CoRE
Internet-Draft
Intended status: Experimental
Expires: September 8, 2011

A. Brandt
Sigma Designs

March 7, 2011

Discovery of CoAP servers across subnets
draft-brandt-coap-subnet-discovery-00

Abstract

The document describes the process of discovering CoAP servers distributed in multiple subnets in a non-specified topology. CoAP Discovery Gateways are used to discover one subnet from another. CoAP Discovery Gateways may provide caching to enable discovery of sleeping nodes in LLN environments. The solution scales to large installations since discovery is handled by the client in an incremental fashion.

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 September 8, 2011.

Copyright Notice

Table of Contents

1. Introduction

This document describes the process of discovering CoAP servers distributed in multiple subnets in a non-specified topology. CoAP (Constrained object Application Protocol) Discovery Gateways are used to discover one subnet from another. CoAP Discovery Gateways may provide caching to enable discovery of sleeping nodes in Low-power and Lossy Network (LLN) environments. Caching CoAP Discovery Gateways may also have the purpose of preserving bandwidth in an LLN environment. No synchronization of databases is required. The solution scales to large installations since discovery is handled by the client in an incremental fashion.

CoAP servers may be running on a variety of physical layers; each implementing a subnet. A lamp module may be running over IEEE 802.15.4. A movement sensor may be running over Z-Wave.

The document presents requirements to a home control discovery solution and proposes a solution based on the CoAP link format [I-D.ietf-core-link-format].

CoAP Subnet Discovery Must support IPv6. An implementation MAY implement support for both IPv4 and IPv6 .

1.1. Requirements Language

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].

2. Requirements for Discovery services for very constrained nodes

The term constrained node covers a range of nodes that are constrained with regards to memory size, power consumption, packet size or CPU capabilities. This document covers nodes for

applications within home control and building control. A Low-power Lossy Network (LLN) is used for communication.

The basic functionality of devices for home control and building control is the same: measure temperature, detect movement, dim light, operate locks, etc. The discovery, management and operation of networks is however somewhat different. This document addresses discovery requirements specific to the most constrained devices and outlines how proper solutions may address home control and building control technologies in the same way.

2.1. Home Control network evolution - scenarios

Home control systems may be constructed from consumer products acquired in a retail store. The products may come from multiple vendors. The user may have no knowledge of network management. The user may have no existing local network. If there is a local network, it may have no DHCP or DNS infrastructure. Devices must always work; the consumer price level does not leave much margin for support hotlines.

The following scenarios assume one common application protocol. Multi-protocol support may be achieved via application gateways; potentially with CoAP as the common language. This is out of scope of this document.

2.1.1. Scenario A: Retail home control starter kit

This scenario outlines how LLN nodes may be used in the most simple network configurations and how such simple networks may grow from there. In such simple networks the 6LoWPAN border router is reduced to a conceptual function. The remote control acts as a coordinating master node on the link layer as well as a 6LoWPAN border router. The remote control enters a sleep state when it is not in use.

1. Starter kit bring-up

   A user starts with an RF remote control and three RF plug-in modules. The remote control assigns unique link-layer addresses and ULA IP addresses [RFC4193] to modules during network

bootstrapping so that the remote control and the plug-in modules are in the same IP network. ULA IP addresses are needed as multi-hop routing may be used inside the LLN network. Prefixes and header compression CIDs have infinite lifetime as there is no listening border router.

Remote control buttons are mapped to (groups of) IP addresses of the plug-in modules. The setup mode may be activated via a special key sequence in the remote control.

2. Adding sensors

The user adds another plug-in module and a movement sensor to the network. The remote control is still used as a coordinating master node. The IP address of the target module is stored in the movement sensor. When the sensor detects a movement, it sends an "ON" command to the plug-in module.

3. Adding home control to the smart phone

The user adds a border router to interface the LLN network to the LAN (and WiFi) of the house.
The user connects a smartphone to the WiFi network. A home control smartphone app performs home control resource discovery. A list of LLN nodes allows the user to configure smartphone widgets for scene control of lamps; e.g. "Watch TV" or "Doing the dishes".

2.1.2. Scenario B: Service provider home control offering

In this scenario, a consumer receives a pre-bundled kit from a service provider:

- A combined internet access router and LLN border router with WiFi support
- Three plug-in modules for installation in the home
- A personal web profile on the service provider web portal

- o A smartphone app for control via WiFi

  Remote access credentials, LLN prefix and other parameters are preconfigured by the service provider.

1. Setting up the kit

   The LLN border router manages link-layer node properties as well as prefix assignment, etc. A web page is used to add the three plug-in modules to the LLN. The smartphone app controls the plug-in modules via WiFi. Routable addresses allow the app to reach the modules from the WiFi subnet.

2. Adding other technologies

   The original starter kit was a wireless LLN. The user connects a power-line LLN border router to the LAN, thus forming a backbone network for the two LLN border routers. The user includes power-line based plug-in modules with the new power-line LLN border router.
   Each individual border router manages local ULA prefixes and header compression CIDs.

3. Re-discovering the network

   The smartphone app rediscovers home control resources in both

   LLNs and the backbone network. The lists of home control resources allow the user to configure smartphone widgets for scene control of lamps in both subnets.
   The border routers runs a routing protocol on the backbone to allow IP packets to traverse from one subnet into the other without any manual configuration of routers.

2.2. Discovery

Discovery serves multiple purposes in a home control network:

1. When the home control network has grown from the original starter kit to a substantial number of nodes, the owner may get a desire for centralized management of the network. The management tool needs a way to request information from each node in the network.

Typically, nodes will carry no visible identification at this time. If possible, non-functioning nodes should also be identified. Once nodes are identified, the user may locate the nodes physically and assign naming and location information, e.g. "bedroom, ceiling light" or "living room, drapes".

2. When setting up a remote control, the user may want to browse all drape controllers in the entire network - both in the wireless LLN and in the powerline LLN. The wireless drape controllers may be battery powered and sleeping most of the time. The powerline drape controllers may be in a dedicated powerline LLN subnet behind several border routers. It is a challenge to discover nodes in other subnets. A multicast-based discovery protocol like mDNS cannot have its messages forwarded by routers since it uses link-local addressing. Even if mDNS messages could be forwarded, some LLNs featuring multi-hop routing do only support multicast in a very slow and inefficient fashion. Assuming multicast messages could be distributed over a LLN, sleeping nodes would not be able to detect discovery requests.

A solution is required which

- Does not flood the LLN with unnecessary discovery traffic
- Does not require multicasting in LLNs
- Does allow the discovery of sleeping nodes

2.3. Zero-Configuration

One major property of home control networks is the absence of a professional installer. When making consumer products, one cannot make any assumptions about the qualifications of the user. Simple

operation is a requirement. As an example, a user may associate a light module by activating a setup sequence on a wall switch and then pushing a button on the light module. The user never sees any data. Most users actually do not care about the IP address of the light module.

No border router, DHCP server or DNS server is present in a starter kit network. Modules that were initially purchased and configured

with a remote control may one day be used in a far more advanced installation with global routable prefixes and hierarchical DNS naming.

2.4. Multiple unique routable subnets

Home control domains may be composed of devices communicating via one or more border routers, e.g power-line to wireless, optionally via a backbone network. A wireless home control LLN may in itself contain routers to do multihop forwarding within the home control LLN. The two subnets may host different MAC/PHYs as well as different routing protocols. The user can extend the home control starter kit with another subnet without having to re-install the original subnet. The construction of unique local subnet prefixes is described in[RFC4193].

3. Building blocks of a CoAP Discovery infrastructure

CoAP defines a protocol for exchange of requests and commands between constrained nodes. Already described in [I-D.ietf-core-coap], the CoAP Server is a host capable of responding to CoAP messages. CoAP Servers may be classified into the following sub-types:

- Stand-alone CoAP Server
- CoAP Discovery Gateway
- Caching CoAP Discovery Gateway

These are described in the following sections. Finally, the CoAP Discovery Client is described.

CoAP Discovery Gateways MAY advertise legacy devices along with native CoAP servers. CoAP Discovery Gateways MAY provide access to legacy services via CoAP request and control messages. Discovery of diverse resource types enables a migration path from legacy technologies towards an all-CoAP infrastructure.

3.1. Stand-alone CoAP Server

A stand-alone CoAP Server does not provide access to other CoAP Servers; physical or logical. Typically a stand-alone CoAP Server is able to perform some action, e.g. measuring a temperature or turning on light.
A CoAP Server reports periodically to the Discovery Gateway via the 6LoWPAN ND address registration process, e.g. once every hour. Battery operated CoAP servers may run out of battery. Light modules may become defect. The reporting allows the Discovery gateway to monitor the availability of CoAP Servers.

3.2. CoAP Discovery Gateway

A CoAP Discovery Gateway is a CoAP enabled router interconnecting different subnets, e.g. a LLN Border Router. The subnets may host stand-alone CoAP Servers as well as other CoAP Discovery Gateways. Each CoAP Discovery Gateway interface MUST respond to the CoAP Discovery request "GET /.well-known/core?n=DiscoveryGateway". When queried, the gateway MUST report all other interfaces maintained by the discovery gateway.

The Discovery Gateway MAY maintain a list of CoAP Servers that recently stopped sending address registrations. How a CoAP Discovery Gateway is to advertize such CoAP Servers is TBD.

3.3. Caching CoAP Discovery Gateway

A Caching CoAP Discovery Gateway performs caching of discovery information on behalf of other nodes in a given subnet. A discovery gateway interface MUST respond to the CoAP Discovery request "GET /.well-known/core?n=DiscoveryGateway". When queried, the discovery gateway MUST report all other interfaces maintained by the discovery gateway. The gateway MUST indicate if respective interfaces are of the type "CoAP Discovery Gateway" or "Caching CoAP Discovery Gateway". This information MAY be be ignored by a discovery client.

CoAP Servers reporting to a Caching CoAP Discovery Gateway MAY respond to CoAP Discovery requests. A Caching CoAP Discovery Gateway MUST intercept discovery requests and respond on behalf of CoAP Servers. This allows sleeping nodes to be discovered, saves LLN bandwidth and allows the Caching CoAP Discovery Gateway to maintain connectivity state information like "online/offline/unstable" per CoAP server.

The Caching CoAP Discovery Gateway MAY maintain a list of CoAP Servers that recently stopped sending address registrations. How a CoAP Discovery Gateway is to advertize such CoAP Servers is TBD.

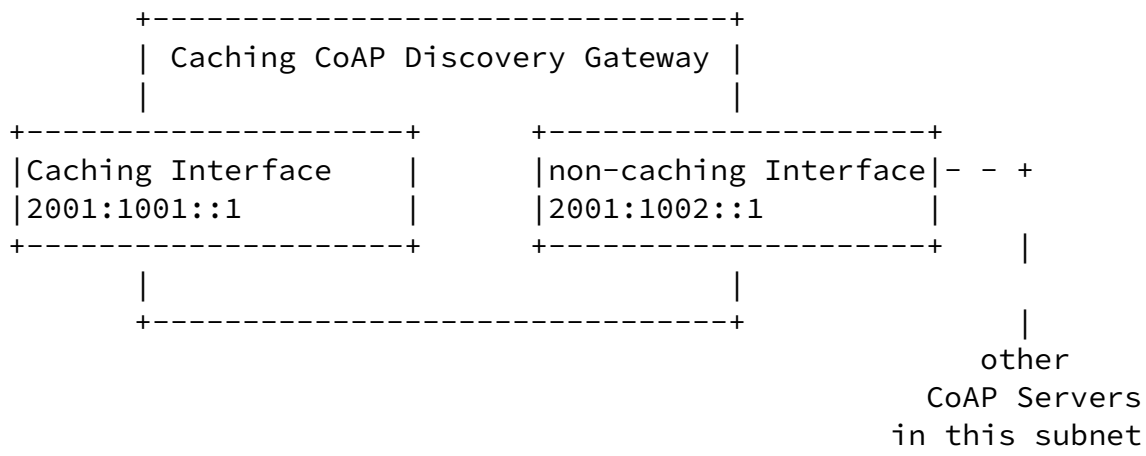


Figure 1: Caching CoAP Discovery Gateway

A Caching CoAP Discovery Gateway MAY report that it is a (non-caching) CoAP Discovery Gateway on some interfaces; refer to Section 3.2 .

The device type "Caching CoAP Discovery Gateway" only indicates that discovery information is cached. Caching of real-time data from CoAP servers is out of scope of this document.

A Caching CoAP Discovery Gateway SHOULD NOT interfere with any other traffic than Discovery Requests for Discovery Gateways or the capabilities of CoAP Servers which report to the CoAP Discovery Gateway, e.g. "Device type = Thermostat". A Caching CoAP Discovery Gateway MUST NOT cache any changing parameters.

3.4. CoAP Discovery Client

A CoAP Discovery Client uses the CoAP Discovery request "GET /.well-known/core?n=DiscoveryGateway" to initiate a discovery session. The target for the discovery request depends on the properties of the actual network. Two cases apply:

1. Client is in a LLN domain with no multicast support

   The initial request is sent to the Authoritative Border Router of that LLN.

2. Client is in a link-local subnet domain with multicast support (like Ethernet)

The initial request is transmitted to the link-local "all-routers" multicast address.

If discovery requests cause CoAP Discovery Gateways to announce other CoAP Discovery Gateways in other subnets, additional discovery requests are directed to those CoAP Discovery Gateways.

A Discovery Client may run in remote controls, smart phone apps or central management systems for home automation or building control.

The discovery process depends on the presence of a CoAP discovery gateway in the subnet of the discovery client. Since CoAP subnet discovery uses normal CoAP messages, link-local discovery works "out of the box" in link-local enabled environments. Refer to [I-D.ietf-core-link-format].

4. CoAP Discovery

CoAP Discovery is a hierarchical process and involves two phases: CoAP Topology Discovery and CoAP Server Discovery.

The purpose of the Topology Discovery phase is to establish a snapshot of the available CoAP Discovery Gateways.

CoAP messages are used to discover CoAP Discovery Gateways in a hierarchical fashion. Having completed the topology discovery phase, a CoAP client may initiate discovery of particular CoAP server resources, e.g. light dimmers, or a more general wildcard discovery may be done by the client; building a complete database. The same request MUST be sent to each discovered CoAP Discovery Gateway interface in a sequential fashion.

The information may be presented, e.g. in lists of different device types. CoAP subnet discovery enables access to end nodes in multiple subnets without any manual configuration of routers. The topology discovery process may return information on Caching CoAP Discovery Gateways. Caching CoAP Discovery Gateways allow the discovery of sleeping and defective nodes but require that CoAP clients implement 6lowPAN-ND address registration [I-D.ietf-6lowpan-nd]with the Authoritative Border Router. Management and naming related issues of CoAP servers in building control are discussed in

[I-D.vanderstok-core-bc].

CoAP Discovery depends on the ability to traverse subnets. Thus, all CoAP Servers MUST have routable IPv6 addresses; either in global prefixes or according to ULA principles. The border router is typically the physical device implementing the CoAP Discovery Gateway function in a LLN. This specification collapses the address of the default gateway (border router) and the discovery gateway in order to limit the number of IP addresses that LLN nodes have to manage - and

to avoid having to distribute the address of the CoAP Discovery Gateway in LLNs. RAs already convey the IP address of the default gateway.

4.1. Topology Discovery

A client may be anywhere in the topology when initiating the Topology Discovery. Any topology may be traversed (if allowed by firewall policies in border routers).

The discovery process allows a client to discover CoAP servers according to the classification described in Section 3.

4.1.1. Topology Path String definitions

A CoAP Discovery Gateway or caching CoAP Discovery Gateway MUST support the following CoAP messages:

- GET /.well-known/core?n=DiscoveryGateway
- GET /.well-known/core/topology
- GET /.well-known/core/topology/device
- GET /.well-known/core/topology/interfaces
- GET /.well-known/core/topology/servers

4.1.1.1. /topology/device

A CoAP Discovery Gateway MUST report one of the device types listed in Figure 2.

Device type	Label	Interpretation
1	Discovery Gateway	This is a CoAP Discovery Gateway interface
2	Discovery Gateway, caching	This CoAP Discovery Gateway interface is caching

Figure 2: CoAP Discovery Device Types

The report formats are described in the following sections.

4.1.1.2. /topology/interfaces

A CoAP Discovery Gateway MUST return the structure

```
[interface address_1]
...
[interface address_n]
```

in response to a query for /topology/interfaces.

The processing of a discovery request depends on the receiving interface:

If the request is targeting the gateway interface that physically received the request, the response contains all subnet interfaces of the discovery gateway.

If the request is targeting another gateway interface than the gateway interface that physically received the request, the response contains all discovery gateways known to the subnet of the targeted interface.

4.1.1.3. /topology/servers

A Caching CoAP Discovery Gateway MUST return the structure

[server address_1]
...
[server address_n]

in response to a query for /topology/servers.

If the request is targeting the gateway interface that physically received the request, the response contains the identity of known CoAP Servers that report to this Caching CoAP Discovery Gateway interface.

If the request is targeting another gateway interface than the gateway interface that physically received the request, a (non-caching) CoAP Discovery Gateway interface in a subnet supporting multicast MUST issue a multicast request for all CoAP Servers in response to a query for /topology/servers. The individual CoAP Server responses are returned directly to the requesting discovery Client.

4.1.2. ?n=DiscoveryGateway

A CoAP Discovery Gateway MUST report the path /topology in response to ?n=DiscoveryGateway. A CoAP Discovery Gateway MAY report other paths as well.

4.1.3. Discovering Gateway interfaces - an example

The query string ?n=DiscoveryGateway is used for discovering topology information in a CoAP enabled infrastructure.

```
[Client sends multicast request to WiFi subnet]
| CoAP GET /.well-known/core?n=DiscoveryGateway

[LLN Border Router responds]
| 200 OK
|
| core://2001:.../topology/device;ct=0;n="DiscoveryGateway"
```

```
[Client sends unicast request to the responding CoAP Discovery Gateway]
[(LLN border router)]
| CoAP GET /.well-known/core/topology/interfaces

[LLN Border Router responds]
| 200 OK
|
| 2001:0db8:85a3:beef:0000:8a2e:0370:7334
| 2001:0db8:85a3:babe:0000:8a2e:0370:4321
```

It is seen that the initial Discovery Gateway request only returns a single string: "/topology/device/...". Since the path "/topology/interfaces" is mandatory for CoAP Discovery Gateways, the client may request this structure as soon as it has detected the CoAP Discovery Gateway.

4.2. Initial Topology Discovery

A Topology Discovery operation is initiated by a CoAP client with the CoAP message GET /.well-known/core?n=DiscoveryGateway in one of the following ways.

1. If a client resides in a multicast enabled environment (like Ethernet or WiFi) the client issues a multicast message (as described in [I-D.ietf-core-link-format]) to the "all nodes" address. All on-link CoAP Discovery Gateways MUST respond to the GET message by returning a list of other interfaces of the respective CoAP Discovery Gateways. In order to avoid collisions, the responding CoAP Discovery Gateways MUST insert a 0..MAX_RA_DELAY_TIME [I-D.ietf-6lowpan-nd] random delay before responding.

2. If a discovery client resides in a LLN environment (like IEEE 802.15.4 or Z-Wave) the client issues a unicast message to the border router of the LLN. The default border router of the LLN MUST respond to a CoAP Discovery request by returning a list of other interfaces of that particular CoAP Discovery Gateway.

The discovery client builds a list of reported subnets that it has to discover. Duplicates MUST be omitted.

4.3. Incremental Topology Discovery

The client holds a list of reported discovery gateway subnets that it has to discover; either from the Initial Topology Discovery or from a previous Topology Discovery.

For each subnet interface, the client sends a unicast GET /.well-known/core?n=DiscoveryGateway message to the interface. In its default configuration, a CoAP Discovery Gateway MUST return the address of each remote interface. A CoAP Discovery Gateway MAY be configured to return URIs for identification. CoAP Discovery MUST support zero-configuration environments like home control where no DNS server can be assumed.

4.3.1. Multicast domain behind routers

If a discovery client initiates discovery from a LLN environment, it may reach a backbone router interface residing in a multicast enabled network domain such as Ethernet. When a CoAP Discovery Gateway receives a unicast discovery request for a multicast enabled network interface via another discovery gateway interface, that CoAP Discovery Gateway interface MUST forward the discovery request in a multicast message for the "all nodes" multicast address.

The discovery client MUST ignore a reported Discovery Gateway interface if that interface is already in the list of known Discovery Gateway interfaces. This is to prevent loops.

A discovery client MAY perform hierarchical discovery by using the general /.well-known/core path. This combines the topology and server discovery phases. The downside is that a client may receive large amounts of data for each individual discovery message. This may be a problem for memory constrained nodes. By discovering the gateway topology first and using filtered server discovery, a client may achieve significant reductions in received data.

4.4. CoAP Server Discovery

CoAP Server Discovery builds on network information revealed during topology discovery. Each discovered subnet must be discovered

individually. As an example, if a client connected to a backbone has discovered two LLNs behind two border routers, the client must perform CoAP Server discovery in the backbone (on-link subnet) as well as each of the two LLN interfaces.

5. Examples

5.1. Discovery across CoAP Discovery Gateways

Consider the following network environment: The client is located in LAN2. The discovery process has to traverse CoAP Discovery Gateway GW4 and LAN1 to locate CoAP Discovery Gateways GW1, GW2 and GW3. CoAP Discovery Gateways 1, 2 and 3 also have to be traversed. When no more new CoAP Discovery Gateways are discovered, discovery for CoAP Servers can be initiated.

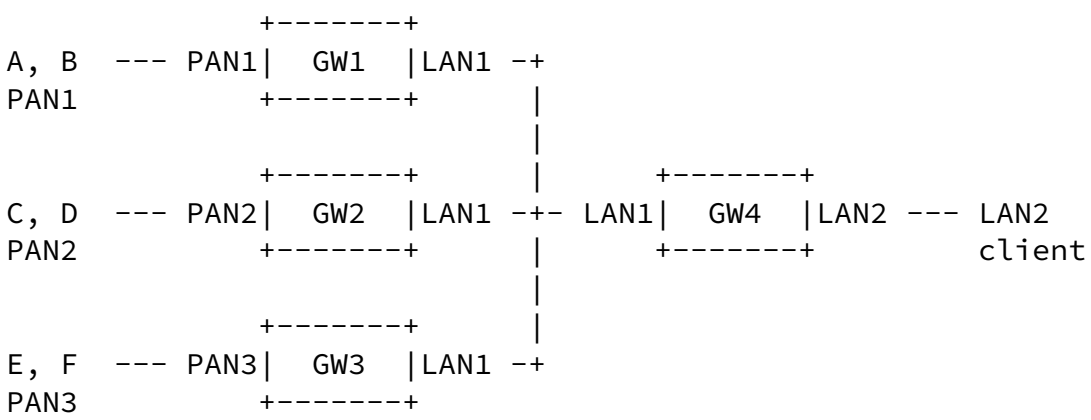


Figure 3: Discovery across CoAP Discovery Gateways

Figure 4: CoAP Discovery Process

1. The client sends out a multicast "GET /.well-known/core?n=DiscoveryGateway"

2. GW4 reports its other interfaces (GW4LAN1).

3. The client sends "GET /.well-known/core?n=DiscoveryGateway" to GW4LAN1

4. GW4LAN1 is not caching and received request in unicast => "GET /.well-known/core?n=DiscoveryGateway" is forwarded as multicast

5. Reports received from GW1LAN1, GW2LAN1 and GW3LAN1 within 2 seconds.

6. The client sends "GET /.well-known/core?n=DiscoveryGateway" to GW1PAN1

7. GW1PAN1 reports that no other gateways were found in PAN1

8. The client sends "GET /.well-known/core?n=DiscoveryGateway" to GW2PAN2

9. GW1PAN1 reports that no other gateways were found in PAN2

10. The client sends "GET /.well-known/core?n=DiscoveryGateway" to GW3PAN3

11. GW1PAN1 reports that no other gateways were found in PAN3

The client now has the following list of CoAP Discovery Gateway interfaces in unique subnets:

1. (mcast in on-link subnet)

2. GW4LAN1

3. GW1PAN1

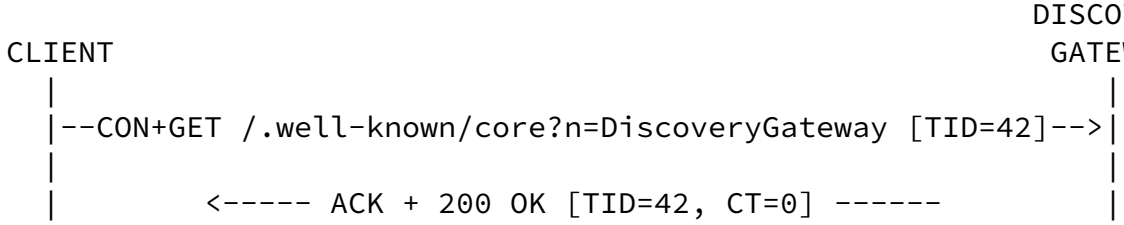
4. GW2PAN2

5. GW3PAN3

The client may now issue searches for other CoAP Servers by sending the request "GET /.well-known/core" to each CoAP Discovery Gateway in the list.

5.2. /topology path formats

Figure 5 shows an example of a client sending a request for /.well-known/core/topology?n=DiscoveryGateway. The discovery gateway sends back a response with the matching resources in the payload.



```
Payload:
<core://.../topology/device>;sh="/??";ct=0;n="DiscoveryGateway"
```

Figure 5: Looking for CoAP Discovery Gateways

Figure 6 shows an example of a client sending a request for /.well-known/core/topology/interfaces. The discovery gateway sends back a response with the actual interfaces provided by the CoAP Discovery

Gateway. A CoAP Discovery Gateway MUST implement the path /topology/interfaces.

```
                                                          DISCOVERY
CLIENT                                                     GATEWAY
  |                                                           |
  |--CON+GET /.well-known/core/topology/interfaces[TID=44]-->|
  |                                                           |
  |        <----- ACK + 200 OK [TID=44, CT=0] ------          |
Payload:
2001:0db8:85a3:beef:0000:8a2e:0370:7334
2001:0db8:85a3:babe:0000:8a2e:0370:4321
```

Figure 6: Using the "/topology/interfaces" path

Figure 7 shows an example of a client sending a request for /.well-known/core/topology/interfaces to a caching CoAP Discovery Gateway. The discovery gateway sends back a response with the actual interfaces provided by the CoAP Discovery Gateway.

Subsequently, the client may sending a request for /.well-known/core/topology/servers to get a list CoAP servers known by the Caching CoAP Discovery Gateway. This list includes sleeping and FLN nodes.

```
                                                          DISCOVERY
CLIENT                                                     GATEWAY
  |                                                           |
  |--CON+GET /.well-known/core/topology/interfaces[TID=45]-->|
  |                                                           |
  |        <----- ACK + 200 OK [TID=45, CT=0] ------          |
Payload:
2001:0db8:85a3:beef:0000:8a2e:0370:7334
```

Figure 7: Receiving addresses from a caching CoAP Discovery Gateway

Figure 8 shows an example of a client sending a request for /.well-known/core/topology/servers to a caching CoAP Discovery Gateway. The discovery gateway sends back a response with the CoAP Servers known by the Caching CoAP Discovery Gateway.

In this case the Caching CoAP Discovery Gateway reports legacy devices which do not necessarily speak CoAP. The client may need to implement multiprotocol support in order to communicate to the devices.

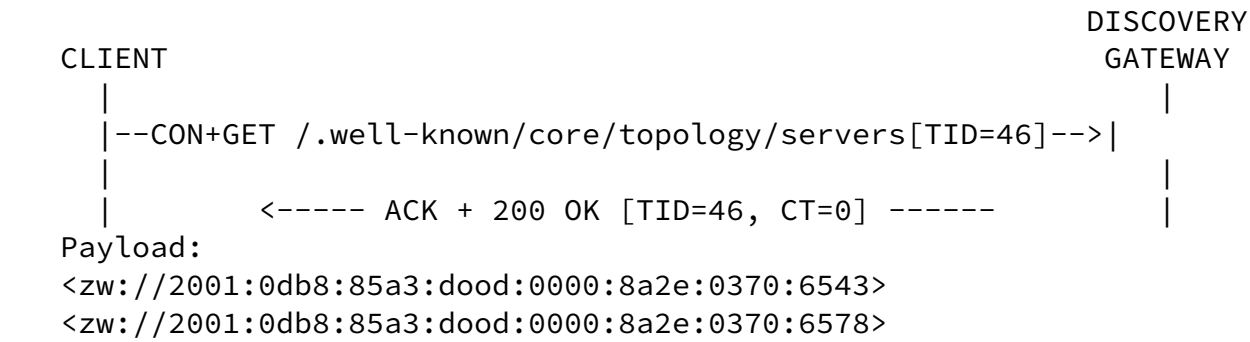


Figure 8: Advertising legacy devices via CoAP Discovery

A more advanced CoAP application gateway may provide translation between legacy protocols and CoAP. This is out of scope of this document.

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

If a CoAP Discovery Gateway receives a generalized CoAP GET /.well-known/core message and that interface resides in a multicast enabled

environment such as Ethernet, the CoAP Discovery Gateway forwards that CoAP request in an "all nodes" multicast message. In response, the CoAP Discovery Gateway potentially receives messages from a number of CoAP Discovery Gateways connected to that link. Forwarding all responses back to the requesting client in individual messages MAY be used for an amplification attack.

Coap discovery is not intended for Internet-wide operation. An internet access router SHOULD NOT forward CoAP messages to or from the Internet domain unless there is a specific application need for doing so. CoAP Discovery depends on a secure perimeter. So does many of the LLN nodes which this discovery mechanism is targeting.

Triggering an amplification attack requires that an attacker has access to the LAN or has control over LLN nodes. If the LLN implements link-layer security, an attacker cannot simply inject a wireless packet. If, however, one is within radio range of a LLN, a modified microwave oven may be a more efficient jamming tool than an amplification attack.

8. References

8.1. Normative References

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

[RFC4193] "Unique Local IPv6 Unicast Addresses", October 2005.

8.2. Informative References

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-04 (work in progress), January 2011.

[I-D.ietf-core-link-format]
Shelby, Z., "CoRE Link Format", draft-ietf-core-link-format-02 (work in progress), December 2010.

[I-D.vanderstok-core-bc]
Stok, P. and K. Lynn, "CoAP Utilization for Building Control", draft-vanderstok-core-bc-02 (work in progress), October 2010.

[I-D.ietf-6lowpan-nd]
Shelby, Z., "Neighbor Discovery Optimization for Low-power and Lossy Networks".

[RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, January 1997.

Appendix A. Open Issues

A.1. Large reports

The CoAP Block transfer mode MUST be implemented in order to support large reports, e.g. from a caching CoAP Discovery Gateway responding on behalf of 100's of CoAP Servers in an LLN.

A.2. M2M Filtering

The document describes the use of ?n=DiscoveryGateway for detecting CoAP Discovery Gateways.

It should be considered if dedicated codes or keywords should be assigned. M2M applications will benefit from shorter codes and avoid the ambiguity of the free text allowed for '?n='.

A.3. Routable subnets

CoAP Discovery requires that all CoAP subnets are all reachable from a given subnet. Some application spaces, e.g. DIY home control, may be set up as off-the-shelf boxes with auto-assigned IPv6 subnets [RFC4193] with no route entries to other subnets.

RIPng [RFC2080] is considered sufficient for a home control application space. Larger installations for building control and the like are expected to be managed networks.

The following requirements could ensure successful plug-and-play

behavior when combining Border Routers with CoAP Discovery Gateways from different vendors:

- A CoAP Discovery Gateway MUST support RIPng
- RIPng SHOULD be enabled in all CoAP Discovery Gateways
- A CoAP Discovery Gateway MAY implement other routing protocols

A.4. Compressing responses from caching CoAP Discovery Gateways

A caching CoAP Discovery Gateway SHOULD omit leading bytes of each reported address if the addresses are all in the same subnet served by the CoAP Discovery Gateway. If omitting leading bytes, a responding CoAP Discovery Gateway MUST provide the prefix information that was omitted from the reported addresses.

If no prefix is specified the interface identifiers MUST be full IP addresses or URIs.

A Caching CoAP Discovery Gateway MAY return more than one response message. If compression is used all addresses in a response message MUST belong to the same subnet prefix.

A.5. Integrated Battery support

A caching CoAP Discovery Gateway MAY be integrated into a LLN border router. This allows for tight integration of support services for sleeping nodes.

The Caching CoAP Discovery Server allows sleeping nodes to be discovered. The border router may implement mailbox delivery services for sleeping nodes. The border router may return "Destination Responding Slowly" ICMP messages to IP hosts sending to a sleeping node. The purpose of the ICMP message is to prevent IP applications from resending messages because they are not receiving

application acks.

A distributed routing protocol MAY distribute the mailbox services. This is out of scope of this specification.

Appendix B. Acknowledgements

Special thanks to Klaus Hartke, Peter Bigot, Peter van der Stok, Kerry Lynn and Zach Shelby for substantial contributions to the ideas and text in the document.

Author's Address

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen O DK-2100
DENMARK

Phone: +4529609501
Email: abr@sdesigns.dk