CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: January 21, 2015

A. Rahman, Ed.
InterDigital Communications, LLC
E. Dijk, Ed.
Philips Research
July 20, 2014

Group Communication for CoAP
draft-ietf-core-groupcomm-20

Abstract

CoAP is a specialized web 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 21, 2015.

Table of Contents

1. Introduction

1.1. Background

Constrained Application Protocol (CoAP) is a Representational State Transfer (REST) based web transfer protocol for resource constrained devices operating in an IP network [RFC7252]. CoAP has many similarities to HTTP [RFC7230] but also has some key differences. Constrained devices can be large in numbers, but are often related to each other in function or by 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.

1.2. Scope

Group communication involves a one-to-many relationship between CoAP endpoints. Specifically, a single CoAP client can simultaneously get (or set) resources 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 sending CoAP requests on top of IP multicast, and processing the (unicast IP) responses are given in Section 8 of [RFC7252]. The main contribution of this document lies in providing additional guidance for key CoAP group communication concepts. Among the topics covered are group definition, group RESTful methods, and group request and response processing (see Section 2). Also, proxy operation and minimizing network congestion for group communication is discussed (see Section 2). Finally,

specific use cases (see Section 3) and deployment guidelines (see Section 4) for group communication are outlined.

1.3. 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 [RFC7252]. In addition, this document defines the following terminology:

Group Communication
: A source node sends a single application layer (e.g. CoAP) 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 mechanisms for CoAP group communication are UDP/IP multicast for the requests, and unicast UDP/IP for the responses. The network involved may be a constrained network such as a low-power, lossy network.

Reliable Group Communication
: A special case of group communication where for each destination node it is guaranteed that the node either 1) eventually receives the message sent by the source node, or 2) does not receive the message and the source node is notified of the non-reception event.

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 approach 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]. A complete IP multicast solution may

include support for managing group memberships, and IP multicast routing/forwarding (see Section 2.1).

Low power and Lossy Network (LLN)
A type of constrained IP network where devices are interconnected by low-power and lossy links. The links may be may composed of one or more technologies such as IEEE 802.15.4, Bluetooth Low Energy (BLE), Digital Enhanced Cordless Telecommunication (DECT), and IEEE P1901.2 power-line communication.

2. Protocol Considerations

2.1. IP Multicast Background

IP multicast protocols have been evolving for decades, resulting in standards such as Protocol Independent Multicast - Sparse Mode (PIM-SM) [RFC4601]. 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 link-local scope, 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]. RPL and MPL do not depend on each other; each can be used in isolation and both can be used in combination in a network. Finally, PIM-SM [RFC4601] is often used for multicast routing in traditional IP networks (i.e. networks that are not constrained).

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 "listener" protocol may be needed for the devices to subscribe to groups (see Section 4.2).

IP multicast is generally classified as an unreliable service in that packets are not guaranteed to be delivered to each and every member of the group. In other words, it cannot be directly used as a basis for "reliable group communication" as defined in Section 1.3. However, the level of reliability can be increased by employing a

multicast protocol that performs periodic retransmissions as is done for example in MPL.

2.2. Group Definition and Naming

A CoAP group is defined as a set of CoAP endpoints, where each endpoint is configured to receive CoAP group communication requests that are sent to the group's associated IP multicast address. The individual response by each endpoint receiver to a CoAP group communication request is always sent back as unicast. An endpoint may be a member of multiple groups. Group membership of an endpoint may dynamically change over time.

All CoAP server nodes SHOULD join the "All CoAP Nodes" multicast group [RFC7252], Section 12.8) by default to enable CoAP discovery. For IPv4, the address is 224.0.1.187 and for IPv6 a server node joins at least both the link-local scoped address FF02::FD and the site-local scoped address FF05::FD. IPv6 addresses of other scopes MAY be enabled.

A CoAP 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 regular CoAP URI syntax [RFC7252].

Note: A group URI is needed to initiate CoAP group communications. For CoAP implementations it is recommended to use the URI composition method of Section 6.5 of [RFC7252] in such way that, from a group URI, a CoAP group communication request is generated.

For sending nodes, it is recommended to use the IP multicast address literal in a group URI. However, in case a group hostname is used, it can be uniquely mapped to an 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:

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 ([RFC1033]). Reverse mapping is important, for example, in trouble shooting to translate IP multicast addresses back to human readable hostnames to show in a diagnostics user interface.

2.3. 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. These rules imply that different ports (for the same IP multicast address) cannot be used to specify different CoAP groups.

CoAP group communication will not work if there is diversity in the authority port (e.g., different dynamic port addresses across the group) or if other parts of the group URI such as the path, or the query, differ on different endpoints. Therefore, some measures must be present to ensure uniformity in port number and resource names/ locations within a group. All CoAP group communication 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 group.

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

For a CoAP server node that supports resource discovery, the default port 5683 MUST be supported (Section 7.1 of [RFC7252] for the "All CoAP Nodes" group.

All CoAP group communication requests SHOULD operate on group URI paths in one of the following ways:

1. Pre-configured group 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. Note that ([RFC7320]). prescribes that any specification must not constrain, define structure or semantics for any path component.

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 group 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 it can use this URI in a CoAP group communication request. For example, a commissioning tool may instruct a sensor device in this way to which target group (group URI) it should report sensor events.

2.4. RESTful Methods

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

2.5. Request and Response Model

All CoAP requests that are sent via IP multicast MUST be Non-confirmable. The Message ID in an IP multicast CoAP message is used for optional message de-duplication as detailed in Section 4.5 of [RFC7252].

A server MAY send back a unicast response to the CoAP group communication request (e.g., response "2.05 Content" to a group GET request). The unicast responses received by the CoAP client 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. Detailed processing rules for IP multicast request acceptance and unicast response suppression are given in Section 2.8.

A CoAP request sent over IP multicast and any unicast response it causes must take into account the congestion control rules defined in Section 2.9.

The CoAP client can distinguish the origin of multiple server responses by source IP address of the UDP message containing the CoAP response, or any other available unique identifier (e.g. contained in the CoAP payload). In case a CoAP client sent multiple group requests, the responses are as usual matched to a request using the CoAP Token.

For multicast CoAP requests there are additional constraints on the re-use of Token values, compared to the unicast case. In the unicast case, receiving a response effectively frees up its Token value for re-use since no more responses will follow. However, for multicast CoAP the number of responses is not bounded a-priori. Therefore the reception of a response cannot be used as a trigger to "free up" a Token value for re-use. Re-using a Token value too early could lead to protocol error i.e. a wrong response/request matching in the client. Therefore the time between re-use of Token values (for Token values used in multicast requests) must be at least:

NON_LIFETIME + MAX_LATENCY + MAX_SERVER_RESPONSE_DELAY

where NON_LIFETIME and MAX_LATENCY are defined in Section 4.8 of [RFC7252]. MAX_SERVER_RESPONSE_DELAY is defined here as the expected maximum response delay over all servers that the client can send a multicast request to. This delay includes the maximum Leisure time period as defined in Section 8.2 of [RFC7252]. Using the CoAP default protocol parameters the re-use time becomes at least 250 seconds, but may need to be much longer in practice since there is no time limit defined in CoAP for generation of responses by a server.

2.6. Member Discovery

CoAP Groups, and the membership of these groups, can be discovered via the lookup interfaces in the Resource Directory (RD) defined in [I-D.ietf-core-resource-directory]. An example of doing some of these RD lookups is given in Section 3.6.

2.7. Membership Configuration

2.7.1. Background

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 using a specific service discovery means. Third, it may be configured by another node (e.g., a commissioning device).

In the first case, the pre-configured group information may be either an IP multicast address or a hostname (FQDN) which is resolved later (during operation) to an 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 case is detailed more in Section 3.6.

In the third case, typical in scenarios such as building control, a dynamic 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.

2.7.2. Membership Configuration RESTful Interface

To achieve better interoperability between endpoints from different manufacturers, an OPTIONAL CoAP membership configuration RESTful interface for configuring 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 CoAP methods (GET/PUT/POST/DELETE) only as it is a method of configuring group information in individual endpoints.

Also, a form of authorization (making use of DTLS-secured CoAP [RFC7252]) SHOULD be used such that only authorized controllers are allowed by an endpoint to configure its group membership.

It is important to note that other approaches may be used to configure CoAP endpoints with relevant group information. These alternate approaches may support a subset or super-set of the membership configuration RESTful interface described in this document. For example, a simple interface to just read the endpoint group information may be implemented via a classical Management Information Base (MIB) approach (e.g. following approach of [RFC3433]).

2.7.2.1. CoAP-Group Resource Type and Media Type

CoAP endpoints implementing the membership configuration RESTful interface MUST support the CoAP group configuration Internet Media Type "application/coap-group+json" (Section 6.2).

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 6.1). An authorized client uses this media type to query/manage group membership of a CoAP endpoint as defined in the following subsections.

The group configuration resource and its sub-resources have a JSON-based content format (as indicated by the "application/coap-group+json" media type). The resource includes zero or more group membership JSON objects in a format as defined in Section 2.7.2.4. A group membership JSON object contains one or more key/value pairs as defined below. It represents a single IP multicast group membership for the CoAP endpoint.

Examples of four different group membership objects are:

```
{ "n": "All-Devices.floor1.west.bldg6.example.com",
  "a": "[ff15::4200:f7fe:ed37:abcd]:4567" }

{ "n": "sensors.floor2.east.bldg6.example.com" }

{ "n": "coap-test",
  "a": "224.0.1.187:56789" }

{ "a": "[ff15::c0a7:15:c00l]" }
```

The OPTIONAL "n" key/value pair stands for "name" and identifies the group with a hostname, for example a FQDN. The OPTIONAL "a" key/value pair specifies the IP multicast address (and optionally the port number) of the group. It contains an IPv4 address (in dotted decimal notation) or an IPv6 address. The following ABNF rule can be used for parsing the address, referring to the definitions in Section 6 of [RFC7252] and [RFC3986].

```
   group-address = IPv4address [ ":" port ]
                 / "[" IPv6address "]" [":" port ]
```

If the port number is not provided then it is assumed to be the default CoAP port (5683). In a response, the "a" key/value pair SHOULD be included if the IP address is known at the time of generating the response, and MUST NOT be included if unknown. If the "a" value is not provided in a request, the "n" value in the same group membership object SHOULD be a valid hostname with optional port number that can be translated to an IP multicast address via DNS resolution, as follows:

```
   group-name = host [ ":" port ]
```

If the port number is not provided then it is assumed to be the default CoAP port (5683). At least one of the "n"/"a" pairs MUST be given per group object.

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.

2.7.2.2. Creating a new multicast group membership (POST)

```
Method:       POST
URI Template: /{+gp}
Location-URI Template: /{+gp}/{index}
URI Template Variables:
  gp    - Group Configuration Function Set path (mandatory).
  index - Group index, SHOULD be a string of 1 or 2 alphanumerical
    characters. It MUST be generated as locally unique.
```

```
Example:
  Req: POST /coap-group
       Content-Format: application/coap-group+json
    { "n": "All-Devices.floor1.west.bldg6.example.com",
      "a": "[ff15::4200:f7fe:ed37:abcd]:4567" }
  Res: 2.01 Created
       Location-Path: /coap-group/12
```

For the ’gp’ variable it is recommended to use the path "coap-group" by default. If the "a" key/value pair is given, this takes priority and the "n" pair becomes informational. If only the "n" pair is given, the CoAP endpoint may perform DNS resolution (if supported) to obtain the IP multicast address from the hostname.

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. When a POST payload contains in "a" an IP multicast address to which the endpoint is already subscribed, no change to that subscription is needed.

2.7.2.3. Deleting a single group membership (DELETE)

```
Method:       DELETE
URI Template: {+location}
URI Template Variables:
  location - The Location-Path returned by the CoAP server as a result
    of a successful group creation.

Example:
  Req: DELETE /coap-group/12
  Res: 2.02 Deleted
```

2.7.2.4. Reading all group memberships at once (GET)

A (unicast) GET on the CoAP-group resource returns a JSON object containing multiple keys and values, the keys being group indices and the values the corresponding group objects. Each group object is a group membership JSON object that indicates one IP multicast group membership. So, the group index is used as a JSON key to point to the group membership object, as shown below.

```
Method: GET
URI Template: /{+gp}
URI Template Variables:
  gp - see earlier definition

Example:
  Req: GET /coap-group
  Res: 2.05 Content
       Content-Format: application/coap-group+json
    { "8" :{ "a": "[ff15::4200:f7fe:ed37:14ca]" },
      "11":{ "n": "sensors.floor1.west.bldg6.example.com",
             "a": "[ff15::4200:f7fe:ed37:25cb]" },
      "12":{ "n": "All-Devices.floor1.west.bldg6.example.com",
             "a": "[ff15::4200:f7fe:ed37:abcd]:4567" }
    }
```

Note: the returned IPv6 address may be a different string from the one originally submitted in group membership creation, due to different choices in IPv6 string representation formatting that may be allowed for the same address (see [RFC5952]).

2.7.2.5. Reading a single group membership (GET)

```
Method: GET
URI Template 1: {+location}
URI Template 2: /{+gp}/{index}
URI Template Variables:
  location, gp, index - see earlier definitions

Example:
  Req: GET /coap-group/12
  Res: 2.05 Content
       Content-Format: application/coap-group+json
    {"n": "All-Devices.floor1.west.bldg6.example.com",
     "a": "[ff15::4200:f7fe:ed37:abcd]:4567"}
```

2.7.2.6. Creating/updating all group memberships at once (PUT)

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. This operation SHOULD only be used to delete or update group membership objects for which the CoAP client, invoking this operation, is responsible. The responsibility is based on application level knowledge. For example, a commissioning tool will be responsible for any group membership objects that it created.

```
Method: PUT
URI Template: /{+gp}
URI Template Variables:
  gp - see earlier definition

Example: (replacing all existing group memberships with two new groups)
  Req: PUT /coap-group
       Content-Format: application/coap-group+json
    { "1":{ "a": "[ff15::4200:f7fe:ed37:1234]" },
      "2":{ "a": "[ff15::4200:f7fe:ed37:5678]" }
    }
  Res: 2.04 Changed

Example: (clearing all group memberships at once)
  Req: PUT /coap-group
       Content-Format: application/coap-group+json
    {}
  Res: 2.04 Changed
```

After a successful PUT on the Group configuration resource, the endpoint MUST effect registration to any new IP multicast group(s) and de-registration from any previous IP multicast group(s), i.e. not

anymore present in the new memberships, as soon as possible. Also it MUST take into account the group indices present in the new resource during the generation of any new unique group indices in the future.

2.7.2.7. Updating a single group membership (PUT)

A (unicast) PUT with a group membership JSON object will replace an existing group membership in the endpoint with the new one defined in the PUT request. This can be used to update the group membership.

```
Method: PUT
URI Template 1: {+location}
URI Template 2: /{+gp}/{index}
URI Template Variables:
  location, gp, index - see earlier definitions

Example: (group name and IP multicast port change)
  Req: PUT /coap-group/12
       Content-Format: application/coap-group+json
    {"n": "All-My-Devices.floor1.west.bldg6.example.com",
     "a": "[ff15::4200:f7fe:ed37:abcd]"}
  Res: 2.04 Changed
```

After a successful PUT on the Group configuration resource, the endpoint MUST effect registration to any new IP multicast group(s) and de-registration from any previous IP multicast group(s), i.e. not anymore present in the new membership, as soon as possible.

2.8. Request Acceptance and Response Suppression Rules

CoAP [RFC7252] and CoRE Link Format [RFC6690] define normative behaviors for:

1. IP multicast request acceptance - in which cases a CoAP request is accepted and executed, and when not.

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

A CoAP response differs from a CoAP ACK; ACKs are never sent by servers in response to an IP multicast CoAP request. This section first summarizes these normative behaviors and then presents additional guidelines for response suppression. Also a number of IP 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 IP multicast by making use of APIs such as IPV6_RECVPKTINFO [RFC3542].

For IP multicast request acceptance, the REQUIRED behaviors are:

- A server SHOULD NOT accept an IP multicast request that cannot be "authenticated" in some way (cryptographically or by some multicast boundary limiting the potential sources) [RFC7252]. See Section 5.3 for examples of multicast boundary limiting methods.

- A server SHOULD NOT accept an IP 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 an IP multicast request that acts on a specific resource for which IP multicast support is not required. (Note that for the resource "/.well-known/core", IP multicast support is required if "multicast resource discovery" is supported as specified in section 1.2.1 of [RFC6690]). Implementers are advised to disable IP multicast support by default on any other resource, until explicitly enabled by an application or by configuration.)

- Otherwise accept the IP multicast request.

For IP multicast response suppression, the REQUIRED behaviors are:

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

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

- Otherwise respond to the IP 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 IP 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 CoAP 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 group communication 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. Note that in this case the client implements a "reliable group communication" (as defined in Section 1.3) function using additional, non-standardized functions above the CoAP layer.

- IP multicast firmware update by sending blocks of data: Suppress all 2.xx and 5.xx responses. After having sent all IP 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 group URI query: Suppress all 2.05 responses with empty payload (i.e., if a query produces no matching results).

2.9. Congestion Control

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

CoAP [RFC7252] reduces IP multicast-specific congestion risks through the following measures:

- A server MAY choose not to respond to an IP multicast request if there’s nothing useful to respond (e.g., error or empty response). See Section 2.8 for more detailed guidelines on response suppression.

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

- An IP multicast request MUST be Non-confirmable.

- A response to an IP multicast request SHOULD be Non-confirmable (Section 5.2.3 of [RFC7252]).

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

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

- A server in an LLN should only support group communication GET for resources that are small. For example, the payload of the response is limited to approximately 5% of the IP Maximum Transmit Unit (MTU) size so it fits into a single link-layer frame in case 6LoWPAN [RFC4944] is used.

- A server can minimize the payload length in response to a group communication 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].

- A server can also minimize the payload length of a response to a group communication 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 an IP multicast CoAP request to "/.well-known/core" SHOULD support core-block.

- A client should use CoAP group communication with the smallest possible IP multicast scope that fulfills the application needs. As an example, site-local scope is always preferred over global scope IP multicast if this fulfills the application needs.

More guidelines specific to use of CoAP in 6LoWPAN networks [RFC4944] are given in Section 4.5.

2.10. Proxy Operation

CoAP [RFC7252] allows a client to request a forward-proxy to process its CoAP request. For this purpose the client either specifies the request group URI as a string in the Proxy-URI option, or it specifies the Proxy-Scheme option with the group 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 CoAP group communication request made in this manner through a proxy.

A proxy may buffer all the individual (unicast) responses to a CoAP group communication request and then send back only a single (aggregated) response to the client. However there are some issues with this aggregation approach:

- Aggregation of (unicast) responses to a CoAP 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. Also the proxy does not know how long to wait before deciding to send back the aggregated response to the client.

- 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 [RFC7252], the proxy would simply forward all the individual (unicast) responses to a CoAP 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 group URI 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 an IP 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 IP 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.

2.11. Exceptions

CoAP group communication using IP multicast offers improved network efficiency and latency amongst other benefits. However, group communication may not always be implementable in a given network. The primary reason for this will be that IP multicast is not (fully) supported in the network.

For example, if only the RPL protocol [RFC6550] is used in a network with its optional multicast support disabled, there will be no IP multicast routing at all. The only multicast that works in this case is link-local IPv6 multicast. This implies that any CoAP group communication request will be delivered to nodes on the local link only, regardless of the scope value used in the IPv6 destination address.

3. Use Cases and Corresponding Protocol Flows

3.1. Introduction

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

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

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

3.2. Network Configuration

To illustrate the use cases we define two IPv6 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.
- 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.

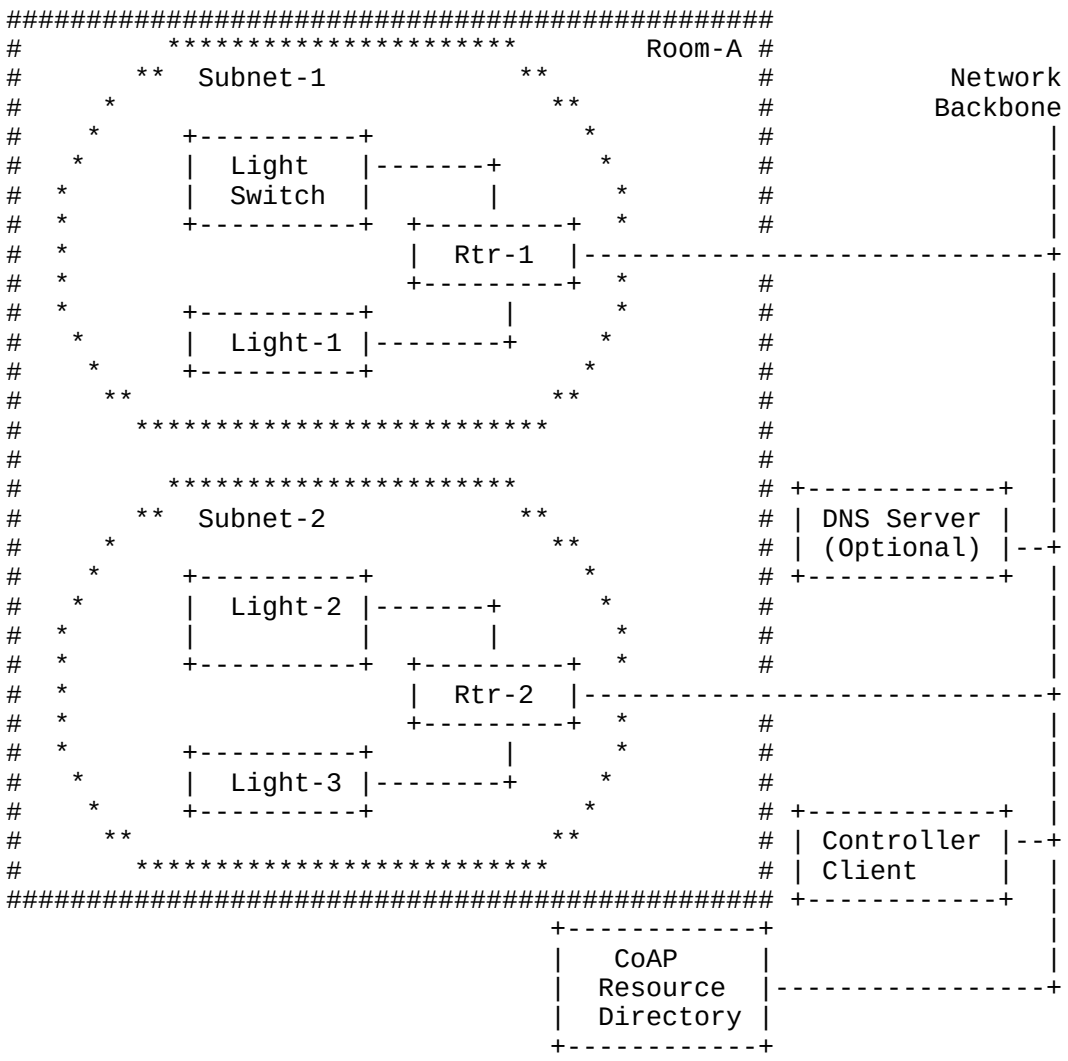

Figure 1: Network Topology of a Large Room (Room-A)

3.3. 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:

o 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 group communication GET request (with the "/.well-known/core?rt=core.rd" request URI) to the site-local "All CoAP Nodes" multicast address (FF05::FD).

- This multicast message will then go to each node in subnet-2. Rtr-2 will then forward it into to the Network Backbone where it will be received by the CoAP RD. All other nodes in subnet-2 will ignore the group communication GET request 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 installed.

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. (See [I-D.ietf-core-resource-directory] for more details).

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

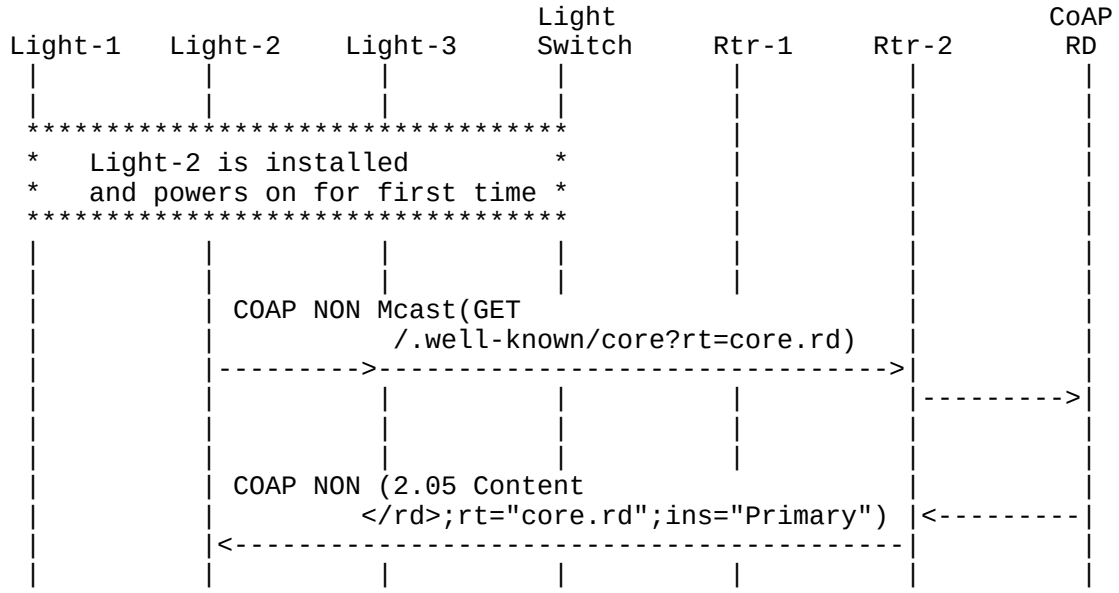

Figure 2: Resource Directory Discovery via Multicast Request

3.4. 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 IP multicast CoAP requests related to lighting control. (See Section 2.8 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 IP 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 IP 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. In case a CoAP message is delivered multiple times to a Light, the subsequent CoAP messages can be filtered out as duplicates, based on the CoAP Message ID.

Reliability of IP 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 IP multicast forwarding/routing will be of sufficient quality to provide on average a very high probability of packet delivery to all CoAP endpoints in an IP 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 IP 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 an IP multicast CoAP request.

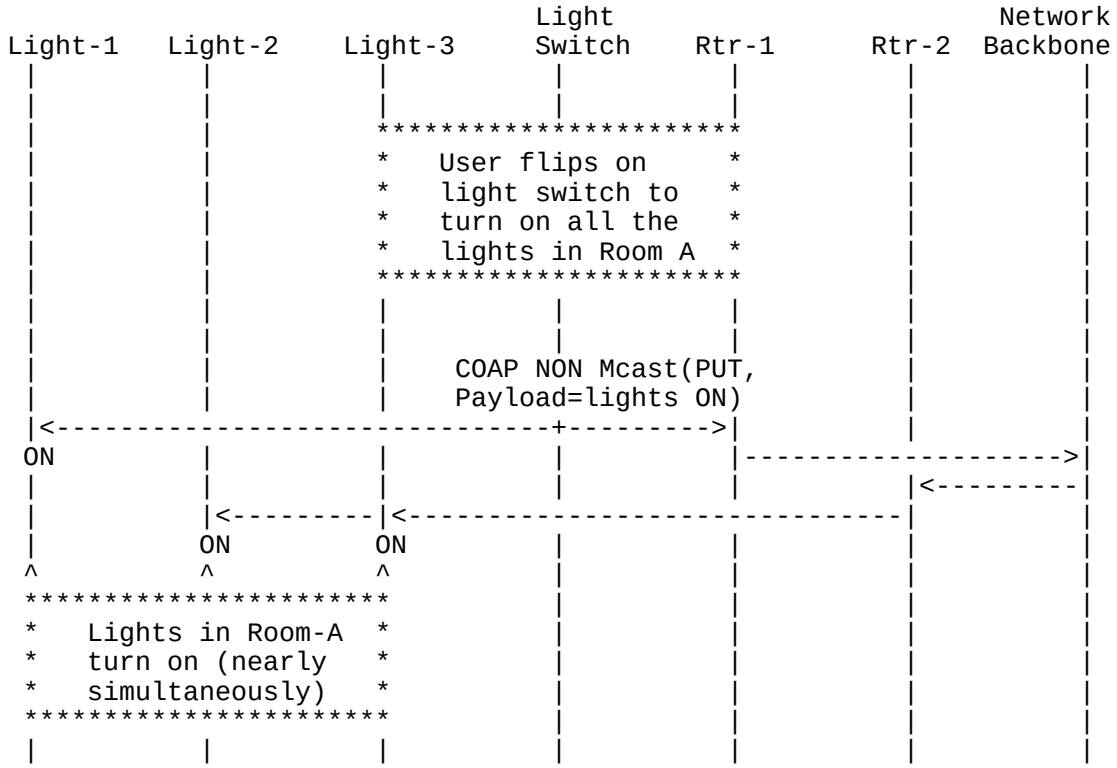

Figure 3: Light Switch Sends Multicast Control Message

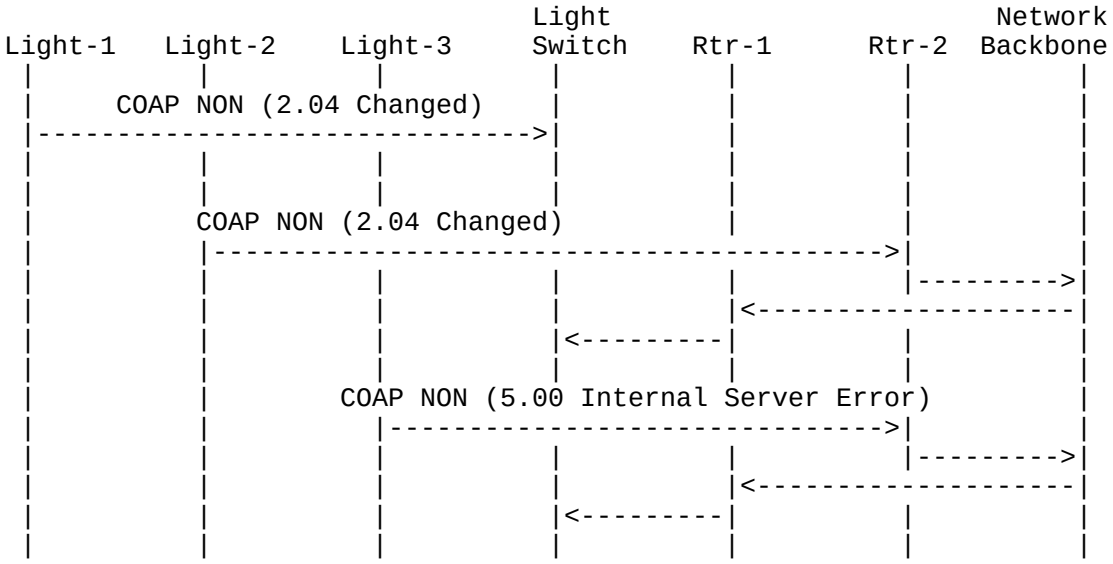

Figure 4: Lights (Optionally) Respond to Multicast CoAP Request

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 group communication request to turn on all lights in Room-A. Every Light sends back a CoAP response to the Controller after being turned on.

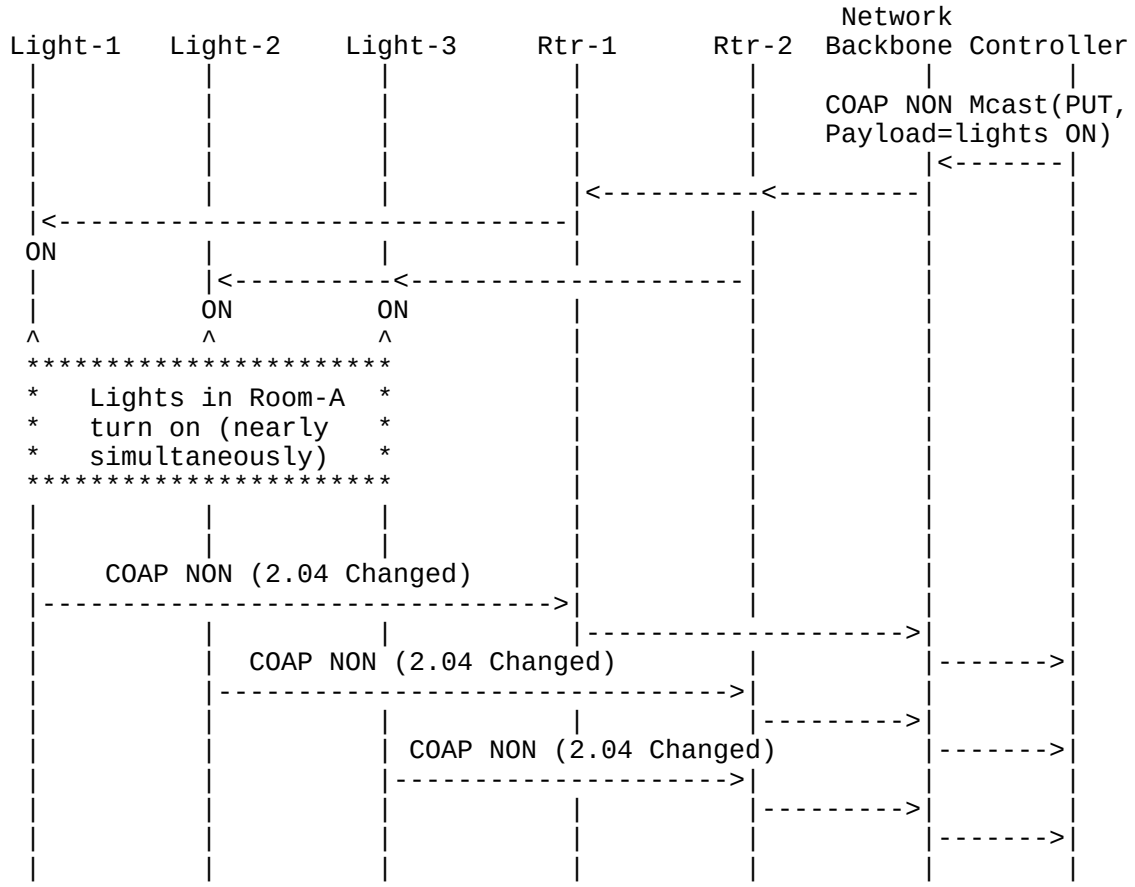

Figure 5: Controller On Backbone Sends Multicast Control Message

3.5. 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. In current 6LoWPAN implementations, MLD is however not supported.

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 IP multicast traffic destined to Room-A-Lights onto their connected subnet. Hence, no manual Network Configuration of routers, as previously indicated in Section 3.4, is needed anymore.

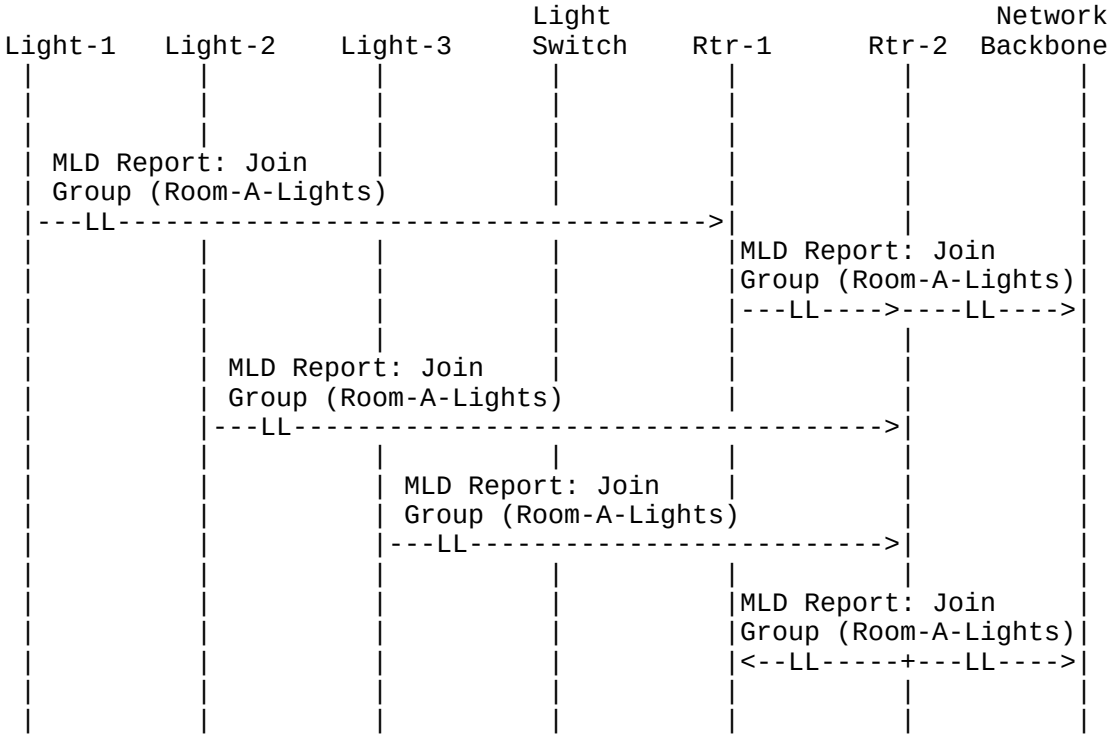

Figure 6: Joining Lighting Groups Using MLD

3.6. 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 IP 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 IP multicast address. For each light in the list the tool learns the IP address of the light and instructs the RD with three (unicast) POST commands to store the endpoints associated with the three lights as prescribed by the RD specification [I-D.ietf-core-resource-directory]. Finally the commissioning tool defines the group in the RD to contain these three endpoints. Also the commissioning tool writes the IP multicast address in the Light endpoints with, for example, the (unicast) POST command discussed in Section 2.7.2.2.

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

4. Deployment Guidelines

This section provides guidelines how IP multicast based CoAP group communication can be deployed in various network configurations.

4.1. Target Network Topologies

CoAP group communication can be deployed in various network topologies. First, the target network may be a traditional IP network, or a LLN such as a 6LoWPAN network, or consist of mixed traditional/constrained 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. To enable all the above target network topologies, an implementation of CoAP group communication needs to allow:

1. Routing/forwarding of IP multicast packets over multiple hops

2. Routing/forwarding of IP multicast packets over subnet boundaries between traditional and constrained (e.g. LLN) networks.

The remainder of this section discusses solutions to enable both features.

4.2. Networks Using the MLD Protocol

CoAP nodes that are IP hosts (i.e., not IP routers) are generally unaware of the specific IP 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 IP multicast routers which IP multicast traffic it wants to receive.

The Multicast Listener Discovery (MLD) protocol [RFC3810] (see Appendix A) is the standard IPv6 method to achieve this; therefore this approach should be used on traditional IP networks. CoAP server nodes would then act in the role of MLD Multicast Address Listener.

The guidelines from [RFC6636] on tuning of MLD for mobile and wireless networks may be useful when implementing MLD in LLNs. However, on LLNs and 6LoWPAN networks the use of MLD may not be feasible at all due to constraints on code size, memory, or network capacity.

4.3. Networks Using RPL Multicast Without MLD

It is assumed in this section that the MLD protocol is not implemented in a network, for example due to resource constraints. The RPL routing protocol (see Section 12 of [RFC6550]) defines the advertisement of IP multicast destinations using DAO messages and routing of multicast IPv6 packets based on this. It requires the RPL Mode of Operation (MOP) to be 3 (Storing Mode with multicast support).

Hence, RPL DAO can be used by CoAP nodes that are RPL Routers, or are RPL Leaf Nodes, to advertise IP multicast group membership to parent routers. Then, the RPL protocol is used to route IP multicast CoAP requests over multiple hops to the correct CoAP servers.

The same DAO mechanism can be used 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 is useful because the edge router then learns which IP multicast traffic it needs to pass through from the backbone network into the LLN subnet. In 6LoWPAN networks, such selective "filtering" helps to avoid congestion of a 6LoWPAN subnet by IP multicast traffic from the traditional backbone IP network.

4.4. Networks Using MPL Forwarding Without MLD

The MPL forwarding protocol [I-D.ietf-roll-trickle-mcast] can be used for propagation of IPv6 multicast packets to all MPL Forwarders within a predefined network domain, over multiple hops. MPL is designed to work in LLNs. In this section it is again assumed that Multicast Listener Discovery (MLD) is not implemented in the network, for example due to resource limitations in an LLN.

The purpose of MPL is to let a predefined group of Forwarders collectively work towards the goal of distributing an IPv6 multicast packet throughout an MPL Domain. (A Forwarder node may be associated to multiple MPL Domains at the same time.) So it would appear there is no need for CoAP servers to advertise their multicast group membership, since any IP multicast packet that enters the MPL Domain is distributed to all MPL Forwarders without regard to what multicast addresses the individual nodes are listening to.

However, if an IP multicast request originates just outside the MPL Domain, the request will not be propagated by MPL. An example of such a case is the network topology of Figure 1 where the Subnets are 6LoWPAN subnets and per 6LoWPAN subnet one Realm-Local ([I-D.droms-6man-multicast-scopes]) MPL Domain is defined. The backbone network in this case is not part of any MPL Domain.

This situation can become a problem in building control use cases. For example, when the Controller Client needs to send a single IP multicast request to the group Room-A-Lights. By default, the request would be blocked by Rtr-1 and by Rtr-2, and not enter the Realm-Local MPL Domains associated to Subnet-1 and Subnet-2. The reason is that Rtr-1 and Rtr-2 do not have the knowledge that devices in Subnet-1/2 want to listen for IP packets destined to IP multicast group Room-A-Lights.

To solve the above issue, the following solutions could be applied:

1. Extend the MPL Domain. E.g. in above example, include the Network Backbone to be part of each of the two MPL Domains. Or in above example, create just a single MPL Domain that includes both 6LoWPAN subnets plus the backbone link, which is possible since MPL is not tied to a single link-layer technology.

2. Manual configuration of edge router(s) as MPL Seed(s) for specific IP multicast traffic. E.g. in above example, first configure Rtr-1 and Rtr-2 to act as MLD Address Listeners for the Room-A-Lights IP multicast group. This step allows any (other) routers on the backbone to learn that at least one node on the backbone link is interested to receive any IP multicast traffic

to Room-A-Lights. Second, configure both routers to "inject" any IP multicast packets destined to group Room-A-Lights into the (Realm-Local) MPL Domain that is associated to that router. Third, configure both routers to propagate any IPv6 multicast packets originating from within their associated MPL Domain to the backbone, if at least one node on the backbone has indicated interest to receive such IPv6 packets (for which MLD is used on the backbone).

3. Use an additional protocol/mechanism for injection of IP multicast traffic from outside an MPL Domain into that MPL Domain, based on IP multicast group subscriptions of Forwarders within the MPL Domain. Such protocol is currently not defined in [I-D.ietf-roll-trickle-mcast].

Concluding, MPL can be used directly in case all sources of IP multicast CoAP requests (CoAP clients) and also all the destinations (CoAP servers) are inside a single MPL Domain. Then, each source node acts as an MPL Seed. In all other cases, MPL can only be used with additional protocols and/or configuration on how IP multicast packets can be injected from outside into an MPL Domain.

4.5. 6LoWPAN Specific Guidelines for the 6LBR

To support multi-subnet 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 (see Appendix A) on the backbone link.

5. 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.

5.1. Security Configuration

As defined in [RFC7252], CoAP group communication based on IP multicast:

- Will operate in CoAP NoSec (No Security) mode, until a future group security solution is developed (see also Section 5.3.3).
- Will use "coap" scheme mode. The "coaps" scheme should only be used when a future group security solution is developed (see also Section 5.3.3).

5.2. Threats

Essentially the above configuration means that there is currently 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 [RFC7252] for IP multicast.

5.3. Threat Mitigation

The [RFC7252] identifies various threat mitigation techniques for CoAP group communication. 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.

5.3.1. WiFi Scenario

In a home automation scenario (using WiFi), the WiFi encryption should be enabled to prevent rogue nodes from joining. The Customer Premise Equipment (CPE) that enables access to the Internet should also have its IP multicast filters set so that it enforces multicast scope boundaries to isolate local multicast groups from the rest of the Internet (e.g., as per [RFC6092]). In addition, the scope of the IP multicast should be set to be site-local or smaller scope. For site-local scope, the CPE will be an appropriate multicast scope boundary point.

5.3.2. 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.

5.3.3. Future Evolution

In the future, to further mitigate the threats, the developing approach for DTLS-based IP multicast security for CoAP communications (see [I-D.keoh-dice-multicast-security]) or similar approaches should be considered. This will allow introduction of a secure mode of CoAP group communication, and use of the "coaps" scheme for that purpose.

5.4. Pervasive Monitoring Considerations

A key additional threat consideration for group communication is pointed to by [RFC7258] which warns of the dangers of pervasive monitoring. CoAP group communication which is built on top of IP multicast should pay particular heed to these dangers. This is because IP multicast is easier to intercept (e.g. and to secretly record) compared to unicast traffic. Also, CoAP traffic is meant for the Internet of Things. This means that CoAP traffic is often used for the control and monitoring of critical infrastructure (e.g. lights, alarms, etc.) which may be prime targets for attack.

For example, an attacker may attempt to record all the CoAP traffic going over the smart grid (i.e. networked electrical utility) of a country and try to determine critical control nodes for further attacks. CoAP multicast traffic is inherently more vulnerable (compared to a unicast packet) as the same packet may be replicated over many links so there is a much higher probability of it getting captured by a pervasive monitoring system.

One useful mitigation to pervasive monitoring is to restrict the scope of the IP multicast to the minimal scope that fulfills the application need. Thus, for example, site-local IP multicast scope is always preferred over global scope IP multicast if this fulfills the application needs. This approach has the added advantage that it coincides with the guidelines for minimizing congestion control (see Section 2.9.

In the future, even if all the CoAP multicast traffic is encrypted (e.g. [I-D.keoh-dice-multicast-security]), an attacker may still attempt to capture the traffic and perform an off-line attack. Though of course having the multicast traffic protected is always desirable as it significantly raises the cost to an attacker (e.g. to break the encryption) versus unprotected multicast traffic.

6. IANA Considerations

6.1. 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 2.7.2.

6.2. 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 [RFC7252].

Type name: application

Subtype name: coap-group+json

Required parameters: None

Optional parameters: None

Encoding considerations: 8bit UTF-8.

JSON to be represented using UTF-8 which is 8bit compatible (and most efficient for resource constrained implementations).

Security considerations:

Denial of Service attacks could be performed by constantly (re-)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 a form of authorization (making use of DTLS-secured CoAP) be used such that only authorized controllers are allowed by an endpoint 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 2.7.2.

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

7. Acknowledgements

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

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.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3433] Bierman, A., Romascanu, D., and K. Norseth, "Entity Sensor Management Information Base", RFC 3433, December 2002.

[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.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[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.

[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.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

[RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.

[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.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, May 2014.

[RFC7320] Nottingham, M., "URI Design and Ownership", BCP 190, RFC 7320, July 2014.

8.2. Informative References

[RFC1033] Lottor, M., "Domain administrators operations guide", RFC 1033, November 1987.

[I-D.ietf-core-block]
Bormann, C. and Z. Shelby, "Blockwise transfers in CoAP", draft-ietf-core-block-15 (work in progress), July 2014.

[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-09 (work in progress), April 2014.

[I-D.keoh-dice-multicast-security]
Keoh, S., Kumar, S., Garcia-Morchon, O., Dijk, E., and A. Rahman, "DTLS-based Multicast Security in Constrained Environments", draft-keoh-dice-multicast-security-08 (work in progress), July 2014.

[I-D.ietf-core-resource-directory]
Shelby, Z., Bormann, C., and S. Krco, "CoRE Resource Directory", draft-ietf-core-resource-directory-01 (work in progress), December 2013.

[I-D.droms-6man-multicast-scopes]
Droms, R., "IPv6 Multicast Address Scopes", draft-droms-6man-multicast-scopes-02 (work in progress), July 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 IP 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 equivalent IGMP [RFC3376]) is today the method of choice used by an (IP multicast enabled) router to discover the presence of IP multicast listeners on directly attached links, and to discover which IP multicast addresses are of interest to those listening nodes. MLD was specifically designed to cope with fairly dynamic situations in which IP multicast listeners may join and leave at any time.

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

Appendix B. Change Log

[Note to RFC Editor: Please remove this section before publication.]

Changes from ietf-19 to ietf-20:

- Replaced obsolete reference [RFC 2616] by [RFC 7230].
- Replaced outdated reference draft-ietf-appsawg-uri-get-off-my-lawn by [RFC 7320] and moved to Normative reference.
- Replaced outdated reference draft-ietf-core-coap by [RFC 7252].
- Moved [RFC 1033] to Informative reference.
- Updated to latest revision numbers for informative draft references by regenerating file through xml2rfc tool.
- Re-ran IETF spell check tool and corrected some minor spelling errors.
- Various minor editorial updates.

Changes from ietf-18 to ietf-19:

- Added guideline on Token value re-use in section 2.5.
- Updated section 5.1 (Security Configuration) and 5.3.3 (Future Security Evolution) to point to latest security developments happening in DICE WG for support of group security.
- Added Pervasive Monitoring considerations in section 5.4.
- Various editorial updates for improved readability.

Changes from ietf-17 to ietf-18:

- Extensive editorial updates based on WGLC comments by Thomas Fossati and Gengyu Wei.
- Addressed ticket #361: Added text for single membership PUT section 2.7.2.7 (Updating a single group membership (PUT)).
- Addressed ticket #360: Added text for server duties upon all-at-once PUT section 2.7.2.6 (Creating/updating all group memberships at once (PUT)).
- Addressed ticket #359: Fixed requirements text for Section 2.7.2.2 (Creating a new multicast group membership (POST)).
- Addressed ticket #358: Fixed requirements text for Section 2.7.2.1 (CoAP-Group Resource Type and Media Type).
- Addressed ticket #357: Added that "IPv6 addresses of other scopes MAY be enabled" in section 2.2 (Group Definition and Naming).
- Various editorial updates for improved readability.

Changes from ietf-16 to ietf-17:

- Added guidelines on joining of IPv6/IPv4 "All CoAP Nodes" multicast addresses (#356).
- Added MUST support default port in case multicast discovery is available.
- In section 2.1 (IP Multicast Background), clarified that IP multicast is not guaranteed and referenced a definition of Reliable Group Communication (#355).

- Added section 2.5 (Messages and Responses) to clarify how responses are identified and how Token/MID are used in multicast CoAP.

- In section 2.6.2 (RESTful Interface for Configuring Group Memberships), clarified that group management interface is an optional approach for dynamic commissioning and that other approaches can also be used if desired.

- Updated section 2.6.2 (RESTful Interface for Configuring Group Memberships) to allow deletion of individual group memberships (#354).

- Various editorial updates based on comments by Peter van der Stok. Removed reference to expired draft-vanderstok-core-dna at request of its author.

- Various editorial updates for improved readability.

Changes from ietf-15 to ietf-16:

- In section 2.6.2, changed DELETE in group management interface to a PUT with empty JSON array to clear the list (#345).

- In section 2.6.2, aligned the syntax for IP addresses to follow RFC 3986 URI syntax, which is also used by coap-18. This allows re-use of the parsing code for CoAP URIs for this purpose (#342).

- Addressed some more editorial comments provided by Carsten Bormann in preparation for WGLC.

- Various editorial updates for improved readability.

Changes from ietf-14 to ietf-15:

- In section 2.2, provided guidance on how implementers should parse URIs for group communication (#339).

- In section 2.6.2.1, specified that for group membership configuration interface the "ip" (i.e. "a" parameter) key/value is not required when it is unknown (#338).

- In section 2.6.2.1, specified that for group membership configuration interface the port configuration be defaulted to standard CoAP port 5683, and if not default then should follow standard notation (#340).

- In section 2.6.2.1, specified that notation of IP address in group membership configuration interface should follow standard notation (#342).

- In section 6.2, "coap-group+json" Media Type encoding simplified to just support UTF-8 (and not UTF-16 and UTF-32) (#344).

- Various editorial updates for improved readability.

Changes from ietf-13 to ietf-14:

- Update to address final editorial comments from the Chair's review (by Carsten Bormann) of the draft. This included restructuring of Section 2.6 (Configuring Group Memberships) and Section 4 (Deployment Guidelines) to make it easier to read. Also various other editorial changes.

- Changed "ip" field to "a" in Section 2.6 (#337)

Changes from ietf-12 to ietf-13:

- Extensive editorial updates due to comments from the Chair's review (by Carsten Bormann) of the draft. The best way to see the changes will be to do a -Diff with Rev. 12.

- The technical comments from the Chair's review will be addressed in a future revision after tickets are generated and the solutions are agreed to on the WG E-mail list.

Changes from ietf-11 to ietf-12:

- Removed reference to "CoAP Ping" in Section 3.5 (Group Member Discovery) and replaced it with the more efficient support of discovery of groups and group members via the CORE RD as suggested by Zach Shelby.

- Various editorial updates for improved readability.

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 [RFC 3542]".
- 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 draft-ietf-core-coap in section 2.2 (Scope).

- Removed IPSec related requirement, as IPSec is not part of draft-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 draft-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 draft-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