CoRE Link-Format to DNS-Based Service Discovery Mapping
draft-lynn-core-discovery-mapping-02

Abstract

   Resource and service discovery are complimentary. Resource discovery
   provides fine-grained detail about the content of a server, while
   service discovery can provide a scalable method to locate servers in
   large networks. This document defines a method for mapping between
   CoRE Link Format attributes and DNS-Based Service Discovery fields to
   allow either method to be used to locate service interfaces down to
   the granularity of function sets in mixed HTTP/CoAP environments.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on April 25, 2013.

Table of Contents

[1](#). Introduction

The Constrained RESTful Environments (CoRE) working group aims at realizing the REST architecture in a suitable form for the most constrained devices (e.g. 8-bit microcontrollers with limited RAM and ROM) and networks (e.g. 6LoWPAN). CoRE is aimed at machine-to-machine (M2M) applications such as smart energy and building automation. The main deliverable of CoRE is the Constrained Application Protocol (CoAP) specification [I-D.ietf-core-coap].

Automated discovery of resources hosted by a constrained server is critical in machine-to-machine applications where human intervention is minimal and static interfaces result in fragility. CoRE Resource Discovery is intended to support fine-grained discovery of hosted resources, their attributes, and possibly other resource relations [RFC6690].

In contrast, service discovery generally refers to a coarse-grained resolution of an end-point's IP address, port number, and protocol. This definition may be extended to include multi-function devices, where the result of the discovery process may include a path to a resource representing a RESTful service interface and possibly a reference to a description of the interface such as a Web Application Description Language (WADL) document [I-D.vial-core-link-format-wadl].

Resource and service discovery are complimentary in the case of large networks, where the latter can facilitate scaling. This document defines a mapping between CoRE Link Format attributes and DNS-Based Service Discovery [I-D.cheshire-dnsext-dns-sd] fields that permits discovery of CoAP services by either means. It also addresses the following CoRE charter requirement [I-D.shelby-core-coap-req]:

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]

[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 "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

1.2. Resource Discovery

The main function of Resource Discovery is to provide Universal
Resource Identifiers (URIs, also called "links") for the resources
hosted by the server, complemented by attributes about those
resources and perhaps additional link relations. In CoRE this
collection of links and attributes is itself a resource (as opposed
to HTTP headers delivered with a specific resource).

[RFC6690] specifies a link format for use in CoRE Resource Discovery
by extending the HTTP Link Header Format [RFC5988] to describe these
link descriptions. The CoRE Link Format is carried as a payload and
is assigned an Internet media type. A well-known URI "/.well-known/
core" is defined as a default entry-point for requesting the list of
links about resources hosted by a server, and thus performing CoRE
Resource Discovery.

Resource Discovery can be performed either via unicast or multicast.
When a server's IP address is already known, either a priori or
resolved via the Domain Name System (DNS) [RFC1034][RFC1035], unicast
discovery is performed in order to locate a URI for the resource of
interest. This is performed using a GET to /.well-known/core on the
server, which returns a payload in the CoRE Link Format. A client
would then match the appropriate Resource Type, Interface
Description, and possible Content-Type [RFC2045] for its application.
These attributes may also be included in the query string in order to
filter the number of links returned in a response.

1.3. Resource Directories

In many M2M scenarios, direct discovery of resources is not practical
due to sleeping nodes, limited bandwidth, or networks where multicast
traffic is inefficient. These problems can be solved by deploying a
network element called a Resource Directory (RD), which hosts
descriptions of resources held on other servers (referred to as "end-
points") and allows lookups to be performed for those resources. An
end-point is a web server associated with specific IP address and
port; thus a physical device may host one or more end-points. End-
points may also act as clients.

The Resource Directory implements a set of REST interfaces for end-
points to register and maintain sets of Web Links, called resource
directory entries. [I-D.shelby-core-resource-directory] specifies
the web interfaces that an RD supports in order for web servers to
discover the RD and to register, maintain, lookup and remove resource
descriptions; for the RD to validate entries; and for clients to
lookup resources from the RD. Furthermore, new link attributes
useful in conjunction with an RD are defined.

1.4. DNS-Based Service Discovery

   DNS-Based Service Discovery (DNS-SD) defines a conventional method of
   configuring DNS PTR, SRV, and TXT resource records to facilitate
   discovery of services (such as CoAP servers in a subdomain) using the
   existing DNS infrastructure. This section gives a brief overview of
   DNS-SD; see [I-D.cheshire-dnsext-dns-sd] for a detailed
   specification.

   DNS-SD service names are limited to 255 octets and are of the form:

          Service Name = <Instance>.<ServiceType>.<Domain>

   The <Domain> part of the service name is identical to the global (DNS
   subdomain) part of the authority in URIs that identify the resources
   on an individual server or group of servers. <Domain> may identify a
   building zone as shown in the examples of [I-D.vanderstok-core-bc].

   The <ServiceType> part is composed of at least two labels. The first
   label of the pair is the application protocol name [RFC6335] preceded
   by an underscore character. The second label indicates the transport
   and is always "_udp" for CoAP services. In cases where narrowing the
   scope of the search may be useful, these labels may be optionally
   preceded by a subtype name followed by the "_sub" label. An example
   of this more specific <ServiceType> is "lamp._sub._dali._udp". Only
   the rightmost pair of labels is used in SRV and TXT record names.

   The default <Instance> part of the service name may be set at the
   factory or during the commissioning process. It SHOULD uniquely
   identify an instance of <ServiceType> within a <Domain>. Taken
   together, these three elements comprise a unique name for an SRV/ TXT
   record pair within the DNS subdomain.

   The granularity of a service name MAY be that of a host or group, or
   it could represent a particular resource within a CoAP server. The
   SRV record contains the host name (AAAA record name) and port of the
   service while protocol is part of the service name. In the case
   where a service name identifies a particular resource, the path part
   of the URI must be carried in a corresponding TXT record.

   A DNS TXT record is in practice limited to a few hundred octets in
   length, which is indicated in the resource record header in the DNS
   response message [I-D.cheshire-dnsext-dns-sd]. The data consists of
   one or more strings comprising a key=value pair. By convention, the
   first pair is txtver=<number> (to support different versions of a
   service description). An example string is:

                | 0x08 | t | x | t | v | e | r | = | 1 |

2. Mapping CoRE Link Attributes to DNS-SD Records

[I-D.shelby-core-resource-directory] defines two new CoRE Link Format
attributes that are particularly useful in conjunction with RDs:

```
link-extension   = ( "ins" "=" quoted-string ) ; Max 63 octets
link-extension   = ( "exp" )
```

The Resource Instance "ins" attribute is an identifier for a resource
that distinguishes it from other similar resources. This attribute
SHOULD be unique across resources with the same Resource Type
attribute in the domain in which it is used.

The Export "exp" attribute is used as a flag to indicate that a link
description MAY be exported by a resource directory to external
directories. If the "exp" attribute is defined for a link, then the
following CoRE specific target attributes (defined in [RFC6690]) are
mapped directly into DNS-SD labels. The values are subject to format
and length constraints as specified in [I-D.cheshire-dnsext-dns-sd].

2.1. Mapping Resource Instance "ins" attribute into <Instance>

The Resource Instance "ins" attribute maps to the <Instance> part of
a DNS-SD service name. It is stored directly in the DNS as a single
DNS label of canonical precomposed UTF-8 [RFC3629] "Net-Unicode"
(Unicode Normalization Form C) [RFC5198] text. However, to the
extent that the "ins" attribute may be chosen to match the DNS host
name of a service, it SHOULD use the syntax defined in Section 3.5 of
[RFC1034] and Section 2.1 of [RFC1123].

The <Instance> part of the name of a service being offered on the
network SHOULD be configurable by the user setting up the service, so
that he or she may give it an informative name. However, the device
or service SHOULD NOT require the user to configure a name before it
can be used. A sensible choice of default name can allow the device
or service to be accessed in many cases without any manual
configuration at all. The default name should be short and
descriptive, and MAY include a collision-resistent substring such as
the lower bits of the device's MAC address, serial number,
fingerprint, or other identifier in an attempt to make the name
relatively unique.

DNS labels are currently limited to 63 octets in length and the
entire service name may not exceed 255 octets.

2.2. Mapping Resource Type "rt" attribute into <ServiceType>

The resource type "rt" attribute is mapped into the <ServiceType>
part of a DNS-SD service name (as defined in Section 1.4) and SHOULD
conform to the reg-rel-type production of the Link Format defined in
Section 2 of [RFC6690].

The "rt" attribute MUST be composed of at least a single Net-Unicode
text string, without underscore '_' or period '.' and limited to 15
octets in length, which represents the application protocol name.
This string is mapped to the DNS-SD <ServiceType> by prepending an
underscore and appending a period followed by the "_udp" label. For
example, rt="dali" is mapped into "_dali._udp".

The application protocol name may be optionally followed by a period
and a service subtype name consisting of a Net-Unicode text string,
without underscore or period and limited to 63 octets. This string
is mapped to the DNS-SD <ServiceType> by appending a period followed
by the "_sub" label and then appending a period followed by the
service type label pair derived as in the previous paragraph. For
example, rt="dali.light" is mapped into "light._sub._dali._udp".

The resulting string is used to form labels for DNS-SD records which
are stored directly in the DNS.

2.3. <Domain> mapping

[TBD: A method must be specified to determine in which DNS zone the
CoAP service should be registered. See, for example, Section 11 in
[I-D.cheshire-dnsext-dns-sd].]

2.4. TXT Record key=value strings

The resource <URI> is exported to the TXT record key=value string
"path=<URI>".

The Interface Description "if" attribute is exported to the TXT
record key=value string "if=<Interface Description>".

The DNS TXT record can be further populated by importing any other
resource description attributes as they share the same key=value
format specified in Section 6 of [I-D.cheshire-dnsext-dns-sd].

3. Examples

3.1. Importing resource links into DNS-SD

   Assuming the ability to query a Resource Directory or multicast a GET
   (?exp) over the local link, CoAP resource discovery may be used to
   populate the DNS-SD database in an automated fashion. CoAP resource
   descriptions (links) can be exported to DNS-SD for exposure to
   service discovery by using the Resource Instance attribute as the
   basis for a unique service name, composed with the Resource Type as
   the <ServiceType>, and registered in the correct <Domain> (see
   section 2.3). The agent responsible for exporting records to the DNS
   zone file SHOULD be authenticated to the DNS server.

   The following example shows an agent discovering a resource to be
   exported:

```
  Agent                                                            RD
    |                                                              |
    | --- GET /rd-lookup/res?exp ------------------------------->  |
    |                                                              |
    |                                                              |
    | <-- 2.05 Content "<coap://node1/light/27>;exp; ------------ |
    |                   rt="dali.light";ins="FrontSpot"           |
    |                                                              |
```

   Req: GET /rd-lookup/res?exp

   Res: 2.05 Content
   <coap://[FDFD::1234]:61616/light/1>;
     exp;ct=41;rt="dali.light";ins="FrontSpot"

   The agent subsequenly registers the following DNS-SD RRs:

```
node1.example.com.     IN AAAA FDFD::1234
_dali._udp             IN PTR  FrontSpot._dali._udp
light._sub._dali._udp  IN PTR  FrontSpot._dali._udp
FrontSpot._dali._udp   IN SRV  0 0 5678 node1.example.com.
                       IN TXT  txtver=1
                       IN TXT  path=/light/27
```

4. IANA Considerations

This document makes no request of IANA.

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

5. Security Considerations

[TBD]

6. Acknowledgments

Valuable contributions and review comments were made by Anders Brandt, Angelo Castellani, Esko Dijk, and Peter van der Stok.

7. References

7.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

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

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

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

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

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

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S.

                 Cheshire, "Internet Assigned Numbers Authority (IANA)
                 Procedures for the Management of the Service Name and
                 Transport Protocol Port Number Registry", BCP 165,
                 RFC 6335, August 2011.

   [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link
                 Format", RFC 6690, August 2012.

7.2. 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.ietf-core-coap]
                 Shelby, Z., Hartke, K., Bormann, C., and B. Frank,
                 "Constrained Application Protocol (CoAP)",
                 draft-ietf-core-coap-12 (work in progress), October 2012.

   [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-resource-directory]
                 Shelby, Z., Krco, S., and C. Bormann, "CoRE Resource
                 Directory", draft-shelby-core-resource-directory-04 (work
                 in progress), July 2012.

   [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.vial-core-link-format-wadl]
                 Vial, M. and Z. Shelby, "Interface description with WADL
                 in CoRE", draft-vial-core-link-format-wadl-01 (work in
                 progress), September 2011.

   [dns-sd]   "dns-sd service type registration",
                 Web http://www.dns-sd.org/ServiceTypes.html, 2012.

Authors' Addresses

    Kerry Lynn
    Consultant

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

    Zach Shelby
    Sensinode
    Kidekuja 2
    Vuokatti 88600
    FINLAND

    Phone: +358407796297
    Email: zach@sensinode.com