CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: January 15, 2014

A. Rahman, Ed.
InterDigital Communications, LLC
E. Dijk, Ed.
Philips Research
July 14, 2013

Group Communication for CoAP
draft-ietf-core-groupcomm-11

Abstract

CoAP is a RESTful transfer protocol for constrained devices and constrained networks. It is anticipated that constrained devices will often naturally operate in groups (e.g., in a building automation scenario all lights in a given room may need to be switched on/off as a group). This document provides guidance for how the CoAP protocol should be used in a group communication context. An approach for using CoAP on top of IP multicast is detailed. Also, various use cases and corresponding protocol flows are provided to illustrate important concepts. Finally, guidance is provided for deployment in various network topologies.

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 January 15, 2014.

Table of Contents

1. Conventions and Terminology 3
2. Introduction 4
2.1. Background 4
2.2. Scope 4
3. Protocol Considerations 5
3.1. IP Multicast Background 5
3.2. Group Definition and Naming 5
3.3. Port and URI Configuration 6
3.4. Group Methods 7
3.5. Group Member Discovery 8
3.6. Configuring Group Membership in Endpoints 8
3.7. Multicast Request Acceptance and Response Suppression . . 10
3.8. Congestion Control 12
3.9. Proxy Operation 13
3.10. Exceptions 15
4. Use Cases and Corresponding Protocol Flows 15
4.1. Introduction 15
4.2. Network Configuration 15
4.3. Discovery of Resource Directory 17
4.4. Lighting Control 18
4.5. Lighting Control in MLD Enabled Network 21
4.6. Commissioning the Network Based On Resource Directory . . 22
5. Deployment Guidelines 23
5.1. Target Network Topologies 23
5.2. Advertising Membership of Multicast Groups 24
5.2.1. Using the MLD Listener Protocol 24
5.2.2. Using the RPL Routing Protocol 24
5.2.3. Using the MPL Forwarding Protocol 25
5.3. 6LoWPAN Specific Guidelines 25
6. Security Considerations 26
6.1. Security Configuration 26
6.2. Threats 26
6.3. Threat Mitigation 26
6.3.1. WiFi Scenario 26
6.3.2. 6LoWPAN Scenario 27
6.3.3. Future Evolution 27
7. IANA Considerations 27
7.1. New 'core.gp' Resource Type 27
7.2. New 'coap-group+json' Internet Media Type 28

8. Acknowledgements 29
9. References 29
9.1. Normative References 29
9.2. Informative References 30
Appendix A. Multicast Listener Discovery (MLD) 31
Appendix B. Change Log 32
Authors' Addresses 38

1. Conventions and 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 above key words are used to establish a set of guidelines for CoAP group communication. An implementation of CoAP group communication MAY implement these guidelines; an implementation claiming compliance to this document MUST implement the set of guidelines.

This document assumes readers are familiar with the terms and concepts that are used in [I-D.ietf-core-coap]. In addition, this document defines the following terminology:

Group Communication
: A source node sends a single message which is delivered to multiple destination nodes, where all destinations are identified to belong to a specific group. The source node itself may be part of the group. The underlying mechanism for group communication is assumed to be multicast based. The network involved may be a constrained network such as a low-power, lossy network.

Multicast
: Sending a message to multiple destination nodes with one network invocation. There are various options to implement multicast including layer 2 (Media Access Control) and layer 3 (IP) mechanisms.

IP Multicast
: A specific multicast solution based on the use of IP multicast addresses as defined in "IANA Guidelines for IPv4 Multicast Address Assignments" [RFC5771] and "IP Version 6 Addressing Architecture" [RFC4291].

Low power and Lossy Network (LLN)
: A type of constrained IP network where devices are interconnected by a variety of low-power and lossy links (such as IEEE 802.15.4,

Bluetooth Low Energy (BLE), Digital Enhanced Cordless Telecommunication (DECT)) or lossy links (such as IEEE P1901.2 power-line communication).

2. Introduction

2.1. Background

Constrained Application Protocol (CoAP) is a Representational State Transfer (REST) based approach for resource constrained devices operating in an IP network [I-D.ietf-core-coap]. CoAP has many similarities to HTTP [RFC2616] but also has some key differences. Constrained devices can be large in number, but are often highly correlated to each other (e.g., by type or location). For example, all the light switches in a building may belong to one group and all the thermostats may belong to another group. Groups may be pre-configured before deployment or dynamically formed during operation. If information needs to be sent to or received from a group of devices, group communication mechanisms can improve efficiency and latency of communication and reduce bandwidth requirements for a given application. HTTP does not support any equivalent functionality to CoAP group communication.

2.2. Scope

Group communication involves a one-to-many relationship between CoAP endpoints. Specifically, a single CoAP client will simultaneously get (or set) resource representations from multiple CoAP servers using CoAP over IP multicast. An example would be a CoAP light switch turning on/off multiple lights in a room with a single CoAP group communication PUT request, and handling the potential multitude of (unicast) responses.

The normative protocol aspects of running CoAP on top of IP Multicast and processing the responses are given in [I-D.ietf-core-coap]. The main contribution of this document lies in providing additional guidance for several important group communication features. Among the topics covered are group definition, group resource manipulation, and group configuration. Also, proxy operation and minimizing network congestion for group communication is discussed. Finally, specific use cases and deployment guidelines are for CoAP group communication outlined.

3. Protocol Considerations

3.1. IP Multicast Background

IP Multicast protocols have been evolving for decades, resulting in proposed standards such as Protocol Independent Multicast - Sparse Mode (PIM-SM) [RFC4601]. Yet, due to various technical and business reasons, IP Multicast is not widely deployed on the general Internet. However, IP Multicast is very popular in specific deployments such as in enterprise networks (e.g., for video conferencing), smart home networks (e.g., Universal Plug and Play (UPnP)) and carrier IPTV deployments. The packet economy and minimal host complexity of IP multicast make it attractive for group communication in constrained environments.

To achieve IP multicast beyond a subnet, an IP multicast routing or forwarding protocol needs to be active on IP routers. An example of a routing protocol specifically for LLNs is the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) (Section 12 of [RFC6550]) and an example of a forwarding protocol for LLNs is Multicast Protocol for Low power and Lossy Networks (MPL) [I-D.ietf-roll-trickle-mcast]. Finally, PIM-SM [RFC4601] is often used for multicast routing in un-constrained networks.

IP multicast can also be run in a Link-Local (LL) scope. This means that there is no routing involved and an IP multicast message is only received over the link on which it was sent.

For a complete IP multicast solution, in addition to a routing/ forwarding protocol, a so-called "listener" protocol is needed for the devices to subscribe to groups (see Section 5.2).

3.2. Group Definition and Naming

A group is defined as a set of CoAP endpoints, where each endpoint is configured to receive multicast CoAP requests that are sent to the group's associated IP multicast address. An endpoint MAY be a member of multiple groups. Group membership of an endpoint MAY dynamically change over time.

To initiate CoAP group communication, a Group URI is used as the request URI in a CoAP request. A Group URI has the scheme 'coap' and includes in the authority part either a group IP multicast address or a hostname (e.g., Group Fully Qualified Domain Name (FQDN)) that can be resolved to the group IP multicast address. A Group URI also contains an optional CoAP port number in the authority part. Group URIs follow the CoAP URI syntax [I-D.ietf-core-coap].

It is recommended, for sending nodes, to use the IP multicast address literal in a Group URI. In case a Group hostname is used, it can be uniquely mapped to a site-local or global IP multicast address via DNS resolution (if supported). Some examples of hierarchical Group FQDN naming (and scoping) for a building control application are shown below ([I-D.vanderstok-core-dna]):

URI authority	Targeted group of nodes
all.bldg6.example.com	"all nodes in building 6"
all.west.bldg6.example.com	"all nodes in west wing, building 6"
all.floor1.west.bldg6.example.com	"all nodes in floor 1, west wing, building 6"
all.bu036.floor1.west.bldg6.example.com	"all nodes in office bu036, floor1, west wing, building 6"

Similarly, if supported, reverse mapping (from IP multicast address to Group FQDN) is possible using the reverse DNS resolution technique ([I-D.vanderstok-core-dna]).

3.3. Port and URI Configuration

A CoAP server that is a member of a group listens for CoAP messages on the group's IP multicast address, on a specified UDP port. The default UDP port is the CoAP default port 5683 but a non-default UDP port MAY be specified for the group; in which case implementers MUST ensure that all group members are configured to use this same port.

Multicast based group communication will not work if there is diversity in the authority port (e.g., different dynamic port addresses across the group) or if the resources are located at different paths on different endpoints. Therefore, some measures must be present to ensure uniformity in port number and resource names/locations within a group. All CoAP multicast requests MUST be sent using a port number according to one of below options:

1. A pre-configured port number. The pre-configuration mechanism MUST ensure that the same port number is pre-configured across all endpoints in a group and across all CoAP clients performing the group requests.

2. If the client is configured to use service discovery including port discovery, it uses a port number obtained via a service discovery lookup operation for the targeted CoAP multicast group.

3. Use the default CoAP UDP port (5683).

All CoAP multicast requests SHOULD operate on URI paths ("links") only in one or more of the following ways:

1. Pre-configured URI paths, if available. The pre-configuration mechanism SHOULD ensure that these paths are pre-configured across all CoAP servers in a group and all CoAP clients performing the group requests.

2. If the client is configured to use default CoRE resource discovery, it uses URI paths retrieved from a "/.well-known/core" lookup on a group member. The URI paths the client will use MUST be known to be available also in all other endpoints in the group. The URI path configuration mechanism on servers MUST ensure that these URIs (identified as being supported by the group) are configured on all group endpoints.

3. If the client is configured to use another form of service discovery, it uses URI paths from an equivalent service discovery lookup which returns the resources supported by all group members.

4. If the client has received a Group URI through a previous RESTful interaction with a trusted server, for the purpose of the client using this URI in a request, it can use this URI in a multicast request. For example, a commissioning tool may instruct a sensor device in this way to which target (multicast URI) it should report sensor events.

3.4. Group Methods

Idempotent methods (i.e., CoAP GET, PUT, and DELETE) SHOULD be used for group communication, with one exception as follows. A non-idempotent method (i.e., CoAP POST) MAY be used for group communication if the resource being POSTed to has been designed to cope with the lossy nature of multicast. Note that not all group members are guaranteed to receive the multicast request, and the sender cannot readily find out which group members did not receive it.

All CoAP messages that are sent via multicast MUST be Non-Confirmable. A unicast response per server MAY be sent back to answer the group communication request (e.g., response "2.05 Content" to a group GET request) taking into account the congestion control rules defined in Section 3.8. The unicast responses received may be a mixture of success (e.g., 2.05 Content) and failure (e.g., 4.04 Not Found) codes depending on the individual server processing results.

3.5. Group Member Discovery

CoAP defines a resource discovery capability [RFC6690], but does not specify how to discover groups (e.g., find a group to join or send a multicast message to) or how to discover members of a group (e.g., to address selected group members by unicast). A simple ad-hoc method to discover members of a CoAP group would be to send a multicast "CoAP ping" [I-D.ietf-core-coap]. The collected responses to the ping would then give an indication of the group members.

3.6. Configuring Group Membership in Endpoints

The group membership of a CoAP endpoint may be configured in one of the following ways. First, the group membership may be pre-configured before node deployment. Second, a node may be programmed to discover (query) its group membership during operation using a specific service discovery means. Third, it may be configured during operation by another node (e.g., a commissioning device).

In the first case, the pre-configured group information may be either directly a IP multicast address, or a hostname (FQDN) which is during operation resolved to a IP multicast address by the endpoint using DNS (if supported).

For the second case, a CoAP endpoint may look up its group membership using techniques such as DNS-SD and Resource Directory [I-D.ietf-core-resource-directory]. The latter is detailed more in Section 4.6.

In the third case, typical in scenarios such as building control, a commissioning tool determines to which group a sensor or actuator node belongs, and writes this information to the node, which can subsequently join the correct IP multicast group on its network interface. The information written may again be an IP multicast address or a hostname.

To achieve better interoperability between endpoints from different manufacturers, an OPTIONAL RESTful interface for configuring CoAP endpoints with relevant group information is described here. This interface provides a solution for the third case mentioned above. To access this interface a client MUST use unicast methods (GET/PUT/POST /DELETE) only as it is a method of configuring group information in individual endpoints. Using multicast operations in this situation may lead to unexpected (possibly circular) behavior in the network.

CoAP endpoints implementing this optional mechanism MUST support the group configuration Internet Media Type "application/coap-group+json" (Section 7.2). A resource offering this representation can be

annotated for direct discovery [RFC6690] using the resource type (rt) "core.gp" where "gp" is shorthand for "group" (Section 7.1). An authorized controller uses this media type to query/manage group membership of a CoAP endpoint as defined below.

The group configuration resource has a JSON-based content format (as indicated by the media type). A (unicast) GET on a CoAP endpoint with a resource with this format returns a JSON array of group objects, each group object being a JSON object. Below example shows a client requesting group membership to a CoAP server, where the response is in the "application/coap-group+json" content format containing a single group object:

```
Req: GET /gp
Res: 2.05 Content (Content-Format: application/coap-group+json)
[ { "n": "Room-A-Lights.floor1.west.bldg6.example.com",
    "ip": "ff15::4200:f7fe:ed37:14ca" }
]
```

In a response, the OPTIONAL "n" key/value pair stands for "name" and identifies the group with a hostname, for example a FQDN. The REQUIRED "ip" key/value pair specifies the IP multicast address of the group. Its value can be empty if unknown at the time of generating the response.

Note that each group object in the JSON array represents a single IP multicast group for the endpoint. If there are multiple elements in the array then the endpoint is a member of multiple IP multicast groups.

When the content format is used in a request, the "ip" key/value are OPTIONAL to define the group's associated IP multicast address. The "n" key/value are also OPTIONAL then. If the "ip" key and its value are given, this takes priority. The "n" key/value are just informational in this case. If only the "n" key/value are given, the CoAP endpoint has to do DNS resolution (if supported) to obtain the IP multicast address from the hostname. At least one of the "n" or "ip" key/value MUST be given in a group object in a request.

A (unicast) POST with a group configuration media type as payload instructs the CoAP endpoint to join the defined group(s). The endpoint adds the specified IP multicast address(es) to its network interface configuration. The endpoint also updates the resource by adding the specified group object(s) to the existing ones:

```
Req: POST /gp (Content-Format: application/coap-group+json)
[ { "n": "floor1.west.bldg6.example.com",
```

```
  "ip": "ff15::4200:f7fe:ed37:14cb" } ]
Res: 2.04 Changed
```

A (unicast) PUT with a group configuration media type as payload will replace all current group memberships in the endpoint with the new ones defined in the PUT request. A (unicast) DELETE with a group configuration media type will delete all group memberships from the endpoint.

After any change on a Group configuration resource, the endpoint MUST effect registration/de-registration from the corresponding IP multicast group(s) as soon as possible. Finally, any (unicast) operation to change a CoAP endpoint group membership configuration (i.e., PUT/POST/DELETE) SHOULD use DTLS-secured CoAP [I-D.ietf-core-coap]. Thus only authorized controllers should be allowed by an endpoint to configure its group membership.

3.7. Multicast Request Acceptance and Response Suppression

CoAP [I-D.ietf-core-coap] and CoRE Link Format [RFC6690] define normative behaviors for:

1. Multicast request acceptance - in which cases a coAP request is accepted and executed, and when not.

2. Multicast response suppression - in which cases the CoAP response to an already-executed request is returned to the requesting endpoint, and when not.

Note that a CoAP response differs from a CoAP ACK; ACKs are never sent by servers in response to a multicast CoAP request. This section first summarizes these normative behaviors and then presents additional guidelines for response suppression. Also a number of multicast example applications are given to illustrate the overall approach.

To apply any rules for request and/or response suppression, a CoAP server must be aware that an incoming request arrived via multicast by making use of APIs such as IPV6_RECVPKTINFO [RFC3542].

For multicast request acceptance, the REQUIRED behaviors are:

- A server SHOULD NOT accept a multicast request that cannot be "authenticated" in some way (cryptographically or by some multicast boundary limiting the potential sources) [I-D.ietf-core-coap]. See Section 6.3 for examples of multicast boundary limiting methods.

- A server SHOULD NOT accept a multicast discovery request with a query string (as defined in CoRE Link Format [RFC6690]) if filtering ([RFC6690]) is not supported by the server.

- A server SHOULD NOT accept a multicast request that acts on a specific resource for which multicast support is not required. (Note that for the discovery resource "/.well-known/core" multicast support is always required. Implementers are advised to disable multicast support by default on any other resource, until explicitly enabled by an application or by configuration.)

- Otherwise accept the multicast request.

For multicast response suppression, the REQUIRED behaviors are:

- A server SHOULD NOT respond to a multicast discovery request if the filter specified by the request's query string does not match.

- A server MAY choose not to respond to a multicast request, if there's nothing useful to respond (e.g., error or empty response).

- If the server API cannot indicate that an incoming message was multicast, then the server SHOULD NOT respond for incoming messages for selected resources which are known (through application knowledge) to be used for multicast requests.

- Otherwise respond to the multicast request.

The above response suppression behaviors are complemented by the following guidelines. CoAP servers SHOULD implement configurable response suppression, enabling at least the following options per resource that supports multicast requests:

- Suppression of all 2.xx success responses;

- Suppression of all 4.xx client errors;

- Suppression of all 5.xx server errors;

- Suppression of all 2.05 responses with empty payload.

A number of group communication example applications are given below to illustrate how to make use of response suppression:

- CoAP resource discovery: Suppress 2.05 responses with empty payload and all 4.xx and 5.xx errors.

- Lighting control: Suppress all 2.xx responses after a lighting change command.

- Update configuration data in a group of devices using multicast PUT: No suppression at all. The client uses collected responses to identify which group members did not receive the new configuration; then attempts using CoAP CON unicast to update those specific group members.

- Multicast firmware update by sending blocks of data: Suppress all 2.xx and 5.xx responses. After having sent all multicast blocks, the client checks each endpoint by unicast to identify which data blocks are still missing in each endpoint.

- Conditional reporting for a group (e.g., sensors) based on a URI query: Suppress all 2.05 responses with empty payload (i.e., if a query produces no matching results).

3.8. Congestion Control

Multicast CoAP requests may result in a multitude of responses from different nodes, potentially causing congestion. Therefore both the sending of multicast requests, and the sending of the unicast CoAP responses to these multicast requests should be conservatively controlled.

CoAP [I-D.ietf-core-coap] reduces multicast-specific congestion risks through the following measures:

- A server MAY choose not to respond to a multicast request if there's nothing useful to respond (e.g., error or empty response). See Section 3.7 for more detailed guidelines on response suppression.

- A server SHOULD limit the support for multicast requests to specific resources where multicast operation is required.

- A multicast request MUST be Non-Confirmable.

- A response to a multicast request SHOULD be Non-Confirmable (Section 5.2.3 of [I-D.ietf-core-coap]).

- A server does not respond immediately to a multicast request, but SHOULD first wait for a time that is randomly picked within a predetermined time interval called the Leisure.

- A server SHOULD NOT accept multicast requests that can not be authenticated in some way. See Section 3.7 for more details on request suppression and multicast source authentication.

Additional guidelines to reduce congestion risks defined in this document are:

- A server in an LLN should only support multicast GET for resources that are small. For example, the payload of the response is 5% of the IP Maximum Transmit Unit (MTU) size (e.g. so it fits into a single link-layer frame).

- A server can minimize the payload length in response to a multicast GET on "/.well-known/core" by using hierarchy in arranging link descriptions for the response. An example of this is given in Section 5 of [RFC6690].

- Alternatively a server can also minimize the payload length of a response to a multicast GET (e.g., on "/.well-known/core") using CoAP blockwise transfers [I-D.ietf-core-block], returning only a first block of the CoRE Link Format description. For this reason, a CoAP client sending a multicast CoAP request to "/.well-known/core" SHOULD support core-block.

- A client should always aim to use IP multicast with link-local scope if possible. If this is not possible, then site-local scope IP multicast should be considered. If this is not possible, then global scope IP multicast should be considered as a last resort only.

More guidelines specific to use of CoAP in 6LoWPAN networks are given in Section 5.3.

3.9. Proxy Operation

CoAP [I-D.ietf-core-coap] allows a client to request a forward-proxy to process its CoAP request. For this purpose the client either specifies the request URI as a string in the Proxy-URI option, or it specifies the Proxy-Scheme option with the URI constructed from the usual Uri-* options. This approach works well for unicast requests. However, there are certain issues and limitations of processing the (unicast) responses to a group communication request made in this manner through a proxy. Specifically, if a proxy would apply aggregation of responses in such a case:

- Aggregation of (unicast) responses to a group communication request in a proxy is difficult. This is because the proxy does not know how many members there are in the group or how many group members will actually respond.

- There is no default format defined in CoAP for aggregation of multiple responses into a single response.

Alternatively, if a proxy follows directly the specification for a CoAP Proxy [I-D.ietf-core-coap], the proxy would simply forward all the individual (unicast) responses to a group communication request to the client (i.e., no aggregation). There are also issues with this approach:

- The client may be confused as it may not have known that the Proxy-URI contained a multicast target. That is, the client may be expecting only one (unicast) response but instead receives multiple (unicast) responses potentially leading to fault conditions in the application.

- Each individual CoAP response will appear to originate (IP Source address) from the CoAP Proxy, and not from the server that produced the response. This makes it impossible for the client to identify the server that produced each response.

Due to above issues, a guideline is defined here that a CoAP Proxy SHOULD NOT support processing a multicast CoAP request but rather return a 501 (Not Implemented) response in such case. The exception case here (i.e., to process it) is allowed under following conditions:

- The CoAP Proxy MUST be explicitly configured (whitelist) to allow proxied multicast requests by specific client(s).

- The proxy SHOULD return individual (unicast) CoAP responses to the client (i.e., not aggregated). The exception case here occurs when a (future) standardized aggregation format is being used.

- It MUST be known to the person/entity doing the configuration of the proxy, or otherwise verified in some way, that the client configured in the whitelist supports receiving multiple responses to a proxied unicast CoAP request.

3.10. Exceptions

Group communication using IP multicast offers improved network efficiency and latency amongst other benefits. However, group communication may not always be possible to implement in a given network. The primary reason for this will be if IP multicast is not (fully) supported in the network. For example, in an LLN where the RPL protocol is used for routing in "Non-storing mode" [RFC6550] and no other routing/forwarding protocol is defined, there will be no IP multicast routing beyond link-local scope. This means that any CoAP group communication above link-local scope will not be supported in this network.

4. Use Cases and Corresponding Protocol Flows

4.1. Introduction

The use of CoAP group communication is shown in the context of the following two use cases and corresponding protocol flows:

- Discovery of Resource Directory (RD, [I-D.ietf-core-resource-directory]): discovering the local CoAP RD which contains links (URIs) to resources stored on other CoAP servers [RFC6690].

- Lighting Control: synchronous operation of a group of IPv6-connected lights (e.g., 6LoWPAN [RFC4944] lights).

4.2. Network Configuration

To illustrate the use cases we define two network configurations. Both are based on the topology as shown in Figure 1. The two configurations using this topology are:

1. Subnets are 6LoWPAN networks; the routers Rtr-1 and Rtr-2 are 6LoWPAN Border Routers (6LBRs, [RFC6775]).

2. Subnets are Ethernet links; the routers Rtr-1 and Rtr-2 are multicast-capable Ethernet routers.

Both configurations are further specified by the following:

- A large room (Room-A) with three lights (Light-1, Light-2, Light-3) controlled by a Light Switch. The devices are organized into two subnets. In reality, there could be more lights (up to several hundreds) but these are not shown for clarity.

- Light-1 and the Light Switch are connected to a router (Rtr-1).

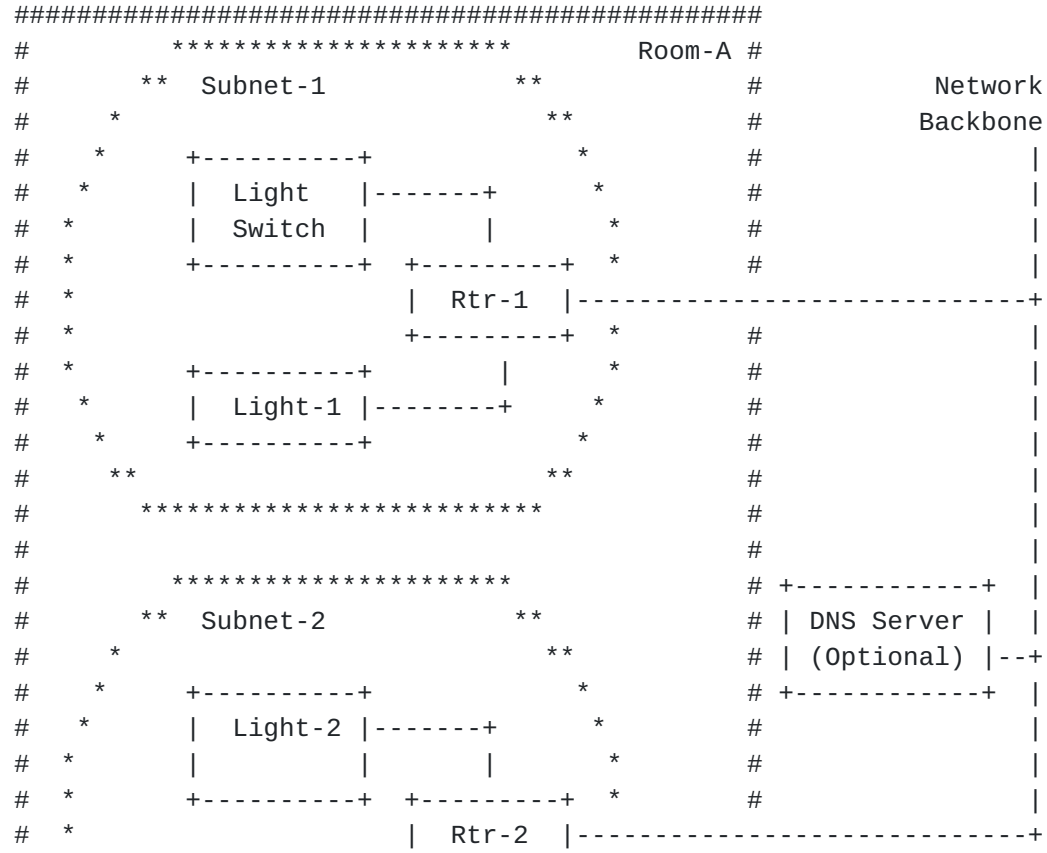
- Light-2 and the Light-3 are connected to another router (Rtr-2).

- The routers are connected to an IPv6 network backbone which is also multicast enabled. In the general case, this means the network backbone and Rtr-1/Rtr-2 support a PIM based multicast routing protocol, and Multicast Listener Discovery (MLD) for forming groups. In a limited case where the network backbone is one link, then the routers only have to support MLD-snooping (Appendix A) for the following use cases to work.

- A CoAP RD is connected to the network backbone.

- The DNS server is optional. If the server is there (connected to the network backbone) then certain DNS based features are available (e.g., DNS resolution of hostname to IP multicast address). If the DNS server is not there, then different provisioning of the network is required (e.g., IP multicast addresses are hard-coded into devices, or manually configured, or obtained via a service discovery method).

- A Controller (CoAP client) is connected to the backbone, which is able to control various building functions including lighting.



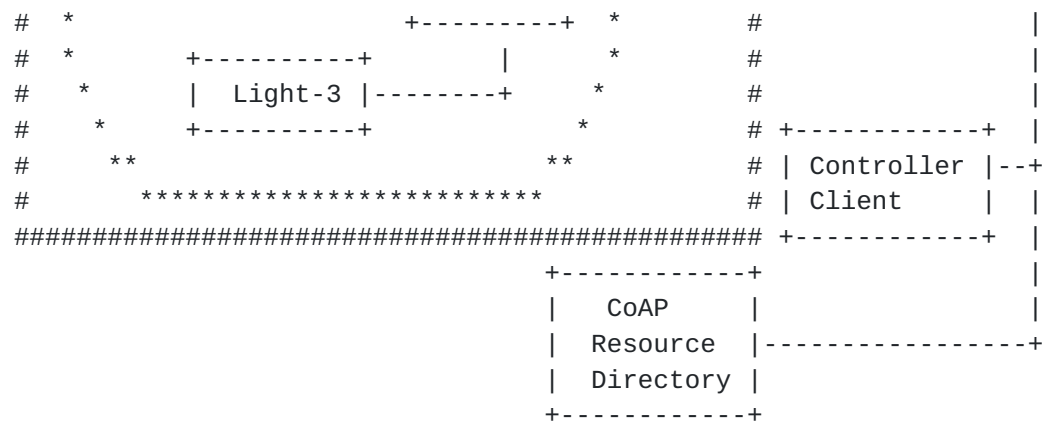


Figure 1: Network Topology of a Large Room (Room-A)

4.3. Discovery of Resource Directory

The protocol flow for discovery of the CoAP RD for the given network (of Figure 1) is shown in Figure 2:

- Light-2 is installed and powered on for the first time.
- Light-2 will then search for the local CoAP RD by sending out a GET request (with the "/.well-known/core?rt=core.rd" request URI) to the site-local "All CoAP Nodes" multicast address.
- This multicast message will then go to each node in subnet-2. Rtr-2 will then forward into to the Network Backbone where it will be received by the CoAP RD. All other nodes in subnet-2 will ignore the multicast GET because it is qualified by the query string "?rt=core.rd" (which indicates it should only be processed by the endpoint if it contains a resource of type core.rd).
- The CoAP RD will then send back a unicast response containing the requested content, which is a CoRE Link Format representation of a resource of type core.rd.
- Note that the flow is shown only for Light-2 for clarity. Similar flows will happen for Light-1, Light-3 and the Light Switch when they are first powered on.

The CoAP RD may also be discovered by other means such as by assuming a default location (e.g., on a 6LBR), using DHCP, anycast address, etc. However, these approaches do not invoke CoAP group communication so are not further discussed here.

For other discovery use cases such as discovering local CoAP servers, services or resources group communication can be used in a similar fashion as in the above use case. Both Link-Local (LL) and site-local discovery are possible this way.

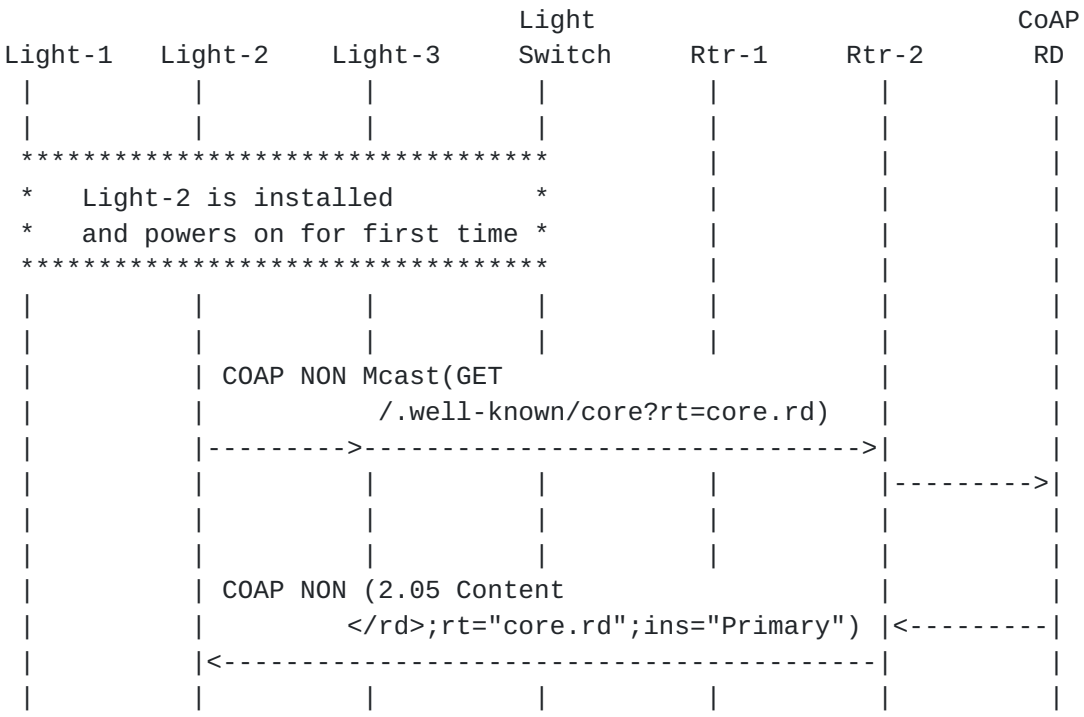


Figure 2: Resource Directory Discovery via Multicast Request

4.4. Lighting Control

The protocol flow for a building automation lighting control scenario for the network (Figure 1) is shown in Figure 3. The network is assumed to be in a 6LoWPAN configuration. Also, it is assumed that the CoAP servers in each Light are configured to suppress CoAP responses for any multicast CoAP requests related to lighting control. (See Section 3.7 for more details on response suppression by a server.)

In addition, Figure 4 shows a protocol flow example for the case that servers do respond to a lighting control multicast request with (unicast) CoAP NON responses. There are two success responses and one 5.00 error response. In this particular case, the Light Switch does not check that all Lights in the group received the multicast request by examining the responses. This is because the Light Switch is not configured with an exhaustive list of the IP addresses of all

Lights belonging to the group. However, based on received error responses it could take additional action such as logging a fault or alerting the user via its LCD display.

Reliability of CoAP multicast is not guaranteed. Therefore, one or more lights in the group may not have received the CoAP control request due to packet loss. In this use case there is no detection nor correction of such situations: the application layer expects that the multicast forwarding/routing will be of sufficient quality to provide on average a very high probability of packet delivery to all CoAP endpoints in a multicast group. An example protocol to accomplish this using randomized retransmission is the MPL forwarding protocol for LLNs [I-D.ietf-roll-trickle-mcast].

We assume the following steps have already occurred before the illustrated flows:

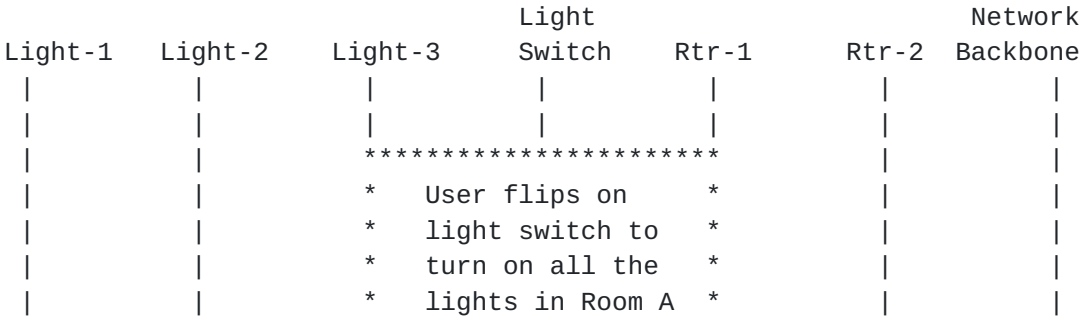
1. Startup phase: 6LoWPANs are formed. IPv6 addresses assigned to all devices. The CoAP network is formed.

2. Network configuration (application-independent): 6LBRs are configured with multicast addresses, or address blocks, to filter out or to pass through to/from the 6LoWPAN.

3. Commissioning phase (application-related): The IP multicast address of the group (Room-A-Lights) has been configured in all the Lights and in the Light Switch.

4. As an alternative to the previous step, when a DNS server is available, the Light Switch and/or the Lights have been configured with a group hostname which each nodes resolves to the above IP multicast address of the group.

Note for the Commissioning phase: the switch's 6LoWPAN/CoAP software stack supports sending unicast, multicast or proxied unicast CoAP requests, including processing of the multiple responses that may be generated by a multicast CoAP request.



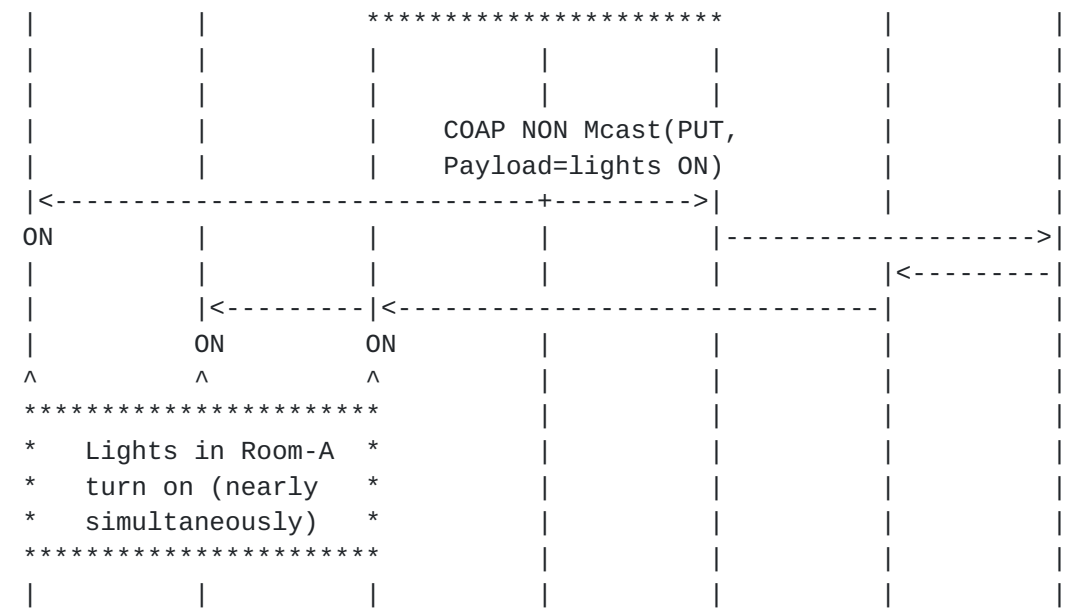


Figure 3: Light Switch Sends Multicast Control Message

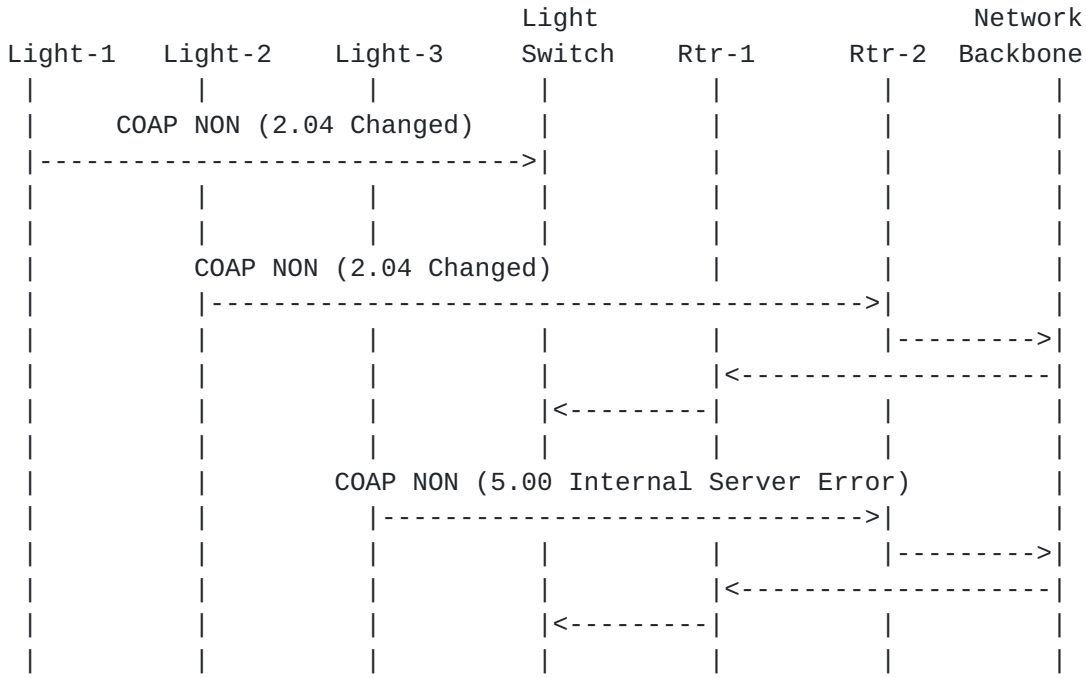


Figure 4: Lights (Optionally) Respond to Multicast CoAP Request

Another, but similar, lighting control use case is shown in Figure 5. In this case a controller connected to the Network Backbone sends a CoAP multicast request to turn on all lights in Room-A. Every Light sends back a CoAP response to the Controller after being turned on.

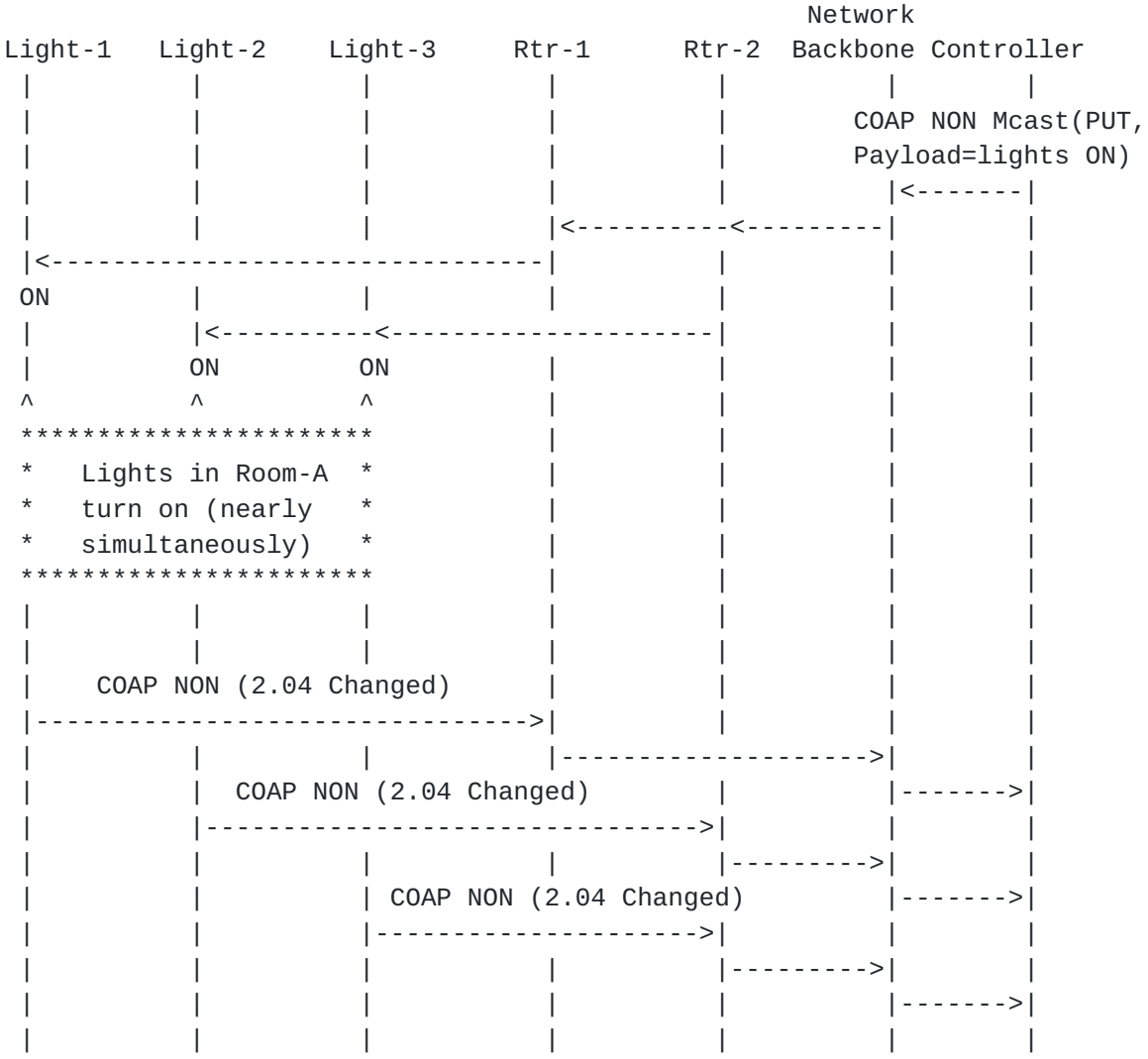


Figure 5: Controller On Backbone Sends Multicast Control Message

4.5. Lighting Control in MLD Enabled Network

The use case of previous section can also apply in networks where nodes support the MLD protocol [RFC3810]. The Lights then take on the role of MLDv2 listener and the routers (Rtr-1, Rtr-2) are MLDv2 Routers. In the Ethernet based network configuration, MLD may be available on all involved network interfaces. Use of MLD in the 6LoWPAN based configuration is also possible, but requires MLD

support in all nodes in the 6LoWPAN which is usually not implemented in many deployments.

The resulting protocol flow is shown in Figure 6. This flow is executed after the commissioning phase, as soon as Lights are configured with a group address to listen to. The (unicast) MLD Reports may require periodic refresh activity as specified by the MLD protocol. In the figure, LL denotes Link Local communication.

After the shown sequence of MLD Report messages has been executed, both Rtr-1 and Rtr-2 are automatically configured to forward multicast traffic destined to Room-A-Lights onto their connected subnet. Hence, no manual Network Configuration of routers, as previously indicated in Section 4.4, is needed anymore.

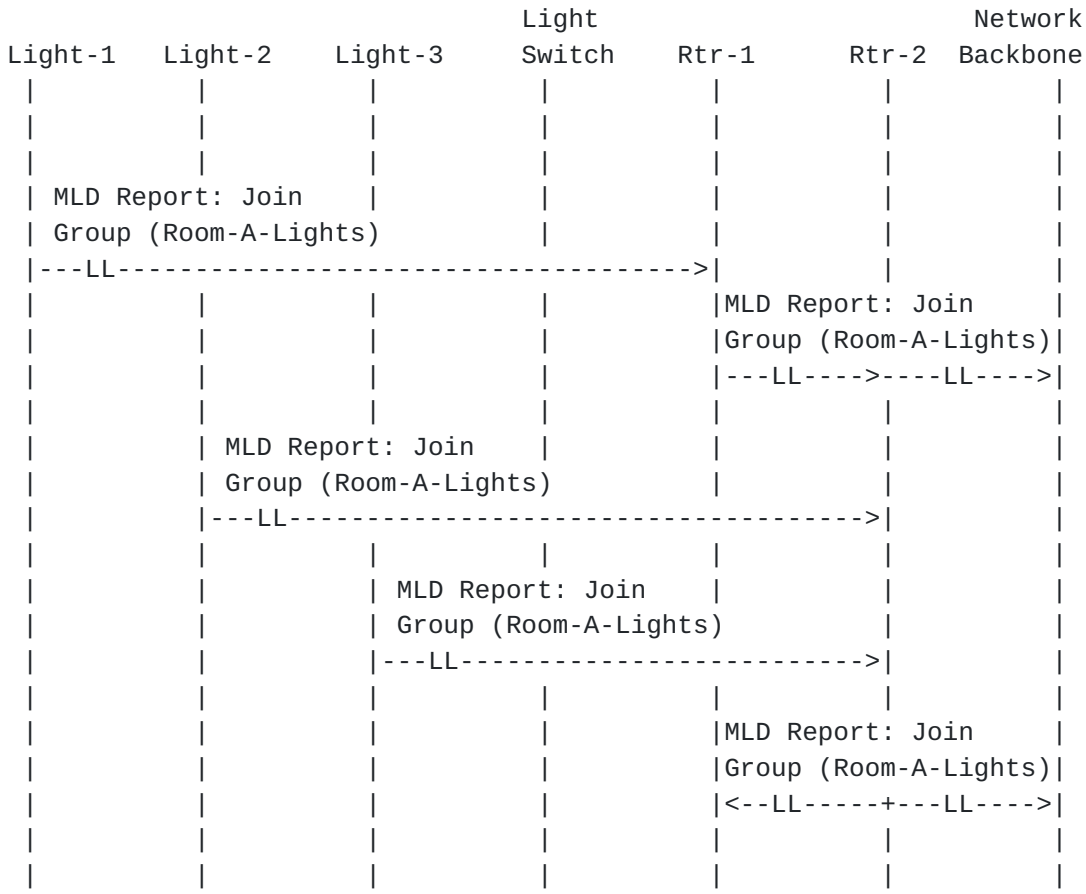


Figure 6: Joining Lighting Groups Using MLD

4.6. Commissioning the Network Based On Resource Directory

This section outlines how devices in the lighting use case (both Switches and Lights) can be commissioned, making use of Resource Directory [I-D.ietf-core-resource-directory] and its group configuration feature.

Once the Resource Directory (RD) is discovered, the Switches and Lights need to be discovered and their groups need to be defined. For the commissioning of these devices, a commissioning tool can be used that defines the entries in the RD. The commissioning tool has the authority to change the contents of the RD and the Light/Switch nodes. DTLS based security is used by the commissioning tool to modify operational data in RD, Switches and Lights.

In our particular use case, a group of three lights is defined with one multicast address and hostname "Room-A-Lights.floor1.west.bldg6.example.com". The commissioning tool has a list of the three lights and the associated multicast address. For each light in the list the tool learns the IP address of the light and instructs the RD with three POST commands to store the end-points associated with the three lights as prescribed by the RD specification. Finally the commissioning tool defines the group in the RD to contain these three end-points. Also the commissioning tool writes the multicast address in the Lights with, for example, the POST /gp command discussed in Section 3.6.

The light switch can discover the group in RD and thus learn the multicast address of the group. The light switch will use this address to send multicast commands to the members of the group. When the message arrives the Lights should recognize the multicast address and accept the message.

5. Deployment Guidelines

This section provides guidelines how an IP Multicast based solution for CoAP group communication can be deployed in various network configurations.

5.1. Target Network Topologies

CoAP group communication can be deployed in various network topologies. First, the target network may be a regular IP network, or a LLN such as a 6LoWPAN network, or consist of mixed constrained/ unconstrained network segments. Second, it may be a single subnet only or multi-subnet; e.g., multiple 6LoWPAN networks joined by a single backbone LAN. Third, a wireless network segment may have all its nodes reachable in a single IP hop (fully connected), or it may require multiple IP hops for some pairs of nodes to reach each other.

Each topology may pose different requirements on the configuration of routers and protocol(s), in order to enable efficient CoAP group communication.

5.2. Advertising Membership of Multicast Groups

If a multicast routing/forwarding protocol is used in a network, server nodes that intend to receive CoAP multicast requests generally require a method to advertise the specific IP multicast address(es) they want to receive (i.e., a method to join specific IP multicast groups). This section identifies two ways in which group joining is accomplished (with MLD, with RPL) and one situation (with MPL) where group joining is not required.

5.2.1. Using the MLD Listener Protocol

CoAP nodes that are IP hosts (i.e., not IP routers) are generally unaware of the specific multicast routing/forwarding protocol being used. When such a host needs to join a specific (CoAP) multicast group, it requires a way to signal to multicast routers which multicast traffic it wants to receive. For efficient multicast routing (i.e., avoid always flooding IP multicast packets), routers must know which hosts need to receive packets addressed to specific IP multicast destinations.

The Multicast Listener Discovery (MLD) protocol [RFC3810] (Appendix A) is the standard IPv6 method to achieve this. [RFC6636] discusses tuning of MLD for mobile and wireless networks. These guidelines may be useful when implementing MLD in LLNs.

Alternatively, to avoid the use of MLD in LLN deployments, either all nodes can be configured as multicast routers in an LLN, or a multicast forwarding/flooding protocol can be used that forwards any IP multicast packet to all forwarders (routers) in the LLN.

5.2.2. Using the RPL Routing Protocol

The RPL routing protocol [RFC6550] defines in Section 12 the advertisement of IP multicast destinations using DAO messages. This mechanism can be used by CoAP nodes (which are also RPL routers) to advertise IP multicast group membership to other RPL routers. Then, the RPL protocol can route multicast CoAP requests over multiple hops to the correct CoAP servers.

This mechanism can be used as a means to convey IP multicast group membership information to an edge router (e.g., 6LBR), in case the edge router is also the root of the RPL DODAG. This could be useful in LLN segments where MLD is not available and the edge router needs

to know what IP multicast traffic to pass through from the backbone network into the LLN subnet.

5.2.3. Using the MPL Forwarding Protocol

The MPL forwarding protocol [I-D.ietf-roll-trickle-mcast] can be used in a predefined network domain for propagation of IP multicast packets to all MPL routers, over multiple hops. MPL is designed to work in LLN deployments. There is one specific case in which there is no need for CoAP server nodes to advertise IP multicast group membership. This case occurs when any IP multicast source is inside the MPL domain and if all nodes that listen to IP multicast CoAP requests are also MPL routers.

5.3. 6LoWPAN Specific Guidelines

To support multi-LoWPAN scenarios for CoAP group communication, it is recommended that a 6LoWPAN Border Router (6LBR) will act in an MLD Router role on the backbone link. If this is not possible then the 6LBR should be configured to act as an MLD Multicast Address Listener and/or MLD Snooper (Appendix A) on the backbone link.

To avoid that backbone IP multicast traffic needlessly congests 6LoWPAN network segments, it is recommended that a filtering means is implemented to block IP multicast traffic from 6LoWPAN segments where none of the 6LoWPAN nodes listen to this traffic. Possible means are:

- Filtering in 6LBRs based on information from the routing/ forwarding protocol. This allows a 6LBR to only forward multicast traffic onto the LoWPAN, for which it is known that there exists at least one listener on the LoWPAN. This does not work for all protocols, for example in MPL this is not defined.

- Filtering in 6LBRs based on MLD reports. Similar as previous but based directly on MLD reports from 6LoWPAN nodes. This only works in a single-IP-hop 6LoWPAN network, such as a mesh-under routing network or a star topology network, because MLD relies on link-local communication.

- Filtering in 6LBRs based on settings. Filtering tables with blacklists/whitelists can be configured in the 6LBR by system administration for all 6LBRs or configured on a per-6LBR basis.

- Filtering in router(s) or firewalls that provide access to constrained network segments. For example, in an access router/ bridge that connects a regular intranet LAN to a building control IPv6 segment. This building control segment connects multiple 6LoWPAN subnets, each subnet connected via one 6LBR.

6. Security Considerations

This section describes the relevant security configuration for CoAP group communication using IP multicast. The threats to CoAP group communication are also identified and various approaches to mitigate these threats are summarized.

6.1. Security Configuration

As defined in [I-D.ietf-core-coap], CoAP group communication based on IP multicast:

- MUST operate in CoAP NoSec (No Security) mode.

- MUST NOT use "coaps" scheme. That is, all group communication MUST use only "coap" scheme.

6.2. Threats

Essentially the above configuration means that there is no security at the CoAP layer for group communication. This is due to the fact that the current DTLS based approach for CoAP is exclusively unicast oriented and does not support group security features such as group key exchange and group authentication. As a direct consequence of this, CoAP group communication is vulnerable to all attacks mentioned in [I-D.ietf-core-coap] for IP multicast.

6.3. Threat Mitigation

The [I-D.ietf-core-coap] identifies various threat mitigation techniques for CoAP multicast. In addition to those guidelines, it is recommended that for sensitive data or safety-critical control, a combination of appropriate link-layer security and administrative control of IP multicast boundaries should be used. Some examples are given below.

6.3.1. WiFi Scenario

In a home automation scenario (using WiFi), the WiFi encryption should be enabled to prevent rogue nodes from joining. Also, if MAC address filtering at the WiFi Access Point is supported that should also be enabled. The IP router should have the firewall enabled to

isolate the home network from the rest of the Internet. In addition, the domain of the IP multicast should be set to be either link-local scope or site-local scope. Finally, if possible, devices should be configured to accept only Source Specific Multicast (SSM) packets (see [RFC4607]) from within the trusted home network. For example, all lights in a particular room should only accept IP multicast traffic originating from the master light switch in that room. In this case, the Spoofed Source Address considerations of Section 7.4 of [RFC4607] should be heeded.

6.3.2. 6LoWPAN Scenario

In a building automation scenario, a particular room may have a single 6LoWPAN network with a single Edge Router (6LBR). Nodes on the subnet can use link-layer encryption to prevent rogue nodes from joining. The 6LBR can be configured so that it blocks any incoming (6LoWPAN-bound) IP multicast traffic. Another example topology could be a multi-subnet 6LoWPAN in a large conference room. In this case, the backbone can implement port authentication (IEEE 802.1X) to ensure only authorized devices can join the Ethernet backbone. The access router to this secured network segment can also be configured to block incoming IP multicast traffic.

6.3.3. Future Evolution

In the future, to further mitigate the threats, the developing approach for DTLS-based IP multicast security for CoAP networks (see [I-D.keoh-tls-multicast-security]) or similar approaches should be considered once they mature.

7. IANA Considerations

7.1. New 'core.gp' Resource Type

This memo registers a new resource type (rt) from the CoRE Parameters Registry called 'core.gp'.

(Note to IANA/RFC Editor: This registration follows the process described in section 7.4 of [RFC6690]).

Attribute Value: core.gp

Description: Group Configuration resource. This resource is used to query/manage the group membership of a CoAP server.

Reference: See Section 3.6.

7.2. New 'coap-group+json' Internet Media Type

This memo registers a new Internet Media Type for CoAP group configuration resource called 'application/coap-group+json'.

(Note to IANA/RFC Editor: This registration follows the guidance from [RFC6839], and (last paragraph) of section 12.3 of [I-D.ietf-core-coap].

Type name: application

Subtype name: coap-group+json

Required parameters: None

Optional parameters: None

Encoding considerations: 8bit if UTF-8; binary if UTF-16 or UTF-32.

JSON may be represented using UTF-8, UTF-16, or UTF-32. When JSON is written in UTF-8, JSON is 8bit compatible. When JSON is written in UTF-16 or UTF-32, the binary content-transfer-encoding must be used.

If the client is aware that the server group configuration resource is 8bit encoded (which is most efficient for a constrained device), that encoding should be respected by the client (i.e., it should not try to replace it by a binary encoded group configuration resource).

Security considerations:

Denial of Service attacks could be performed by constantly setting the group configuration resource of a CoAP endpoint to different values. This will cause the endpoint to register (or de-register) from the related IP multicast group. To prevent this it is recommended that DTLS-secured CoAP communication be used for setting the group configuration resource. Thus only authorized clients will be allowed by a server to configure its group membership.

Interoperability considerations: None

Published specification: (This I-D when it becomes an RFC)

Applications that use this media type:

CoAP client and server implementations that wish to set/read the group configuration resource via 'application/coap-group+json' payload as described in Section 3.6.

Additional Information:

Magic number(s): None

File extension(s): *.json

Macintosh file type code(s): TEXT

Intended usage: COMMON

Restrictions on usage: None

Author: CoRE WG

Change controller: IETF

8. Acknowledgements

Thanks to Peter Bigot, Carsten Bormann, Anders Brandt, Angelo Castellani, Bjoern Hoehrmann, Matthias Kovatsch, Guang Lu, Salvatore Loreto, Kerry Lynn, Dale Seed, Zach Shelby, Peter van der Stok, and Juan Carlos Zuniga for their helpful comments and discussions that have helped shape this document.

9. References

9.1. Normative References

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542, May 2003.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

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

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, 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.

[RFC5771] Cotton, M., Vegoda, L., and D. Meyer, "IANA Guidelines for IPv4 Multicast Address Assignments", BCP 51, RFC 5771, March 2010.

[RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC6636] Asaeda, H., Liu, H., and Q. Wu, "Tuning the Behavior of the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) for Routers in Mobile and Wireless Networks", RFC 6636, May 2012.

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

[RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.

[RFC6839] Hansen, T. and A. Melnikov, "Additional Media Type Structured Syntax Suffixes", RFC 6839, January 2013.

[I-D.ietf-core-coap] Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

9.2. Informative References

[I-D.ietf-core-block] Bormann, C. and Z. Shelby, "Blockwise transfers in CoAP", draft-ietf-core-block-12 (work in progress), June 2013.

[I-D.vanderstok-core-dna] Stok, P., Lynn, K., and A. Brandt, "CoRE Discovery, Naming, and Addressing", draft-vanderstok-core-dna-02 (work in progress), July 2012.

[I-D.ietf-roll-trickle-mcast]
Hui, J. and R. Kelsey, "Multicast Protocol for Low power and Lossy Networks (MPL)", draft-ietf-roll-trickle-mcast-04 (work in progress), February 2013.

[I-D.keoh-tls-multicast-security]
Keoh, S., Kumar, S., and E. Dijk, "DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs)", draft-keoh-tls-multicast-security-00 (work in progress), October 2012.

[I-D.ietf-core-resource-directory]
Shelby, Z., Krco, S., and C. Bormann, "CoRE Resource Directory", draft-ietf-core-resource-directory-00 (work in progress), June 2013.

Appendix A. Multicast Listener Discovery (MLD)

In order to extend the scope of IP multicast beyond link-local scope, an IP multicast routing or forwarding protocol has to be active in routers on an LLN. To achieve efficient multicast routing (i.e., avoid always flooding IP multicast packets), routers have to learn which hosts need to receive packets addressed to specific IP multicast destinations.

The Multicast Listener Discovery (MLD) protocol [RFC3810] (or its IPv4 pendant IGMP) is today the method of choice used by an (IP multicast enabled) router to discover the presence of multicast listeners on directly attached links, and to discover which multicast addresses are of interest to those listening nodes. MLD was specifically designed to cope with fairly dynamic situations in which multicast listeners may join and leave at any time.

IGMP/MLD Snooping is a technique implemented in some corporate LAN routing/switching devices. An MLD snooping switch listens to MLD State Change Report messages from MLD listeners on attached links. Based on this, the switch learns on what LAN segments there is interest for what IP multicast traffic. If the switch receives at some point an IP multicast packet, it uses the stored information to decide onto which LAN segment(s) to send the packet. This improves network efficiency compared to the regular behavior of forwarding every incoming multicast packet onto all LAN segments. An MLD snooping switch may also send out MLD Query messages (which is normally done by a device in MLD Router role) if no MLD Router is present.

[RFC6636] discusses optimal tuning of the parameters of MLD for routers for mobile and wireless networks. These guidelines may be useful when implementing MLD in LLNs.

Appendix B. Change Log

Changes from ietf-10 to ietf-11:

- Added text to section 3.8 (Congestion Control) to clarify that a "CoAP client sending a multicast CoAP request to /.well-known/core SHOULD support core-block" (#332).
- Various editorial updates for improved readability.

Changes from ietf-09 to ietf-10:

- Various editorial updates including:
- Added a fourth option in section 3.3 on ways to obtain the URI path for a group request.
- Clarified use of content format in GET/PUT requests for Configuring Group Membership in Endpoints (in section 3.6).
- Changed reference "draft-shelby-core-resource-directory" to "draft-ietf-core-resource-directory".
- Clarified (in section 3.7) that ACKs are never used for a multicast request (from #296).
- Clarified (in section 5.2/5.2.3) that MPL does not support group membership advertisement.
- Adding introductory paragraph to Scope (section 2.2).
- Wrote out fully the URIs in table section 3.2.
- Reworded security text in section 7.2 (New Internet Media Type) to make it consistent with section 3.6 (Configuring Group Membership).
- Fixed formatting of hyperlinks in sections 6.3 and 7.2.

Changes from ietf-08 to ietf-09:

- Cleaned up requirements language in general. Also, requirements language are now only used in section 3 (Protocol Considerations) and section 6 (Security Considerations). Requirements language

has been removed from other sections to keep them to a minimum (#271).

- Addressed final comment from Peter van der Stok to define what "IP stack" meant (#296). Following the lead of CoAP-17, we know refer instead to "APIs such as IPV6_RECVPKTINFO [RFC3542]".

- Changed text in section 3.4 (Group Methods) to allow multicast POST under specific conditions and highlighting the risks with using it (#328).

- Various editorial updates for improved readability.

Changes from ietf-07 to ietf-08:

- Updated text in section 3.6 (Configuring Group Membership in Endpoints) to make it more explicit that the Internet Media Type is used in the processing rules (#299).

- Addressed various comments from Peter van der Stok (#296).

- Various editorial updates for improved readability including defining all acronyms.

Changes from ietf-06 to ietf-07:

- Added an IANA request (in section 7.2) for a dedicated content-format (Internet Media type) for the group management JSON format called 'application/coap-group+json' (#299).

- Clarified semantics (in section 3.6) of group management JSON format (#300).

- Added details of IANA request (in section 7.1) for a new CORE Resource Type called 'core.gp'.

- Clarified that DELETE method (in section 3.6) is also a valid group management operation.

- Various editorial updates for improved readability.

Changes from ietf-05 to ietf-06:

- Added a new section on commissioning flow when using discovery services when end devices discover in which multicast group they are allocated (#295).

- Added a new section on CoAP Proxy Operation (section 3.9) that outlines the potential issues and limitations of doing CoAP multicast requests via a CoAP Proxy (#274).

- Added use case of multicasting controller on the backbone (#279).

- Use cases were updated to show only a single CoAP RD (to replace the previous multiple RDs with one in each subnet). This is a more efficient deployment and also avoids RD specific issues such as synchronization of RD information between serves (#280).

- Added text to section 3.6 (Configuring Group Membership in Endpoints) that clarified that any (unicast) operation to change an endpoint's group membership must use DTLS-secured CoAP.

- Clarified relationship of this document to [I-D.ietf-core-coap] in section 2.2 (Scope).

- Removed IPSec related requirement, as IPSec is not part of [I-D.ietf-core-coap] anymore.

- Editorial reordering of subsections in section 3 to have a better flow of topics. Also renamed some of the (sub)sections to better reflect their content. Finally, moved the URI Configuration text to the same section as the Port Configuration section as it was a more natural grouping (now in section 3.3) .

- Editorial rewording of section 3.7 (Multicast Request Acceptance and Response Suppression) to make the logic easier to comprehend (parse).

- Various editorial updates for improved readability.

Changes from ietf-04 to ietf-05:

- Added a new section 3.9 (Exceptions) that highlights that IP multicast (and hence group communication) is not always available (#187).

- Updated text on the use of [RFC2119] language (#271) in Section 1.

- Included guidelines on when (not) to use CoAP responses to multicast requests and when (not) to accept multicast requests (#273).

- Added guideline on use of core-block for minimizing response size (#275).

- Restructured section 6 (Security Considerations) to more fully describe threats and threat mitigation (#277).

- Clearly indicated that DNS resolution and reverse DNS lookup are optional.

- Removed confusing text about a single group having multiple IP addresses. If multiple IP addresses are required then multiple groups (with the same members) should be created.

- Removed repetitive text about the fact that group communication is not guaranteed.

- Merged previous section 5.2 (Multicast Routing) into 3.1 (IP Multicast Routing Background) and added link to section 5.2 (Advertising Membership of Multicast Groups).

- Clarified text in section 3.8 (Congestion Control) regarding precedence of use of IP multicast domains (i.e. first try to use link-local scope, then site-local scope, and only use global IP multicast as a last resort).

- Extended group resource manipulation guidelines with use of pre-configured ports/paths for the multicast group.

- Consolidated all text relating to ports in a new section 3.3 (Port Configuration).

- Clarified that all methods (GET/PUT/POST) for configuring group membership in endpoints should be unicast (and not multicast) in section 3.7 (Configuring Group Membership In Endpoints).

- Various editorial updates for improved readability, including editorial comments by Peter van der Stok to WG list of December 18th, 2012.

Changes from ietf-03 to ietf-04:

- Removed section 2.3 (Potential Solutions for Group Communication) as it is purely background information and moved section to draft-dijk-core-groupcomm-misc (#266).

- Added reference to draft-keoh-tls-multicast-security to section 6 (Security Considerations).

- Removed Appendix B (CoAP-Observe Alternative to Group Communications) as it is as an alternative to IP Multicast that the WG has not adopted and moved section to draft-dijk-core-groupcomm-misc (#267).

- Deleted section 8 (Conclusions) as it is redundant (#268).

- Simplified light switch use case (#269) by splitting into basic operations and additional functions (#269).

- Moved section 3.7 (CoAP Multicast and HTTP Unicast Interworking) to draft-dijk-core-groupcomm-misc (#270).

- Moved section 3.3.1 (DNS-SD) and 3.3.2 (CoRE Resource Directory) to draft-dijk-core-groupcomm-misc as these sections essentially just repeated text from other drafts regarding DNS based features. Clarified remaining text in this draft relating to DNS based features to clearly indicate that these features are optional (#272).

- Focus section 3.5 (Configuring Group Membership) on a single proposed solution.

- Scope of section 5.3 (Use of MLD) widened to multicast destination advertisement methods in general.

- Rewrote section 2.2 (Scope) for improved readability.

- Moved use cases that are not addressed to draft-dijk-core-groupcomm-misc.

- Various editorial updates for improved readability.

Changes from ietf-02 to ietf-03:

- Clarified that a group resource manipulation may return back a mixture of successful and unsuccessful responses (section 3.4 and Figure 6) (#251).

- Clarified that security option for group communication must be NoSec mode (section 6) (#250).

- Added mechanism for group membership configuration (#249).

- Removed IANA request for multicast addresses (section 7) and replaced with a note indicating that the request is being made in [I-D.ietf-core-coap] (#248).

- Made the definition of 'group' more specific to group of CoAP endpoints and included text on UDP port selection (#186).

- Added explanatory text in section 3.4 regarding why not to use group communication for non-idempotent messages (i.e. CoAP POST) (#186).

- Changed link-local RD discovery to site-local in RD discovery use case to make it more realistic.

- Fixed lighting control use case CoAP proxying; now returns individual CoAP responses as in coap-12.

- Replaced link format I-D with RFC6690 reference.

- Various editorial updates for improved readability

Changes from ietf-01 to ietf-02:

- Rewrote congestion control section based on latest CoAP text including Leisure concept (#188)

- Updated the CoAP/HTTP interworking section and example use case with more details and use of MLD for multicast group joining

- Key use cases added (#185)

- References to [I-D.vanderstok-core-dna] and draft-castellani-core-advanced-http-mapping added

- Moved background sections on "MLD" and "CoAP-Observe" to Appendices

- Removed requirements section (and moved it to draft-dijk-core-groupcomm-misc)

- Added details for IANA request for group communication multicast addresses

- Clarified text to distinguish between "link local" and general multicast cases

- Moved lengthy background section 5 to draft-dijk-core-groupcomm-misc and replaced with a summary

- Various editorial updates for improved readability

- Change log added

Changes from ietf-00 to ietf-01:

- Moved CoAP-observe solution section to section 2
- Editorial changes
- Moved security requirements into requirements section
- Changed multicast POST to PUT in example use case
- Added CoAP responses in example use case

Changes from rahman-07 to ietf-00:

- Editorial changes
- Use cases section added
- CoRE Resource Directory section added
- Removed section 3.3.5. IP Multicast Transmission Methods
- Removed section 3.4 Overlay Multicast
- Removed section 3.5 CoAP Application Layer Group Management
- Clarified section 4.3.1.3 RPL Routers with Non-RPL Hosts case
- References added and some normative/informative status changes

Authors' Addresses

Akbar Rahman (editor)
InterDigital Communications, LLC

Email: Akbar.Rahman@InterDigital.com

Esko Dijk (editor)
Philips Research

Email: esko.dijk@philips.com