a network address, possibly based on the EUI-64. The physical device is identified by reading in a unique identifier (e.g. EUI-64 of interface, UUID of device) with a reader (e.g. barcode reader). Consequently, the device name to network address binding is stored into DNS (or elsewhere). Alternatives for identifying devices are pushing buttons on the device or remotely switching on/off the device.

3.4. Function Set grouping

Groups can be used to express that devices are related (e.g. HVAC equipment controlled by the closest temperature sensor). Grouping is also necessary when a set of Function Sets has to react together, more or less synchronously, to a sequence of commands sent by one or more devices. A common example is provided by lighting applications where a subset of lights in the building are dimmed to the same level, set to the same colour, or switched off simultaneously. Another example is provided by a power-strip supporting a set of power-outlets. Power-outlets are switched on/off individually or all together. Other examples concern the home, such as a "sleep mode" setting of all media devices in the home when the user activates the night scene.

Group naming is done the same way as for device naming. Related devices are grouped and named. The group name is constructed like a FQDN with the group name as prefix. Adressing the group can be done in two ways: (1) by addressing each Function Set of the group individually (which requires serial access), or (2) by defining a multicast address for a multicast group. In the latter case, each hosting device must enable reception of the messages sent to the multicast address. The Function Sets of the multicast group must have identical port number and path, because their values are specified in a single multicast message.

The Function Set can contain a collection of resources of the same

subtype. The Function Set path postfixed with an identifier refers to the individual resources of the collection. For example: the /path/light points to the resource collection of the service subtype light. Each member of the collection can be identified with /path/light/x, with x in {1,2,3,..}. Consequently, the /path/light/1/onoff specifies the onoff resource of collection member 1 of Function Set with /path/light, and /path/light/onoff specifies the resource onoff of all collection members contained by the Function Set with /path/light. When /path/light/onoff is used in a multicast message, it is interpreted as a message to a single light resource by devices having only one, and to all members of the collection for devices having several light resources.

It is expected that SDOs will define group naming conventions, extending the service type name conventions for individual devices.

Figure 1 illustrates some of the concepts described above. A device is identified with the name "Power-strip". In the example, no domain names are associated with the device, and the name "power-strip" resolves to a Unique Local Address (ULA)[RFC4193]. The device provides two end-points: one delivers a http service and the second a CoAP service. The http server and CoAP server share the same IP-address and use different ports 80 and 61616 respectively. Two different service subtypes, one identified by Function Set, ps, and one identified by Function Set, pm, are supported by one end-point. The Function Set "Power strip" contains a collection of four resources, "Outlet 1" to "Outlet 4", each one accessible via the accompanying paths /ps/1 to ps/4. The path /ps interfaces to the entire resource collection. The attribute "output" is defined in the service subtype specification.

TBD..... Relation with [I-D.shelby-core-interfaces]

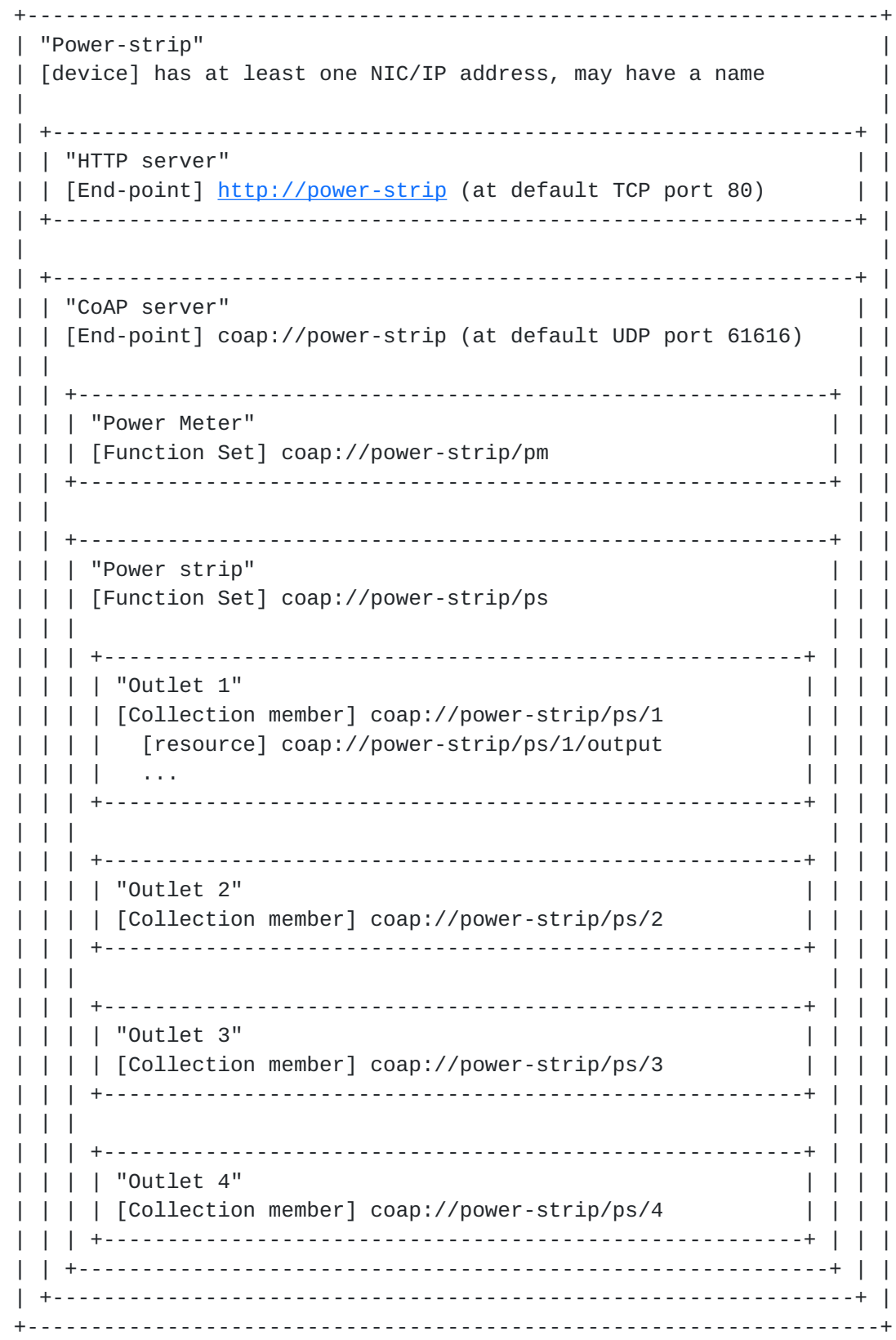

Figure 1: device with end-points, Function Sets and resources

3.5. Discovery queries

Service discovery should support that a device can learn its domain and all the devices within a domain providing a given service (e.g. temperature measurement). Devices need to learn the groups to which they belong and learn all the members of those groups. This section motivates that a discovery service supports the following queries:

Goal	Description
Name_resolution	Resolve the group or device name to IP address and optional port number
Return_device	Return all devices supporting a given service (sub-)type within a given domain
Create_group	Create a group of devices possibly hosting a given service (sub-)type within a given domain
Enroll_member	Enroll a given device as member of a given group
Remove_member	Remove a given device as member of a given group
Return_group	Return all groups of which a given device is a member
Return_member	Return devices belonging to a given group

Name_resolution is supported by DNS and CoAP resource discovery. Names are required in the context of home control and manual setup of installations. Names are persistent and meaningful as compared to IP addresses and are preferably used in applications when IP addresses can change.

Return_device is the most common use of service discovery and was originally dsigned for interactive use. The canonical IT example is finding all printers within a zone, which allows a user to select the desired printer from the returned list. Another example is in the context of UPnP [UPNP], where all media players are returned on a screen and the user can select the desired media player on the screen and play the selected content. In M2M applications, the returned names are not displayed on a screen but an application uses the returned list to select a (set of) Function Set(s) to control. Consequently, names in M2M applications need not be human interpretable (for example, they can be unique numbers).

Create_group is useful in commissioning scenarios, where devices need to be grouped to receive the same command in a possibly synchronous fashion. Groups can also be created to express relations between devices such as ownership. The command creates a group name and creates a list of the members of the group. When the group is a multicast group, the command defines a unique multicast address and port, and specifies the path.

Enroll_member supports network and device reconfiguration. When the physical lay-out of an installation changes because devices are

added, changed or removed, the associated groups also need to be modified.

Remove_member, see motivation under Enroll_member

Return_group is needed to learn the groups of which a device is member. The command is necessary for commissioning purposes where a Commissioning Tool (CT) is used. The CT, on the basis of designs provided by architects, decorators, sound/light engineers, defines groups and group members and stores that information in the service discovery database. In the next phase, the members of a group need to learn their membership from the service discovery to enable reception of messages.

Return_member can be used to learn which devices are member of a given group. This command is useful in connection with Return_group. The device knowing to which groups it belongs can establish communication with the group members. For example, membership of a group instructs new devices, replacing faulty ones, which other devices share access rights or need to be consulted regularly.

3.6. Sleeping devices

This section suggests that service discovery of sleeping devices is mostly a matter of discovering the proxy. It is expected that a proxy will handle communications for the sleeping device. The message sent to the sleeping device is directed to the proxy. The proxy will send the message on with a delay, or send the result of a function on the history of messages, when the sleeping device is ready. The communication protocol between proxy and sleeping device is currently proprietary, but efforts are under way to standardize. The setting up of the proxy is preferably standardized for a large set of proxy types. During the setting-up process, (offline or online) the proxy will take over all the entry-points of the sleeping device. The entry-points of the proxy can be entered into the discovery respository and consequently discovered like any other device.

For groups, two cases need to be considered (1) sleeping device is member of a group and receives group messages, and (2) the sleeping device sends messages to a group. Ad (1), when the sleeping device needs to receive messages sent to a group, the proxy will receive those messages and the Function Set of the proxy is entered as group member to receive the group messages. Ad (2) When the sleeping device sends messages to a group, it is preferable that the sleeping device sends just one multicast message to the group to minimize energy costs. It is required that when one member of the group receives the message, all other group members receive it as well

(unanimity), covered by the "reliability" REQ1 in [I-D.ietf-core-groupcomm]). A simple broadcast over a lowpan will not always succeed and additional multicast algorithms like Trickle [RFC6206] need to be introduced.

4. DNS and RD examples

The following device configuration and environment are assumed for the examples. The devices are placed on floor-x (fx) in two rooms room-y (ry) and room-z (rz). Both rooms contain a powerstrip with a powermeter and four power outlets. In each room there are two luminaires and one presence sensor (PIR). Each luminaire contains a dimmable light and a light sensor. Per floor there is a clock to set day and night time modes of the devices. The domains are: ry.fx.bldg.org, rz.fx.bldg.org and fx.bldg.org. The device names of the 4 luminaires are lm00203, lm00204, lm00205, and lm00206. The device names of the two powerstrips are ps0057, and ps0078. The paths of the Function Sets of the luminaire are: /lamp with resource /lamp/dim for the dimmable light, and /light with resource /light/lumen for the light sensor. The Function Set path for four outlets is /ps with resource /ps/output. The path of each individual outlet is /ps/x with x in {1,2,3,4}, and with resource ps/x/output. The name of the two PIRs is pir. The entry-point path of the PIR is /occup, also being the resource.

Relating location to the domain name is a relevant example of domain naming. Multiple domain names, related to other application aspects, can be specified and applied simultaneously.

Separate subsections provide examples for discovery of devices and of groups. As described in section 3, devices do not announce themselves to the discovery repository, as usual for IT applications, but they are entered (partially) with the aid of a central tool, for example a Remote Control, dedicated device, IPAD or other means.

4.1. DNS-SD examples

4.1.1. Basic Concepts

In conformance with [I-D.cheshire-dnsext-dns-sd], DNS-based discovery uses A or AAAA, PTR, SRV, and TXT Resource Records (RR). The SRV RR [RFC2782] specifies an endpoint. An associated (identically named) TXT RR can contain a URI path. Together the associated SRV and TXT RRs can specify a Function Set. An A or AAAA RR [RFC1035] binds a device name or multicast group name to an IP address. The PTR RR binds a service type to an end-point, or a service subtype to a Function Set.

In cases where the end-point port may be dynamic, e.g. in the IPHC [RFC6282] compressible range, a new 'coap+srv' scheme is proposed (after [I-D.jennings-http-srv]). The authority part of a coap+srv URI specifies the name and location of an SRV record, which in turn contains values for host (IP address) and port.

4.1.2. Commissioning devices

Commissioning is the process to store the relation between a FQDN and a device. It is assumed that either a Remote Control (RC) in the home or a Commissioning Tool (CT) in the professional domain store the relation in DNS.

In the professional domain, the CT is assumed to contain information about the devices as prescibed by architect or installation company. The information in the CT contains device name, device domain name, and location in the building, but the relation with the installed processor, identified with an unique identifier (e.g. EUI-64) is not established. By reading a bar code (or pushing buttons, switching on/off equipment, etc.), the CT learns the identifier of the device to be commissioned. All kinds of techniques can be used to establish the relation between IP address and unique identifier. When the identifier is the EUI-64 value, the IP address of the device can be constructed. When the identifier is not the EUI-64, a proprietary procol can be used to ask a given device its identifier. Etc. etc. The CT can learn the Function Sets (services) available on the device by querying /.well-known/core. In some cases the CT already obtained the Function Sets from a configuration file. Given these data, the CT can enter the devices and its services into DNS. Either automatically, or on instructions of an operator, the CT defines the groups in the DNS.

The home domain is different from the professional domain in the sense that no configuration information exists. The RC can for example use xmDNS to learn the addresses of all the devices present in its site. The RC can query devices for the presence of a given service. The RC can query DNS for its own domain name and use that for the other devices in the site. Once (new) devices are named, this information can be stored in DNS for use in the network.

4.1.3. device examples

The relation between device name and IP address is expressed for the example devices in the following table.

```
lm00203.ry.fx.bldg.org. IN AAAA fdfd::1234
lm00204.ry.fx.bldg.org. IN AAAA fdfd::1235
ps0057.ry.fx.bldg.org.  IN AAAA fdfd::1236
pir.ry.fx.bldg.org.     IN AAAA fdfd::1237
lm00205.rz.fx.bldg.org. IN AAAA fdfd::1238
lm00206.rz.fx.bldg.org. IN AAAA fdfd::1239
ps0058.rz.fx.bldg.org.  IN AAAA fdfd::1240
pir.rz.fx.bldg.org.     IN AAAA fdfd::1241
clock.fx.bldg.org.      IN AAAA fdfd::1242
```

The next part defines the Function Sets related to the device names. The names of the SRV RRs (Function Sets) need to be unique to the DNS server. The names of the Function Sets are valid within the authority zone, bldg.org, of the name server. Consequently, lamp1 is short for lamp1.bldg.org, sensor1 for sensor1.bldg.org, etc. The luminaires with name "lm00xxx" host two Function Sets: a lamp and a sensor.

```
lamp1     IN SRV 0 0 Port    lm00203.ry.fx.bldg.org
          IN TXT path=/lamp
sensor1   IN SRV 0 0 Port    lm00203.ry.fx.bldg.org
          IN TXT path=/light
lamp2     IN SRV 0 0 Port    lm00204.ry.fx.bldg.org
          IN TXT path=/lamp
sensor2   IN SRV 0 0 Port    lm00204.ry.fx.bldg.org
          IN TXT path=/light
powerc1   IN SRV 0 0 Port    ps0057.ry.fx.bldg.org
          IN TXT path=/ps
presence1 IN SRV 0 0 Port    pir.ry.fx.bldg.org
          IN TXT path=/occup
lamp3     IN SRV 0 0 Port    lm00205.rz.fx.bldg.org
          IN TXT path=/lamp
sensor3   IN SRV 0 0 Port    lm00205.rz.fx.bldg.org
          IN TXT path=/light
lamp4     IN SRV 0 0 Port    lm00206.rz.fx.bldg.org
          IN TXT path=/lamp
sensor4   IN SRV 0 0 Port    lm00206.rz.fx.bldg.org
          IN TXT path=/light
powerc2   IN SRV 0 0 Port    ps0058.rz.fx.bldg.org
          IN TXT path=/ps
presence2 IN SRV 0 0 Port    pir.rz.fx.bldg.org
          IN TXT path=/occup
timer     IN SRV 0 0 Port    clock.fx.bldg.org
          IN TXT path=/time
```

The above list of SRV RRs specifies the attributes of the Function Sets: devices, port numbers, IP addresses and path with the accompanying AAAA and TXT records. The names of the SRV records can

be created automatically, as long as they identify the SRV records uniquely within the set of RR entries in the DNS zone. The SRV record with name powercx (x = 1,2) stands for power collection x, accessed via the path /ps which refers to a Function Set that contains a collection of two resources.

Assuming that the service type "_bc" has the service subtype "_lamp", the names of the SRV RRs can also be created from the service subtype prefixed by the EUI-64 value. With the service subtype "_lamp" and EUI-64 value "1234" the SRV name 1234_lamp can be created automatically instead of lamp1.

PTR records enable the service discovery. The names of the PTR records are the names of the service types, defined by IANA, and they refer to the names of the SRV records. To support the query "all lamps within fx.bldg.org", the following PTR records need to be added for service subtype:_lamp._sub._bc._udp.

```
_lamp._sub._bc._udp.bldg.org IN PTR lamp1.bldg.org
_lamp._sub._bc._udp.bldg.org IN PTR lamp2.bldg.org
_lamp._sub._bc._udp.bldg.org IN PTR lamp3.bldg.org
_lamp._sub._bc._udp.bldg.org IN PTR lamp4.bldg.org
```

Equally to query to all services within a domain, PTR records with as name the building control service, "_bc.udp", refer to all SRV records describing Function Sets.

```
_bc._udp.bldg.org IN PTR lamp1.bldg.org
_bc._udp.bldg.org IN PTR lamp2.bldg.org
_bc._udp.bldg.org IN PTR sensor1.bldg.org
_bc._udp.bldg.org IN PTR power1.bldg.org
_bc._udp.bldg.org IN PTR presence1.bldg.org
_bc._udp.bldg.org IN PTR etc, etc.
```

It is shown above how PTR records support the queries filtered on service type. Filtering on domain can be done adding additional PTR records which select the devices of a given type within a given domain. The set of PTR records below filters on all lamps within domain ry.fx.

```
_lamp._sub._bc._udp.ry.fx.bldg.org. IN PTR lamp1.bldg.org
_lamp._sub._bc._udp.ry.fx.bldg.org. IN PTR lamp2.bldg.org
```

4.1.4. Group examples

As an example, five multicast-groups are defined to group all lamps on floor "fx", all lamps in office "ry", all lamps in office "rz", all power-strips on floor "fx", and all devices in the building

controlled by a central timer. The multicast-group names are entered into DNS like the device names to enable resolution from multicast-group name to multicast address.

```
lamp-fx.fx.bldg.org.    IN AAAA ff15::11
lamp-ry.ry.fx.bldg.org. IN AAAA ff15::12
lamp-rz.rz.fx.bldg.org. IN AAAA ff15::13
power-fx.fx.bldg.org.   IN AAAA ff15::14
timer-bldg.bldg.org.    IN AAAA ff15::15
```

It is expected that SDOs will specify naming conventions for group names, extending the service (sub)type names for devices.

The path and port of the multicast-groups is defined with SRV and TXT RRs. (Remark lampgp1 is short for lampgp1.bldg.org, etc.)

```
lampgp1 IN SRV 0 0 Port   lamp-fx.fx.bldg.org.
        IN TXT path=/lamp
lampgp2 IN SRV 0 0 Port   lamp-ry.ry.fx.bldg.org.
        IN TXT path=/lamp
lampgp3 IN SRV 0 0 Port   lamp-rz.rz.fx.bldg.org.
        IN TXT path=/lamp
powergp IN SRV 0 0 Port   power-fx.fx.bldg.org.
        IN TXT path=/ps
timergp IN SRV 0 0 Port   timer-bldg.bldg.org.
        IN TXT path=/tm
```

The groups for the power strips need extra attention because the power strips include a collection of resources. The path to the group can be defined as /ps or as /ps/x with x in {1,2,3,4}. When using /ps/x the group contains the outlet x of the power strips. Using the path /ps as done in the table above refers to all outlets of a powerstrip. When sending a message to the power-fx group with as path /ps/x then the message will be received by Function Sets with path ps/x only.

The members of the groups can be stored in DNS by using the reverse DNS resolution technique. It is not unusual that a given IP address refers to multiple FQDNs. Extrapolating to group names extends the reverse DNS resolution in a natural manner. Below the members of group lamp-fx.fx.bldg.org with IP address ff15::11 containing all four lamps is shown.

```
1.1.0.....0.5.1.f.f.IP6.arpa. IN PTR lm00206.rz.fx.bldg.org.
1.1.0.....0.5.1.f.f.IP6.arpa. IN PTR lm00205.rz.fx.bldg.org.
1.1.0.....0.5.1.f.f.IP6.arpa. IN PTR lm00204.ry.fx.bldg.org.
1.1.0.....0.5.1.f.f.IP6.arpa. IN PTR lm00203.ry.fx.bldg.org.
1.1.0.....0.5.1.f.f.IP6.arpa. IN PTR lamp-fx.fx.bldg.org.
```

With the above table queries like all members of lamp-fx can be answered.

Additional tables are needed to specify the multicast groups to which the Function Set of a device belongs. A PTR RR with the name of the Function Set can refer to the name of the group. In the table below the Function Set "lamp1" is part of the groups "lamp-fx", "lamp-ry" and "timer-bldg":

```
lamp1.bldg.org IN PTR lamp-fx.fx.bldg.org
lamp1.bldg.org IN PTR lamp-ry.ry.fx.bldg.org
lamp1.bldg.org IN PTR timer-bldg.bldg.org
```

Consequently, a device can query DNS for the groups to which its Function Sets belong, and consecutively enable the reception of the associated multicast messages.

4.1.5. Discovery validation

This section describes how the disovery requirements are met with DNS-SD.

4.2. RD examples

Resource discovery in CoAP handles resource paths (called links) for the resources hosted on the server, augmented with attributes of these resources. A well-known path "/.well-known/core" [RFC5785] is a default entry-point for requesting the list of links on a given server [I-D.shelby-core-resource-directory]. The Resource Directory (RD) stores links to resources hosted by other servers. The link-format [I-D.ietf-core-link-format] defines link extensions to specify the service type and service instance as used by DNS-SD. When querying the Resource Directory for links, filters can be applied to return only links with specified attribute values. A node learns the IP-address of the RD by for example sending a multicast request to a predefined multicast address registered with IANA, or by assuming that the RD is located in the edge router.

Contrary to DNS-SD, the RD has not defined a process which permits SDOs to specify service (sub)types. Consequently, the same service type examples are used for DNS-SD as for RD, where service type postfixed with subtype (DNS) equals "resource type" (RD), and

Function Set (SRV) name (DNS) equals "Instance" (RD).

This draft adheres to the mapping between DNS and link-format, described in [I-D.lynn-core-discovery-mapping].

4.2.1. Commissioning devices

It is assumed that either a Remote Control (RC) in the home or a Commissioning Tool (CT) in the professional domain is used to fill in the RD. devices cannot enter links into the RD contrary to the suggestion in [I-D.shelby-core-resource-directory]. Both the RC or the CT have to fill in the RD, because at link creation the RD generates a location of the link-entry (e.g. /ed/453), that is returned to the creator of the link. Consequently, the CT or the RD need to create the link, because they need the location to update the link with additional information.

In the professional domain, the CT learns the identity of the device as described earlier, reads the services of the devices with a GET to /.well-known/core on the device, and stores them into the RD.

In the home domain, the RC can read in the EUI-64 of the device to be entered. The user types in domain names and device names on the RC. Consecutively, the RC follows the same procedure as the CT.

4.2.2. Device examples

For convenience, it is assumed that DNS contains the mapping from FQDN to IP address with AAAA RRs and vice-versa with PTR RRs. In all examples the FQDN is used and not the IP-address. It is assumed that the service type "_bc" has the subtypes: "_lamp", "_sensor", "_strip", "_pir", and "_clock". Registration is done with the following statements by CT or RC to the RD with authority: //rd.example.com/ and path /rd.

```
POST coap://rd.example.com/rd?h="lm00203";d="ry.fx.bldg.org"
Etag: 0x21
Payload:
</lamp>;rt="_bc._lamp";ins="lamp1"
</light>;rt="_bc._sensor";ins="sensor1"
Res: 2.01 created
Location: /rd/1234
```

Leaving out Etag:, Res:, and Location: lines, the other Function Sets are defined with:

```
POST coap://rd.example.com/rd?h="lm00204";d="ry.fx.bldg.org"
Payload:
</lamp>;rt="_bc._lamp";ins="lamp2"
</light>;rt="_bc._sensor";ins="sensor2"

POST coap://rd.example.com/rd?h="lm00205";d="rz.fx.bldg.org"
Payload:
</lamp>;rt="_bc._lamp";ins="lamp3"
</light>;rt="_bc._sensor";ins="sensor3"

POST coap://rd.example.com/rd?h="lm00206";d="rz.fx.bldg.org"
Payload:
</lamp>;rt="_bc._lamp";ins="lamp4"
</light>;rt="_bc._sensor";ins="sensor4"

POST coap://rd.example.com/rd?h="ps0057";d="ry.fx.bldg.org"
Payload:
</ps>;rt="_bc._strip";ins="powerc1"
</ps/1>;rt="_bc._strip";ins="outlet1"
</ps/2>;rt="_bc._strip";ins="outlet2"
</ps/3>;rt="_bc._strip";ins="outlet3"
</ps/4>;rt="_bc._strip";ins="outlet4"

POST coap://rd.example.com/rd?h="ps0058";d="rz.fx.bldg.org"
Payload:
</ps>;rt="_bc._strip";ins="powerc2"
</ps/1>;rt="_bc._strip";ins="outlet5"
</ps/2>;rt="_bc._strip";ins="outlet6"
</ps/3>;rt="_bc._strip";ins="outlet7"
</ps/4>;rt="_bc._strip";ins="outlet8"

POST coap://rd.example.com/rd?h="pir";d="ry.fx.bldg.org"
Payload:
</occup>;rt="_bc._pir";ins="presence1"

POST coap://rd.example.com/rd?h="pir";d="rz.fx.bldg.org"
Payload:
</occup>;rt="_bc._pir";ins="presence2"

POST coap://rd.example.com/rd?h="clock";d="fx.bldg.org"
Payload:
</time>;rt="_bc._clock";ins="timer"
```

The "ins" value identifies the Function Set uniquely within the resources attached to the RD, in the same fashion as done for SRV RRs. It is also possible to postfix the EUI-64 value to the service subtype name, (e.g. ins="pir1241" instead of "presence2").

4.2.3. Group examples

The same five multicast groups of the DNS example are used. The group name to address mapping is specified in DNS with AAAA RRs. In all examples the group name is used and not the IP-address. Registration is done with the following statements by CT or RC to the RD with authority: /rd.example.com/ and path /rd, leaving out Etag:, Res:, and Location: lines.

```
POST coap://rd.example.com/rd?h="lamp-fx";d="fx.bldg.org"
Payload:
</lamp>;rt="_bc._lamp";ins="lampgp1"

POST coap://rd.example.com/rd?h="lamp-ry";d="ry.fx.bldg.org"
Payload:
</lamp>;rt="_bc._lamp";ins="lampgp2"

POST coap://rd.example.com/rd?h="lamp-rz";d="rz.fx.bldg.org"
Payload:
</lamp>;rt="_bc._lamp";ins="lampgp3"

POST coap://rd.example.com/rd?h="power-fx";d="fx.bldg.org"
Payload:
</ps>;rt="_bc._strip";ins="powergp"

POST coap://rd.example.com/rd?h="timer-bldg";d="bldg.org"
Payload:
</time>;rt="_bc._clock";ins="timergp"
```

With the above statements the groups are defined in the RD. However, there is no straight forward mechanism to define the members of a multicast group, or to define the inverse: the groups to which a Function Set belongs.

4.2.4. Discovery validation

This section describes how the disovery requirements are met with the RD.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

TBD

7. Acknowledgments

Zach Shelby wrote the original Discovery section in [I-D.ietf-core-coap] which forms the basis for this draft. This I-D has benefited from conversations with and comments from Emmanuel Frimout, Michael Verschoor, Jamie Mc Cormack, Oscar Garcia, Dee Denteneer, Joop Talstra, Jerald Martocci, Matthieu Vial, Jerome Hamel, George Yianni, and Nicolas Riou.

8. References

8.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, April 2010.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, October 2010.

[RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.

8.2. Informative References

[I-D.cheshire-dnsext-dns-sd]
Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-11 (work in progress), December 2011.

[I-D.eggert-core-congestion-control]
Eggert, L., "Congestion Control for the Constrained Application Protocol (CoAP)", draft-eggert-core-congestion-control-01 (work in progress), January 2011.

[I-D.ietf-6man-uri-zoneid]
Carpenter, B. and R. Hinden, "Representing IPv6 Zone Identifiers in Uniform Resource Identifiers", draft-ietf-6man-uri-zoneid-00 (work in progress), February 2012.

[I-D.ietf-core-coap]
Frank, B., Bormann, C., Hartke, K., and Z. Shelby, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-08 (work in progress), October 2011.

[I-D.ietf-core-groupcomm]
Rahman, A. and E. Dijk, "Group Communication for CoAP", draft-ietf-core-groupcomm-00 (work in progress), January 2012.

[I-D.ietf-core-link-format]
Shelby, Z., "CoRE Link Format", draft-ietf-core-link-format-11 (work in progress),

January 2012.

[I-D.lynn-core-discovery-mapping]
Lynn, K. and Z. Shelby, "CoRE Link-Format to DNS-Based Service Discovery Mapping", draft-lynn-core-discovery-mapping-01 (work in progress), July 2011.

[I-D.lynn-dnsext-site-mdns]
Lynn, K. and D. Sturek, "Extended Multicast DNS", draft-lynn-dnsext-site-mdns-01 (work in progress), March 2011.

[I-D.shelby-core-coap-req]
Shelby, Z., Stuber, M., Sturek, D., Frank, B., and R. Kelsey, "CoAP Requirements and Features", draft-shelby-core-coap-req-02 (work in progress), October 2010.

[I-D.shelby-core-interfaces]
Shelby, Z., "CoRE Interfaces", draft-shelby-core-interfaces-01 (work in progress), January 2012.

[I-D.shelby-core-resource-directory]
Krco, S. and Z. Shelby, "CoRE Resource Directory", draft-shelby-core-resource-directory-02 (work in progress), October 2011.

[I-D.vanderstok-core-bc]
Stok, P. and K. Lynn, "CoAP Utilization for Building Control", draft-vanderstok-core-bc-05 (work in progress), October 2011.

[I-D.jennings-http-srv]
Jennings, C., "DNS SRV Records for HTTP", draft-jennings-http-srv-05 (work in progress), March 2009.

[ZeroConf]
Cheshire, S. and D. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc. , ISBN 0-596-10100-7, 2006.

[UPNP] "Universal Plug and Play", Web http://www.upnp.org, 2012.

Authors' Addresses

Peter van der Stok (editor)
Philips Research
High Tech Campus 34-1
Eindhoven, 5656 AA
The Netherlands

Email: peter.van.der.stok@philips.com

Kerry Lynn
Consultant

Phone: +1-978-460-4253
Email: kerlyn@ieee.org

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen O, 2100
Denmark

Email: Anders_Brandt@sigmadesigns.com