                      CoRE Resource Directory: DNS-SD mapping
                          draft-ietf-core-rd-dns-sd-05

Abstract

   Resource and service discovery are complementary. Resource discovery
   provides fine-grained detail about the content of a web 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 records
   to facilitate the use of either method to locate RESTful service
   interfaces (APIs) in heterogeneous HTTP/CoAP environments.

Status of This Memo

Copyright Notice

carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction and Background

   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 [RFC4944]). 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 [RFC7252].

   CoRE Link Format [RFC6690] is intended to support fine-grained
   discovery of hosted resources, their attributes, and possibly other
   related resources. Automated dynamic discovery of resources hosted

by a constrained server is critical in M2M applications, where human
intervention is minimal and static configurations result in
brittleness.

DNS-Based Service Discovery (DNS-SD) [RFC6763] supports wide-area
search for instances of a given service type (i.e. servers that
support a particular application protocol stack). A service instance
consists of a server's name, IP address, and port number plus
additional meta-data about the server. This data may extend to
support multi-function devices, where multiple services are available
at the same endpoint. The result of the discovery process may
include a path to a resource representing the entry point to each
function's RESTful service interface and possibly a link to a formal
description of that interface (e.g. a JSON Hyper-Schema document
[I-D.handrews-json-schema-hyperschema]).

Resource and service discovery are complementary 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 records that permits discovery of CoAP services by
either method. It also addresses the CoRE charter goal to
interoperate with DNS-SD.

The primary use case for mapping between resource and service
discovery is to support heterogeneous HTTP/CoAP environments where,
for example, HTTP clients may discover and communicate with CoAP
servers that are behind a "cross proxy" [RFC8075].

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in
[RFC2119]. The term "byte" is used in its now conventional sense as
a synonym for "octet".

This specification requires readers to be familiar with all the terms
and concepts that are discussed in [RFC6690] and [RFC8288]. Readers
should also be familiar with the terms and concepts discussed in
[RFC7252].

This specification also incorporates the terminology of
[I-D.ietf-core-resource-directory].

In particular, the following terms are used frequently:

Endpoint: a web server associated with a specific IP address and
port; thus a physical device may host one or more endpoints.
Endpoints may also act as clients.

Link: Web Linking [RFC8288] defines a Web Link (link) as a typed
connection between two resources, comprised of:

o a link context,

o a link relation type (see Section 2.1 of [RFC8288],

o a link target, and

o optionally, target attributes (see Section 2.2 of [RFC8288]).

A link can be viewed as a statement of the form "link context has a
link relation type resource at link target, which (optionally) has
target attributes", where link target and context are typically
Universal Resource Identifiers (URIs) [RFC3986]. For example,
"https://www.example.com/" has a "canonical" resource at
"https://example.com", which has a "type" of "text/html".

1.2. CoRE Resource Discovery

The main function of Resource Discovery is to return links to the
resources hosted by a server, complemented by attributes about those
resources and additional link relations. In CoRE this collection of
links and attributes is itself a resource (in contrast to HTTP, where
headers delivered with a specific resource describe its attributes).

Resource Discovery can be performed either 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 the entry point to 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 [RFC6690]. A client would then match the appropriate Resource
Type, Interface Description, and possible media 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.

Multicast Resource Discovery is useful when a client needs to locate
a resource within a limited scope, and that scope supports IP
multicast. A GET request to the appropriate multicast address is
made for "/.well-known/core". In order to limit the number and size
of responses, a query string is recommended with the known
attributes. Typically, a resource would be discovered based on its

   Resource Type and/or Interface Description, along with possible
   application-specific attributes.

1.3. CoRE 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 that originate on other endpoints and
   allows indirect lookups to be performed for those resources.

   The Resource Directory implements a set of REST interfaces for
   endpoints to register and maintain collections of links, called
   Resource Directory registrations. [I-D.ietf-core-resource-directory]
   specifies the web interfaces that an RD supports for endpoints 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.

1.4. DNS-Based Service Discovery

   DNS-Based Service Discovery (DNS-SD) defines a conventional method of
   naming and 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; for a detailed specification see
   [RFC6763].

   DNS-SD Service Names are limited to 255 bytes and are of the form:

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

   The Service Name identifies a SRV/TXT Resource Record (RR) pair. The
   SRV RR specifies the hostname and port of an endpoint. The TXT RR
   provides additional information in the form of key/value pairs. DNS-
   Based Service Discovery is accomplished by sending a DNS request for
   PTR records with the name <ServiceType>.<Domain>, which will return a
   list of zero or more Service Names.

   The <Domain> part of the Service Name is identical to the global (DNS
   subdomain) part of the authority in URIs [RFC3986] that identify the
   resources on an individual server or group of servers.

   The <ServiceType> part is generally composed of two labels. The
   first label of the pair is the application protocol name [RFC6335]
   preceded by an underscore character. For example, an organization
   such as the Open Connectivity Foundation [OCF] that specifies

Resource Types [RFC6335] might register application protocol names
beginning with "oic", which all servers that advertise OCF resources
would use as part of their ServiceType. The second label indicates
the transport protocol binding and is typically "_udp" for CoAP
services.

The default <Instance> part of the Service Name SHOULD be set to a
default value at the factory and MAY be modified during the
commissioning process. It MUST 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 might represent a particular resource within a CoAP server. The
SRV record contains the host name (AAAA record name) and port of the
endpoint, 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 bytes in
length, which is indicated in the resource record header in the DNS
response message (See section 6 of [RFC6763]). The data consist 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). Each string is formatted as a single length
byte followed by 0-255 bytes of text. An example string is:

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

2. New Link-Format Attributes

When using the CoRE Link Format to describe resources being
discovered by or posted to a resource directory service, additional
information about those resources is often useful. This
specification defines the following new attributes for use in the
CoRE Link Format [RFC6690] to enable the data-driven mappings
described in Section 3:

```
    link-extension    = ( "exp" )
    link-extension    = ( "ins" "=" (ptoken | quoted-string) )
                        ; The token or string is max 63 bytes
    link-extension    = ( "st" "=" (ptoken | quoted-string) )
                        ; The token or string is max 15 bytes
```

2.1. Export attribute "exp"

The Export "exp" attribute is used as a flag to indicate that a link description MAY be exported from a resource directory to external directories.

The CoRE Link Format is used for many purposes between CoAP endpoints. Some are useful mainly locally; for example checking the observability of a resource before accessing it, determining the size of a resource, or traversing dynamic resource structures. However, other links are very useful to be exported to other directories, for example the entry point resource to a functional service. This attribute MAY be used as a query parameter in the RD Lookup Function Set defined in Section 6 of [I-D.ietf-core-resource-directory].

2.2. Resource Instance attribute "ins="

The Resource Instance "ins=" attribute is an identifier for this resource, which makes it possible to distinguish it from other similar resources in a Resource Directory. This attribute specifies the value to be used for the <Instance> portion of an exported DNS-SD Service Name (see Section 1.4), and SHOULD be unique across resources with the same Resource Type "rt=" attribute in the domain in which it is used.

A Resource Instance SHOULD be a descriptive human readable string like "Ceiling Light, Room 3". This attribute MUST NOT be more than 63 bytes in length. The resource identifier attribute MUST NOT appear more than once in a link description. This attribute MAY be used as a query parameter in the RD Lookup Function Set defined in Section 7 of [I-D.ietf-core-resource-directory].

2.3. Service Type attribute "st="

The Service Type instance "st=" attribute specifies the value to be used for the <ServiceType> portion of an exported DNS-SD Service Name (see Section 1.4). This attribute MUST NOT be more than 15 bytes in length (see [RFC6335], Section 5.1) and MUST be present in the IANA Service Name registry [st].

3. Mapping CoRE Link Attributes to DNS-SD Record Fields

3.1. Mapping Resource Instance attribute "ins=" to <Instance>

The Resource Instance "ins=" attribute maps directly 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, if the "ins=" attribute is 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 (see Appendix D of [RFC6763]).

   DNS labels are limited to 63 bytes in length and the entire Service
   Name may not exceed 255 bytes.

3.2. Mapping Service Type attribute "st=" to <ServiceType>

   The Service Type "st=" attribute maps directly to the <ServiceType>
   part of a DNS-SD Service Name.

   In practice, the ServiceType should unambiguously identify
   interoperable devices. It is up to individual SDOs to specify how to
   represent their registered Resource Type "rt=" values as registered
   application protocol names according to [RFC6335]. The application
   name is then used as the value of the resource "st=" attribute.

   The resulting application protocol name MUST be composed of at least
   a single Net-Unicode text string, without underscore '_' or period
   '.' and limited to 15 bytes in length (see Section 5.1 of [RFC6335]).
   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="oic.d.light" might correspond to the registered
   application protocol name st="oic-d-light" and would be mapped into
   Service Type "_oic-d-light._udp".

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

3.3. <Domain> Mapping

   TBD: A method must be specified to determine which DNS zone the CoAP
   service description should be exported to. See, for example,
   Section 11 in [RFC6763] and Section 2 in
   [I-D.sctl-service-registration].

3.4. TXT Record key=value strings

   DNS-SD key/value pairs may be derived from CoRE Link Format
   information and exported as key=value strings in a DNS-SD TXT record
   (See Section 6.3 of [RFC6763]).

   The resource <URI> is exported as key/value pair "path=<URI>".

   The Interface Description "if=" attribute is exported as key/value
   pair "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 [RFC6763].

3.5. Exporting 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 Service Type
   attribute as the <ServiceType>, and registered in the appropriate
   <Domain>. The agent responsible for exporting records to the DNS
   zone file SHOULD be authenticated to the DNS server. The following
   example, using the example lookup location /rd-lookup, shows an agent
   discovering a resource to be exported:

    Req: GET /rd-lookup/res?exp

    Res: 2.05 Content
   <coap://[FDFD::1234]:5683/light/1>;
      exp;st=oic-d-light;rt="oic.d.light";ins="Spot";
              d="sector";ep="node1"

   The agent subsequently registers the following DNS-SD RRs, assuming a
   derived DNS zone name "office.example.com":

   _oic-d-light._udp.office.example.com   IN PTR
           Spot._oic-d-light._udp.office.example.com
   Spot._oic-d-light._udp.office.example.com IN TXT
           txtver=1;path=/light/1;rt=oic.d.light;d=sector
   Spot._oic-d-light._udp.office.example.com IN SRV
           0 0 5683 node1.office.example.com.
   node1.office.example.com.   IN AAAA FDFD::1234

4. Exporting Resource Directory Service to DNS

   In some cases it is required that one (or more) Resource Directories
   (RD) in a given DNS domain can be discoverable from DNS. The /.well-
   known/core resource of the RD should reflect this by specifying the
   "ins", "exp", and the "st" attributes in the the link of the RD
   service. This document specifies in Section 5 two servicetypes: _rd-
   lookup-res._udp and _rd-lookup-ep._udp for resource types rt =
   core.rd-lookup-res and rt = core.rd-lookup-ep respectively. The
   default coap and coaps ports are respectively: 5683 and 5684.

   The value of the instance MAY be specified by the manager of the
   resource directories. In case of an unmanaged RD (for example in a
   home network) it is recommended that the ins parameter takes a value
   provided by an Authorization Server during the acceptance of the RD
   to the network (see for example section 7 of
   [I-D.ietf-core-resource-directory]).

   With the assumption that the "ins" value is attributed by
   Authorization Server, and [FDFD::1234] is IP address of RD, Example
   links for RD are:

        Req: GET coap://[FDFD::1234]/.well-known/core?exp

        Res: 2.05 Content
        <rd-lookup/res>;
          exp;st=rd-lookup-res;rt="core.rd-lookup-res";
          ins="505567",
        <rd-lookup/ep>;
          exp;st=rd-lookup-ep;rt="core.rd-lookup-ep";
          ins="505572"

   The link atributes can be exported to RR by the mapping process
   described in Section 3.

5. IANA considerations

   Two registries are affected by this document: (1) "RD Parameters"
   registry under "Core Parameters" registry, and (2) Service Name and
   Transport Protocol Port Number Registry

5.1. RD Parameters Registry

   This specification defines new parameters for the registry "RD
   Parameters" provided under "CoRE Parameters" (TBD).

Full name	Short	Validity	Use	Description
ServiceType	st		RLA	Name of the Service Type, max 63 bytes
Resource Instance	ins		RLA	Instance identifier of the resource
Export	exp		RLA	flag to indicate exportation

5.2. Service Name and Transport Protocol Port Number Registry

This specification defines new parameters for the Service Name and Transport Protocol Port Number Registry:

* _rd-lookup-res._udp at ports 5683 and 5684
* _rd-lookup-ep._udp at ports 5683 and 5684

6. Security considerations

Malicious nodes can export fake link attributes to DNS. It is recommended that the RD can be authenticated, and is authorized to both join the network and export its link attributes. Authentication is specified in [I-D.ietf-ace-oauth-authz].

7. Contributors

Keryy lynn was the initiator of, and major contributor to this document. This document was split out from [I-D.ietf-core-resource-directory]. Zach Shelby was a co-author of the original version of this draft.

8. Acknowledgments

The authors wish to thank Stuart Cheshire, Ted Lemon, and David Thaler for their thorough reviews and clarifying suggestions.

9. References

9.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
              November 1987, <https://www.rfc-editor.org/info/rfc1035>.

   [RFC1123] Braden, R., Ed., "Requirements for Internet Hosts -
              Application and Support", STD 3, RFC 1123,
              DOI 10.17487/RFC1123, October 1989,
              <https://www.rfc-editor.org/info/rfc1123>.

   [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail
              Extensions (MIME) Part One: Format of Internet Message
              Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996,
              <https://www.rfc-editor.org/info/rfc2045>.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO
              10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November
              2003, <https://www.rfc-editor.org/info/rfc3629>.

   [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
              Resource Identifier (URI): Generic Syntax", STD 66,
              RFC 3986, DOI 10.17487/RFC3986, January 2005,
              <https://www.rfc-editor.org/info/rfc3986>.

   [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler,
              "Transmission of IPv6 Packets over IEEE 802.15.4
              Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007,
              <https://www.rfc-editor.org/info/rfc4944>.

   [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network
              Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008,
              <https://www.rfc-editor.org/info/rfc5198>.

   [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, DOI 10.17487/RFC6335, August 2011,
              <https://www.rfc-editor.org/info/rfc6335>.

   [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link
              Format", RFC 6690, DOI 10.17487/RFC6690, August 2012,
              <https://www.rfc-editor.org/info/rfc6690>.

   [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service
              Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013,
              <https://www.rfc-editor.org/info/rfc6763>.

   [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained
              Application Protocol (CoAP)", RFC 7252,
              DOI 10.17487/RFC7252, June 2014,
              <https://www.rfc-editor.org/info/rfc7252>.

   [RFC8075] Castellani, A., Loreto, S., Rahman, A., Fossati, T., and
              E. Dijk, "Guidelines for Mapping Implementations: HTTP to
              the Constrained Application Protocol (CoAP)", RFC 8075,
              DOI 10.17487/RFC8075, February 2017,
              <https://www.rfc-editor.org/info/rfc8075>.

   [RFC8288] Nottingham, M., "Web Linking", RFC 8288,
              DOI 10.17487/RFC8288, October 2017,
              <https://www.rfc-editor.org/info/rfc8288>.

9.2. Informative References

   [I-D.handrews-json-schema-hyperschema]
              Andrews, H. and A. Wright, "JSON Hyper-Schema: A
              Vocabulary for Hypermedia Annotation of JSON", draft-
              handrews-json-schema-hyperschema-01 (work in progress),
              January 2018.

   [I-D.ietf-ace-oauth-authz]
              Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and
              H. Tschofenig, "Authentication and Authorization for
              Constrained Environments (ACE) using the OAuth 2.0
              Framework (ACE-OAuth)", draft-ietf-ace-oauth-authz-24
              (work in progress), March 2019.

   [I-D.ietf-core-resource-directory]
              Shelby, Z., Koster, M., Bormann, C., Stok, P., and C.
              Amsuess, "CoRE Resource Directory", draft-ietf-core-
              resource-directory-22 (work in progress), July 2019.

   [I-D.sctl-service-registration]
              Cheshire, S. and T. Lemon, "Service Registration Protocol
              for DNS-Based Service Discovery", draft-sctl-service-
              registration-02 (work in progress), July 2018.

   [OCF]   Foundation, O., "OCF Specification 2.0", 2018,
              <https://openconnectivity.org/developer/specifications>.

   [REST]   Fielding, R., "Architectural Styles and the Design of
               Network-based Software Architectures", 2000,
               <http://www.ics.uci.edu/~fielding/pubs/dissertation/
               fielding_dissertation.pdf>.

   [rt]   IANA, ., "Resource Type (rt=) Link Target Attribute
               Values", 2012, <https://www.iana.org/assignments/core-
               parameters/
               core-parameters.xhtml#rt-link-target-att-value>.

   [st]   IANA, ., "Service Name and Transport Protocol Port Number
               Registry", 2018, <https://www.iana.org/assignments/
               service-names-port-numbers/
               service-names-port-numbers.xml>.

Authors' Addresses

   Peter van der Stok
   Consultant

   Phone: +31 492474673 (Netherlands), +33 966015248 (France)
   Email: consultancy@vanderstok.org
   URI: www.vanderstok.org

   Michael Koster
   SmartThings
   665 Clyde Avenue
   Mountain View, CA 94043
   USA

   Phone: +1 707-502-5136
   Email: Michael.Koster@smartthings.com

   Christian Amsuess
   Energy Harvesting Solutions
   Hollandstr. 12/4
   1020
   Austria

   Phone: +43 664-9790639
   Email: c.amsuess@energyharvesting.at