Discovery Mapping

CoRE Link Format <-> DNS-SD RRs

draft-lynn-core-discovery-mapping

Kerry Lynn
Zach Shelby
27 July 2011

Use Cases

- Support alternate methods of discovery in mixed environments (e.g. CoAP and HTTP clients)
- Support hierarchical discovery in large environments (e.g. 1000's of points)
  - DNS-SD for coarse-grained discovery
  - Link format for fine-grained discovery
- Rough figure of merit: bandwidth required for discovery should decrease or remain constant as network is scaled and technology is added

DNS-Based Service Discovery

- A conventional use of existing DNS RRs and messages to support service discovery:

DNS Resource Record	Binding
A, AAAA	Host name to IP address
PTR	{ServiceType} to Service instance name
SRV	Service instance name to host, port (end-point)
TXT	Arbitrary key=value pairs (path=entry-point)

- Expand the definition of *service* to include entries to *RESTful interfaces* (e.g. multi-function devices)
- Service names are of the form:

  {Instance}.{ServiceType}.{Domain}

Link-format to DNS-SD mapping

Link Format	DNS-SD
Resource Instance (ins=)	{Instance}
Resource Type (rt=)	{Service}
<uri>	TXT path=
Interface Description (if=)	TXT if=
Attribute (xxx=)	TXT xxx=

Things decided by the mapping entity:

- Domain name (the DNS server where the records are created)
- Host name (if it doesn't exist already)
- txtver=*n* (TXT record version)

Link Format -> DNS-SD Example

CoRE query:

REQ: GET coap://[ff02::1]/.well-known/core?exp

RES: 2.05 "Content" (from [fdfd::1234]:5678)
</sensors/temp/1>;exp;ct=41;rt="tempC.zigbee";ins="indoorTemp";
if="sensor",

Resulting RRs:

node1234.example.com. IN AAAA fdfd::1234

_zigbee._udp IN PTR indoorTemp._zigbee._udp

_tempC._sub._zigbee._udp IN PTR indoorTemp._zigbee._udp

indoorTemp. _zigbee._udp IN SRV 0 0 5678 node1234.example.com.
IN TXT txtver=1
IN TXT path=/sensors/temp/1
IN TXT if=sensor