Group Communication for the Constrained Application Protocol (CoAP)

Abstract

   The Constrained Application Protocol (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 specification defines how CoAP should be used in a group
   communication context. An approach for using CoAP on top of IP
   multicast is detailed based on existing CoAP functionality as well as
   new features introduced in this specification. 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

Copyright Notice

Table of Contents

1. Introduction

1.1. Background

   CoAP is a web transfer protocol based on Representational State
   Transfer (REST) for resource constrained devices operating in an IP
   network [RFC7252]. CoAP has many similarities to HTTP [RFC7230] but
   also 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 preconfigured 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 base protocol aspects of sending CoAP requests on top of IP
   multicast and processing the (unicast IP) responses are given in
   Section 8 of [RFC7252]. To provide a more complete CoAP group
   communication functionality, this specification introduces new CoAP

processing functionality (e.g., new rules for reuse of Token values, request suppression, and proxy operation) and a new management interface for RESTful group membership configuration.

CoAP group communication will run in the Any Source Multicast (ASM) mode [RFC5110] of IP multicast operation. This means that there is no restriction on the source node that sends (originates) the CoAP messages to the IP multicast group. For example, the source node may or may not be part of the IP multicast group. Also, there is no restriction on the number of source nodes.

While Section 9.1 of [RFC7252] supports various modes of security based on Datagram Transport Layer Security (DTLS) for CoAP over unicast IP, it does not specify any security modes for CoAP over IP multicast. That is, it is assumed per [RFC7252] that CoAP over IP multicast is not encrypted, nor authenticated, nor access controlled. This document assumes the same security model (see Section 5.1). However, there are several promising security approaches being developed that should be considered in the future for protecting CoAP group communications (see Section 5.3.3).

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] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings and are not to be interpreted as [RFC2119] key words.

Note that this document refers back to other RFCs, and especially [RFC7252], to help explain overall CoAP group communication features. However, use of [RFC2119] key words is reserved for new CoAP functionality introduced by this specification.

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 that 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. An example of a reliable group communication protocol is
   [RFC5740].

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 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 (LL) 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
the Multicast Protocol for Low-Power and Lossy Networks (MPL)
[MCAST-MPL]. 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 an 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). Also, a multicast
forwarding proxy node [RFC4605] may be required.

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 (Section 12.8 of [RFC7252]) 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 (Section 6 of [RFC7252]).

Note: A group URI is needed to initiate CoAP group communications.
For CoAP client implementations, it is recommended to use the URI
decomposition method of Section 6.4 of [RFC7252] in such a 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. (This is because DNS infrastructure may not
be deployed in many constrained network deployments.) 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,
                                              floor 1, 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
troubleshooting 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, usually on the CoAP default UDP
port, 5683. If the group uses a specified non-default UDP port, be
careful to ensure that all group members are configured to use that
same port.

Different ports for the same IP multicast address are preferably not
used to specify different CoAP groups. If disjoint groups share the
same IP multicast address, then all the devices interested in one
group will accept IP traffic also for the other disjoint groups, only
to ultimately discard the traffic higher in their IP stack (based on
UDP port discrimination).

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 the below options:

1. A preconfigured port number.

2. If the client is configured to use service discovery including URI and port discovery, it uses the 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. Regardless of the method of selecting the port number, the same port MUST be used across all CoAP servers in a group and across all CoAP clients performing the group requests.

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

1. Preconfigured group URI paths, if available. Implementers are free to define the paths as they see fit. However, note that [RFC7320] prescribes that a specification must not constrain or define the structure or semantics for any path component. So for this reason, a predefined URI path is not specified in this document and also must not be provided in other specifications.

2. If the client is configured to use default Constrained RESTful Environments (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 that 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.

   However, when the URI path is selected, the same path MUST be used
   across all CoAP servers in a group and all CoAP clients performing
   the group requests.

2.4. RESTful Methods

   Group communication most often uses the idempotent CoAP methods GET
   and PUT. The idempotent method DELETE can also be used. The non-
   idempotent CoAP method POST may only 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. For example, a client
   may resend a multicast POST request for additional reliability. Some
   servers will receive the request two times while others may receive
   it only once. For idempotent methods, all these servers will be in
   the same state while for POST, this is not guaranteed; so, the
   resource POST operation must be specifically designed to take message
   loss into account.

2.5. Request and Response Model

   All CoAP requests that are sent via IP multicast must be Non-
   confirmable (Section 8.1 of [RFC7252]). The Message ID in an IP
   multicast CoAP message is used for optional message deduplication as
   detailed in Section 4.5 of [RFC7252].

   A server optionally sends 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 2.7.

   A CoAP request sent over IP multicast and any unicast response it
   causes must take into account the congestion control rules defined in
   Section 2.8.

   The CoAP client can distinguish the origin of multiple server
   responses by the 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
reuse of Token values, compared to the unicast case. In the unicast
case, receiving a response effectively frees up its Token value for
reuse 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 reuse. Reusing a Token value too early could lead
to incorrect response/request matching in the client and would be a
protocol error. Therefore, the time between reuse of Token values
used in multicast requests MUST be greater than:

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]. CoAP does not define
a time limit for the server response delay. Using the default CoAP
parameters, the Token reuse time MUST be greater than 250 seconds
plus MAX_SERVER_RESPONSE_DELAY. A preferred solution to meet this
requirement is to generate a new unique Token for every multicast
request, such that a Token value is never reused. If a client has to
reuse Token values for some reason, and also
MAX_SERVER_RESPONSE_DELAY is unknown, then using
MAX_SERVER_RESPONSE_DELAY = 250 seconds is a reasonable guideline.
The time between Token reuses is in that case set to a value greater
than 500 seconds.

2.6. Membership Configuration

2.6.1. Background

2.6.1.1. Member Discovery

CoAP groups, and the membership of these groups, can be discovered
via the lookup interfaces in the Resource Directory (RD) defined in
[CoRE-RD]. This discovery interface is not required to invoke CoAP
group communications. However, it is a potential complementary
interface useful for overall management of CoAP groups. Other
methods to discover groups (e.g., proprietary management systems) can
also be used. An example of doing some of the RD-based lookups is
given in Section 3.6.

2.6.1.2. Configuring Members

   The group membership of a CoAP endpoint may be configured in one of
   the following ways. First, the group membership may be preconfigured
   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 preconfigured group information may be either
   an IP multicast address or a hostname (FQDN) that 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-based Service Discovery (DNS-SD) and RD
   [CoRE-RD].

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

2.6.2. 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 will use
   unicast CoAP methods (GET/PUT/POST/DELETE). This interface is a
   method of configuring group information in individual endpoints.

   Also, a form of authorization (preferably making use of unicast DTLS-
   secured CoAP per Section 9.1 of [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
   alternative approaches may support a subset or superset 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 the approach of
   [RFC3433]).

2.6.2.1. 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=) Link Target
   Attribute "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 content
   format based on JavaScript Object Notation (JSON) (as indicated by
   the "application/coap-group+json" media type). The resource includes
   zero or more group membership JSON objects [RFC7159] in a format as
   defined in Section 2.6.2.4. A group membership JSON object contains
   one or more key/value pairs as defined below, and represents a single
   IP multicast group membership for the CoAP endpoint. Each key/value
   pair is encoded as a member of the JSON object, where the key is the
   member name and the value is the member's value.

   Examples of four different group membership objects are as follows:

       { "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:c001]" }

   The OPTIONAL "n" key/value pair stands for "name" and identifies the
   group with a hostname (and optionally the port number), for example,
   an 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 3.2.2 of [RFC3986] that are also used in
   the base CoAP (Section 6 of [RFC7252].

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

   In any group membership object, if the IP address is known when the
   object is created, it is included in the "a" key/value pair. If the
   "a" value cannot be provided, the "n" value MUST be included,
   containing a valid hostname with an optional port number that can be
   translated to an IP multicast address via DNS.

      group-name = host [":" port]

   If the port number is not provided, then the endpoint will attempt to
   look up the port number from DNS if it supports a method to do this.
   The possible DNS methods include DNS SRV [RFC2782] or DNS-SD
   [RFC6763]. If port lookup is not supported or not provided by DNS,
   the default CoAP port (5683) is assumed.

   After any change on a Group Configuration resource, the endpoint MUST
   effect registration/deregistration from the corresponding IP
   multicast group(s) by making use of APIs such as IPV6_RECVPKTINFO
   [RFC3542].

2.6.2.2. 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. Index MUST be a string of maximum two (2)
       alphanumeric ASCII characters (case insensitive). It MUST be
       locally unique to the endpoint server. It indexes the particular
       endpoint's list of group memberships.

   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. The "a" key/value pair is always used if it is given.
   The "n" pair is only used when there is no "a" pair. If only the "n"
   pair is given, the CoAP endpoint performs DNS resolution to obtain
   the IP multicast address from the hostname in the "n" pair. If DNS
   resolution is not successful, then the endpoint does not attempt
   joining or listening to any multicast group for this case since the
   IP multicast address is unknown.

   After any change on a Group Configuration resource, the endpoint MUST
   effect registration/deregistration from the corresponding IP
   multicast group(s) by making use of APIs such as IPV6_RECVPKTINFO
   [RFC3542]. When a POST payload contains an "a", an IP multicast
   address to which the endpoint is already subscribed, no change to
   that subscription is needed.

2.6.2.3. 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

2.6.2.4. 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 (member names) are
   group indices, and the values (member values) are the corresponding
   group membership objects. Each group membership object describes one
   IP multicast group membership. If no group memberships are
   configured, then an empty JSON object is returned.

   Method: GET

   URI Template: /{+gp}

   URI Template Variables:

   gp - see Section 2.6.2.2

   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 string will represent the same IPv6
address that was originally submitted in group membership creation,
though it might be a different string because of different choices in
IPv6 string representation formatting that may be allowed for the
same address (see [RFC5952]).

2.6.2.5. Reading a Single Group Membership (GET)

Similar to Section 2.6.2.4, but only a single group membership is
read. If the requested group index does not exist, then a 4.04 Not
Found response is returned.

Method: GET

URI Template 1: {+location}

URI Template 2: /{+gp}/{index}

URI Template Variables:

location - see Section 2.6.2.3

gp, index - see Section 2.6.2.2

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"}

2.6.2.6. 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 MUST 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 Section 2.6.2.2

```
   Example: (replacing all existing group memberships with two new
            group memberships)
     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 deregistration from any previous IP multicast group(s), i.e., not
   any more present in the new memberships. An API such as
   IPV6_RECVPKTINFO [RFC3542] should be used for this purpose. 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.

2.6.2.7. 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 - see Section 2.6.2.3

   gp, index - see Section 2.6.2.2

   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 deregistration from any previous IP multicast group(s), i.e., not
any more present in the new membership. An API such as
IPV6_RECVPKTINFO [RFC3542] should be used for this purpose.

2.7. Request Acceptance and Response Suppression Rules

   CoRE Link Format [RFC6690] and Section 8 of CoAP [RFC7252] define
   behaviors for the following:

   1. IP multicast request acceptance -- in which cases a CoAP request
       is accepted and executed, and when it is 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 it is 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 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 behaviors are as follows:

   o A server should not accept an IP multicast request that cannot be
      "authenticated" in some way (i.e, cryptographically or by some
      multicast boundary limiting the potential sources); see
      Section 11.3 of [RFC7252]. See Section 5.3 for examples of
      multicast boundary limiting methods.

   o 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.

   o 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.

o Otherwise, accept the IP multicast request.

For IP multicast response suppression, the behaviors are as follows:

o A server should not respond to an IP multicast discovery request
   if the filter specified by the request's query string does not
   match.

o A server may choose not to respond to an IP multicast request if
   there's nothing useful to respond back (e.g., error or empty
   response).

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:

o Suppression of all 2.xx success responses;

o Suppression of all 4.xx client errors;

o Suppression of all 5.xx server errors; and

o 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:

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

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

o 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 and 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.

o 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.

      o 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).

2.8. 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:

      o A server may choose not to respond to an IP multicast request if
         there's nothing useful to respond to (e.g., error or empty
         response); see Section 8.2 of [RFC7252]. See Section 2.7 for more
         detailed guidelines on response suppression.

      o A server should limit the support for IP multicast requests to
         specific resources where multicast operation is required
         (Section 11.3 of [RFC7252]).

      o An IP multicast request must be Non-confirmable (Section 8.1 of
         [RFC7252]).

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

      o A server does not respond immediately to an IP multicast request
         and should first wait for a time that is randomly picked within a
         predetermined time interval called the Leisure (Section 8.2 of
         [RFC7252]).

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

      o 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
         IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) (see
         Section 4 of [RFC4944]) is used.

   o 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].

   o 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 [BLOCKWISE-CoAP], 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.

   o 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.
      Similarly, realm-local scope is always preferred over site-local
      scope if this fulfills the application needs.

   More guidelines specific to the use of CoAP in 6LoWPAN networks
   [RFC4919] are given in Section 4.5 of this document.

2.9. Proxy Operation

   CoAP (Section 5.7.2 of [RFC7252]) allows a client to request a
   forward-proxy to process its CoAP request. For this purpose, the
   client specifies either the request group URI as a string in the
   Proxy-URI option or 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:

   o 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.

   o 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 (Section 5.7.2 of [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:

   o 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.

   o 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 the above issues, 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 if all the following conditions are met:

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

   o 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.

   o 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.

2.10. Exceptions

   CoAP group communication using IP multicast offers improved network
   efficiency and latency among 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 RPL [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.

CoAP Observe [OBSERVE-CoAP] is a feature for a client to "observe"
resources (i.e., to retrieve a representation of a resource and keep
this representation updated by the server over a period of time).
CoAP Observe does not support a group communication mode. CoAP
Observe only supports a unicast mode of operation.

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:

o Discovery of RD [CoRE-RD]: discovering the local CoAP RD, which
   contains links to resources stored on other CoAP servers
   [RFC6690].

o 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 as follows:

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:

o A large room (Room-A) with three lights (Light-1, Light-2, Light-
   3) controlled by a light switch (Light Switch). The devices are
   organized into two subnets. In reality, there could be more
   lights (up to several hundreds) but, for clarity, only three are
   shown.

o Light-1 and the light switch are connected to a router (Rtr-1).

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

   o The routers are connected to an IPv6 network backbone (Network
      Backbone) that 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.

   o A CoAP RD is connected to the network backbone.

   o The DNS server (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 the hostname to
      the 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).

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

```
      ##################################################
      #         *********************      Room-A #
      #       **  Subnet-1          **            #          Network
      #     *                        **           #          Backbone
      #    *     +----------+          *          #            |
      #   *      | Light    |-------+    *         #            |
      #  *       | Switch   |       |     *        #            |
      #  *       +----------+  +---------+ *        #            |
      #  *                     | Rtr-1   |-----------------------------+
      #  *                     +---------+  *       #            |
      #  *       +----------+       |      *        #            |
      #   *      | Light-1  |--------+      *        #            |
      #    *     +----------+             *         #            |
      #     **                        **           #            |
      #       ************************              #            |
      #                                            #            |
      #         *********************              # +------------+ |
      #       **  Subnet-2          **             # | DNS Server | |
      #     *                        **            # | (Optional) |--+
      #    *     +----------+          *           # +------------+ |
      #   *      | Light-2  |-------+    *          #            |
      #  *       |          |       |     *         #            |
      #  *       +----------+  +---------+ *         #            |
      #  *                     | Rtr-2   |-----------------------------+
      #  *                     +---------+  *        #            |
      #  *       +----------+       |       *        #            |
      #   *      | Light-3  |--------+       *        # +------------+ |
      #    *     +----------+              *         # | Controller |--+
      #     **                        **            # | Client     | |
      #       ************************               # +------------+ |
      ################################################## +------------+ |
                            +------------+                           |
                            |  CoAP      |                           |
                            | Resource   |---------------------------+
                            | Directory  |
                            +------------+
```

                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.

   o 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).

   o This multicast message will then go to each node in Subnet-2.
      Rtr-2 will then forward it into 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").

   o 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".

   o Note that the flow is shown only for Light-2 for clarity. Similar
      flows will happen for Light-1, Light-3, and 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 [CoRE-RD] 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,
   realm-local, admin-local, or site-local scoped discovery can be done
   this way.

```
                                      Light                                CoAP
   Light-1    Light-2    Light-3     Switch      Rtr-1      Rtr-2          RD
      |          |          |          |           |          |          |
      |          |          |          |           |          |          |
      **********************************           |          |          |
      *    Light-2 is installed        *           |          |          |
      *    and powers on for first time *          |          |          |
      **********************************           |          |          |
      |          |          |          |           |          |          |
      |          |          |          |           |          |          |
      |          | COAP NON Mcast(GET               |          |          |
      |          |          /.well-known/core?rt=core.rd)      |          |
      |          |--------->------------------------------->|             |
      |          |          |          |           |          |--------->|
      |          |          |          |           |          |          |
      |          |          |          |           |          |          |
      |          | COAP NON (2.05 Content           |          |          |
      |          |          </rd>;rt="core.rd";ins="Primary") |<---------|
      |          |<----------------------------------------|             |
      |          |          |          |           |          |          |
```
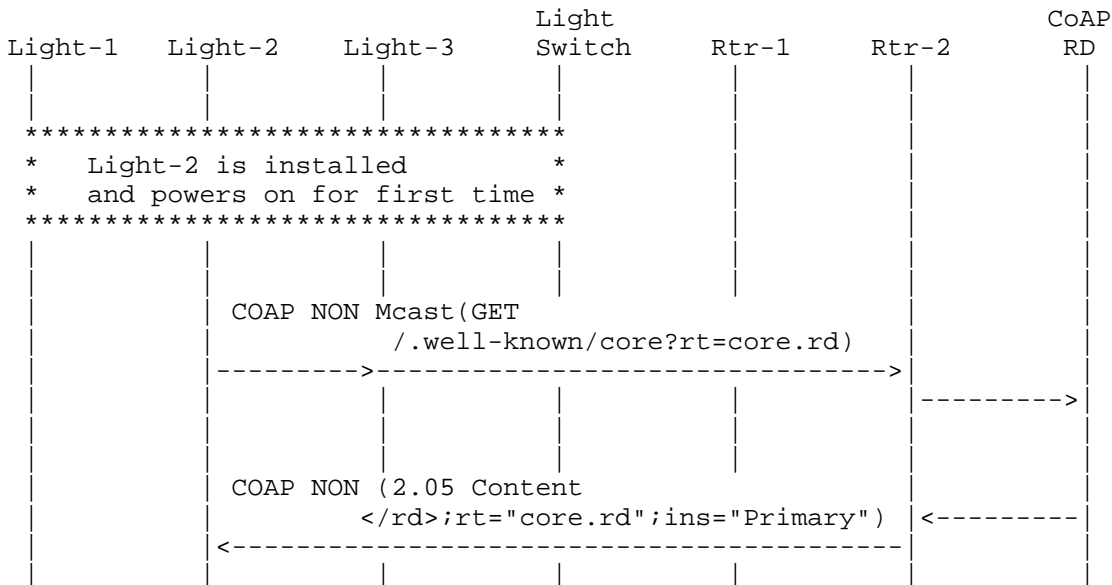
Figure 2: Resource Directory Discovery via Multicast Request

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 2.7 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 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 [MCAST-MPL].

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

1) Startup phase: 6LoWPANs are formed. IPv6 addresses are 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.

3a) 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.

3b) 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 that each node 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.

```
                                Light                         Network
      Light-1    Light-2    Light-3    Switch     Rtr-1     Rtr-2  Backbone
        |          |          |          |          |          |    |
        |          |          |          |          |          |    |
        |          |          ***********************          |    |
        |          |          *    User flips on    *          |    |
        |          |          *    light switch to  *          |    |
        |          |          *    turn on all the  *          |    |
        |          |          *    lights in Room-A *          |    |
        |          |          ***********************          |    |
        |          |          |          |          |          |    |
        |          |          |          |          |          |    |
        |          |          |    COAP NON Mcast(PUT,         |    |
        |          |          |    Payload=lights ON)         |    |
        |<------------------------------+--------->|          |    |
        ON         |          |          |          |-------------------->|
        |          |          |          |          |          |<---------|
        |          |<---------|<---------------------------------|    |
        |          ON         ON         |          |          |    |
        ^          ^          ^          |          |          |    |
        ***********************          |          |          |    |
        *    Lights in Room-A *          |          |          |    |
        *    turn on (nearly  *          |          |          |    |
        *    simultaneously)  *          |          |          |    |
        ***********************          |          |          |    |
        |          |          |          |          |          |    |
```
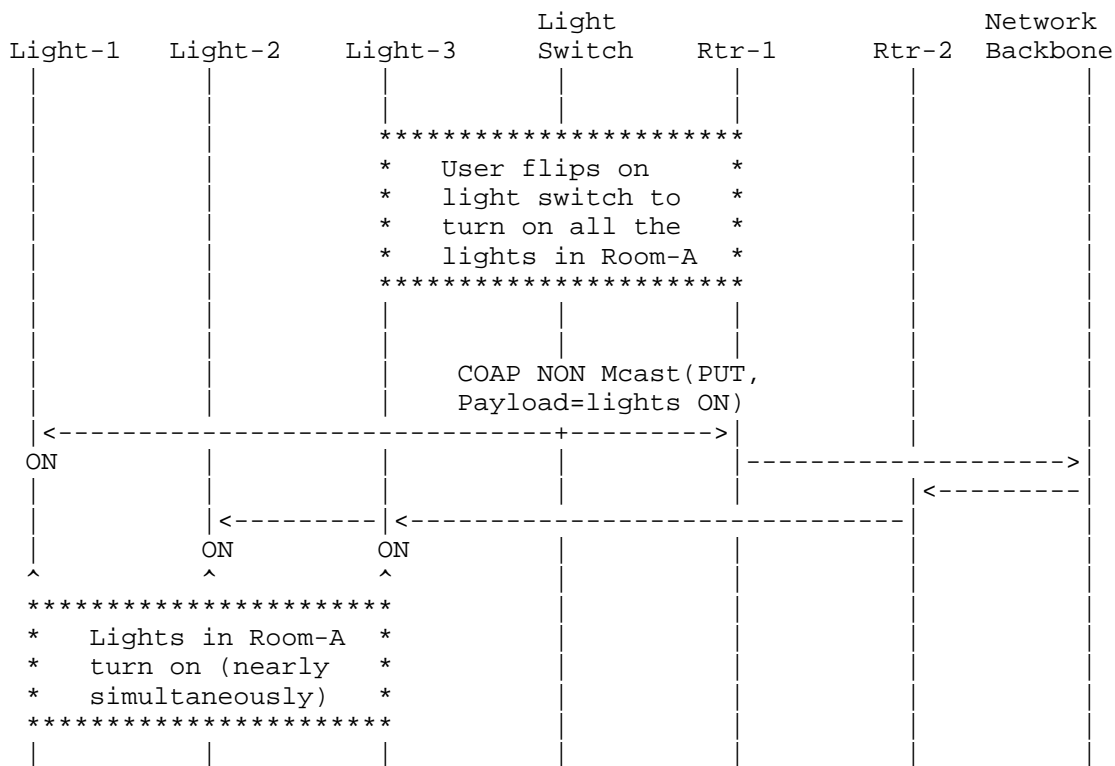
            Figure 3: Light Switch Sends Multicast Control Message

```
                                       Light                      Network
     Light-1    Light-2     Light-3    Switch      Rtr-1    Rtr-2  Backbone
       |          |           |          |          |        |       |
       |      COAP NON (2.04 Changed)    |          |        |       |
       |-------------------------------->|          |        |       |
       |          |           |          |          |        |       |
       |          |           |          |          |        |       |
       |          |    COAP NON (2.04 Changed)      |        |       |
       |          |-----------------------------------------> |       |
       |          |           |          |          |        |------->|
       |          |           |          |          |        |<-------------------|
       |          |           |          |        |<---------|        |       |
       |          |           |          |<---------|          |        |       |
       |          |           |          |          |          |        |       |
       |          |   COAP NON (5.00 Internal Server Error)    |       |
       |          |           |--------------------------------->|       |
       |          |           |          |          |          |------->|
       |          |           |          |          |          |<-------------------|
       |          |           |          |        |<---------|        |       |
       |          |           |          |<---------|          |        |       |
       |          |           |          |          |          |        |       |
```
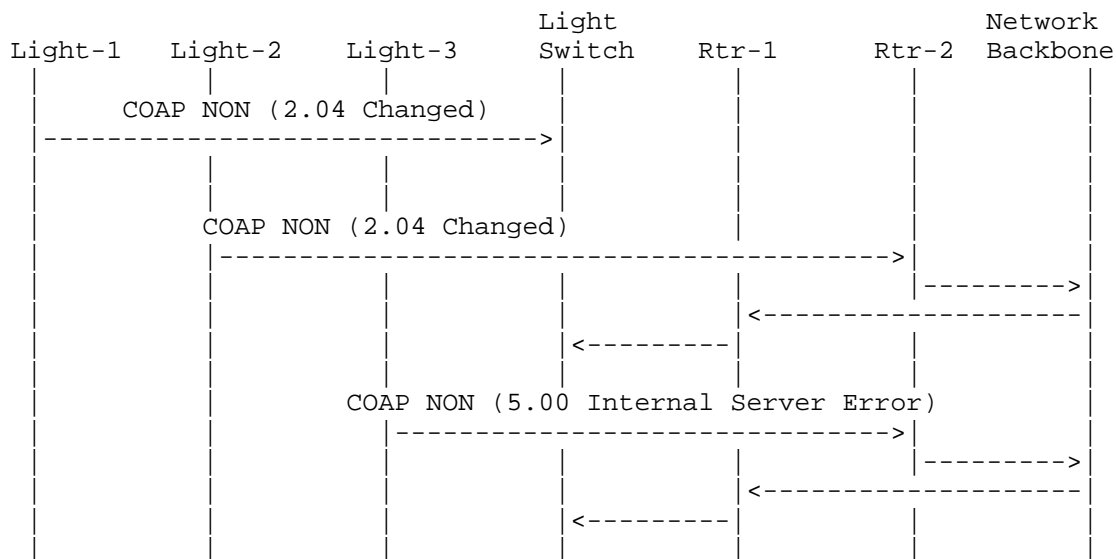
              Figure 4: Lights (Optionally) Respond to Multicast CoAP Request

     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.

```
                                                     Network
   Light-1    Light-2    Light-3     Rtr-1     Rtr-2  Backbone Controller
      |          |          |          |          |          |        |
      |          |          |          |          |    COAP NON Mcast(PUT,
      |          |          |          |          |    Payload=lights ON)
      |          |          |          |          |          |<-------|
      |          |          |          |<----------<---------|        |
      |<--------------------------------|          |          |        |
     ON          |          |          |          |          |        |
      |          |<----------<--------------------|          |        |
      |         ON         ON          |          |          |        |
      ^          ^          ^          |          |          |        |
      ***********************          |          |          |        |
      *    Lights in Room-A  *          |          |          |        |
      *    turn on (nearly    *         |          |          |        |
      *    simultaneously)    *         |          |          |        |
      ***********************          |          |          |        |
      |          |          |          |          |          |        |
      |          |          |          |          |          |        |
      |      COAP NON (2.04 Changed)   |          |          |        |
      |-------------------------------->|          |          |        |
      |          |          |          |-------------------->|        |
      |          |      COAP NON (2.04 Changed)    |          |------->|
      |          |-------------------------------->|          |        |
      |          |          |          |          |--------->|        |
      |          |          |      COAP NON (2.04 Changed)    |------->|
      |          |          |-------------------->|          |        |
      |          |          |          |          |--------->|        |
      |          |          |          |          |          |------->|
      |          |          |          |          |          |        |
```
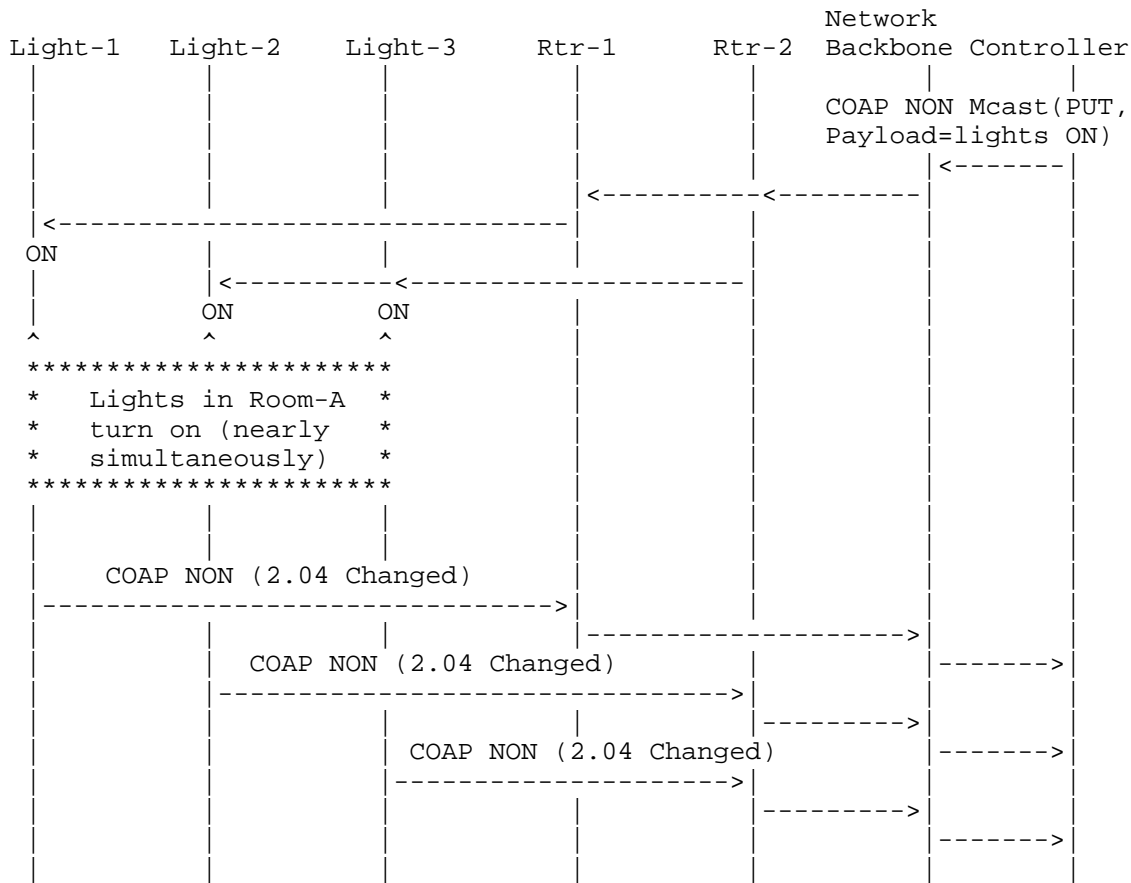
               Figure 5: Controller on Backbone Sends Multicast Control Message

3.5. Lighting Control in MLD-Enabled Network

   The use case in the 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 and 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
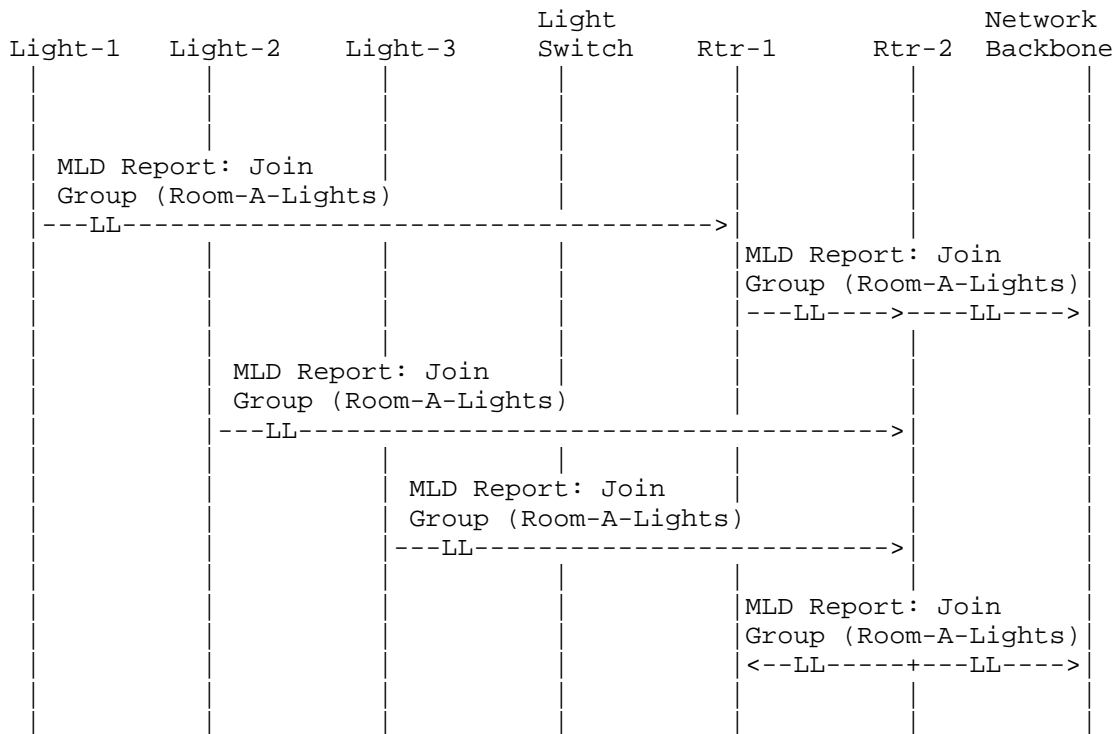previously indicated in Section 3.4, step 2, is needed anymore.

```
                                       Light                     Network
  Light-1   Light-2   Light-3    Switch      Rtr-1     Rtr-2  Backbone
     |         |         |         |           |         |         |
     |         |         |         |           |         |         |
     |         |         |         |           |         |         |
     | MLD Report: Join  |         |           |         |         |
     | Group (Room-A-Lights)       |           |         |         |
     |---LL------------------------------------>|         |         |
     |         |         |         |           |MLD Report: Join    |
     |         |         |         |           |Group (Room-A-Lights)|
     |         |         |         |           |---LL---->----LL---->|
     |         |         |         |           |         |         |
     |         | MLD Report: Join  |           |         |         |
     |         | Group (Room-A-Lights)         |         |         |
     |         |---LL------------------------------------>|         |
     |         |         |         |           |         |         |
     |         |         | MLD Report: Join    |         |         |
     |         |         | Group (Room-A-Lights)         |         |
     |         |         |---LL------------------------->|         |
     |         |         |         |           |         |         |
     |         |         |         |           |MLD Report: Join    |
     |         |         |         |           |Group (Room-A-Lights)|
     |         |         |         |           |<--LL-----+---LL---->|
     |         |         |         |           |         |         |
     |         |         |         |           |         |         |
```

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 the RD
   [CoRE-RD] and its group configuration feature.

   Once the 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 unicast 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 [CoRE-RD].
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 2.6.2.2.

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 on 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 an LLN such as a 6LoWPAN network, or consist of mixed
   traditional/constrained network segments. Second, it may be a single
   subnet only or a 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 the
   following:

   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 MLD protocol [RFC3810] (see Appendix A of this document) 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 the 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 Destination
   Advertisement Object (DAO) messages and routing of multicast IPv6
   packets based on this. It requires the RPL mode of operation 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, RPL 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 Destination-Oriented Directed
   Acyclic Graph (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 [MCAST-MPL] 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 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 that
      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 for each 6LoWPAN subnet, one Realm-Local
      ([RFC7346]) 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 the above example, include the
          network backbone to be part of each of the two MPL Domains. Or,
          in the 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 an edge router(s) as an MPL Seed(s) for
          specific IP multicast traffic. In the above example, this could
          be done through the following three steps: 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 in receiving 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 in receiving 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 a protocol is currently not defined
    in [MCAST-MPL].

In conclusion, 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 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 Sections 8.1 and 9.1 of [RFC7252], CoAP group
communication based on IP multicast will do the following:

o Operate in CoAP NoSec (No Security) mode, until a future group
   security solution is developed (see also Section 5.3.3).

   o Use the "coap" scheme. The "coaps" scheme should only be used
      when a future group security solution is developed (see also
      Section 5.3.3).

   Essentially, the above configuration means that there is currently no
   security at the CoAP layer for group communication. Therefore, for
   sensitive and mission-critical applications (e.g., health monitoring
   systems and alarm monitoring systems), it is currently recommended to
   deploy CoAP group communication with an application-layer security
   mechanism (e.g., data object security) for improved security.

   Application-level security has many desirable properties, including
   maintaining security properties while forwarding traffic through
   intermediaries (proxies). Application-level security also tends to
   more cleanly separate security from the dynamics of group membership
   (e.g., the problem of distributing security keys across large groups
   with many members that come and go).

   Without application-layer security, CoAP group communication should
   only be currently deployed in non-critical applications (e.g., read-
   only temperature sensors). Only when security solutions at the CoAP
   layer are mature enough (see Section 5.3.3) should CoAP group
   communication without application-layer security be considered for
   sensitive and mission-critical applications.

5.2. Threats

   As noted above, 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 Section 11 of
   [RFC7252] for IP multicast.

5.3. Threat Mitigation

   Section 11 of [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
   Premises 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, security enhancements
   need to be developed at the IETF for group communications. This will
   allow introduction of a secure mode of CoAP group communication and
   use of the "coaps" scheme for that purpose.

   At the time of writing this specification, there are various
   approaches being considered for security enhancements for group
   communications. Specifically, a lot of the current effort at the
   IETF is geared towards developing DTLS-based group communication.
   This is primarily motivated by the fact that unicast CoAP security is
   DTLS based (Section 9.1 of [RFC7252]. For example, [MCAST-SECURITY]
   proposes DTLS-based IP multicast security. However, it is too early
   to conclude if this is the best approach. Alternatively,
   [IPSEC-PAYLOAD] proposes IPsec-based IP multicast security. This
   approach also needs further investigation and validation.

5.4. Monitoring Considerations

5.4.1. General Monitoring

   CoAP group communication is meant to be used to control a set of
   related devices (e.g., simultaneously turn on all the lights in a
   room). This intrinsically exposes the group to some unique
   monitoring risks that solitary devices (i.e., devices not in a group)
   are not as vulnerable to. For example, assume an attacker is able to
   physically see a set of lights turn on in a room. Then the attacker
   can correlate a CoAP group communication message to that easily
   observable coordinated group action even if the contents of the
   message are encrypted by a future security solution (see
   Section 5.3.3). This will give the attacker side-channel information
   to plan further attacks (e.g., by determining the members of the
   group, then some network topology information may be deduced).

   One mitigation to group communication monitoring risks that should be
   explored in the future is methods to decorrelate coordinated group
   actions. For example, if a CoAP group communication GET is sent to
   all the alarm sensors in a house, then their (unicast) responses
   should be as decorrelated as possible. This will introduce greater
   entropy into the system and will make it harder for an attacker to
   monitor and gather side-channel information.

5.4.2. Pervasive Monitoring

   A key additional threat consideration for group communication is
   pointed to by [RFC7258], which warns of the dangers of pervasive
   monitoring. CoAP group communication solutions that are built on top
   of IP multicast need to 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 (once future
   security solutions are developed as in Section 5.3.3) may be used for
   the control and monitoring of critical infrastructure (e.g., lights,
   alarms, etc.) that 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 nodes for further attacks. For
   example, the source node (controller) sends out the CoAP group
   communication messages. 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 2.8).

In the future, even if all the CoAP multicast traffic is encrypted,
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=) Link Target Attribute,
   'core.gp', in the "Resource Type (rt=) Link Target Attribute Values"
   subregistry under the "Constrained RESTful Environments (CoRE)
   Parameters" registry.

   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 2.6.2.

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

   This memo registers a new Internet media type for the CoAP Group
   Configuration resource called 'application/coap-group+json'.

   Type name: application

   Subtype name: coap-group+json

   Required parameters: None

   Optional parameters: None

   Encoding considerations: 8-bit UTF-8.

   JSON to be represented using UTF-8, which is 8-bit 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
deregister) from the related IP multicast group. To prevent this, it
is recommended that a form of authorization (making use of unicast
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: RFC 7390

Applications that use this media type:

CoAP client and server implementations that wish to set/read the
Group Configuration resource via the 'application/coap-group+json'
payload as described in Section 2.6.2.

Fragment identifier considerations: N/A

Additional Information:

    Deprecated alias names for this type: None

    Magic number(s): None

    File extension(s): *.json

    Macintosh file type code(s): TEXT

Person and email address to contact for further information:

    Esko Dijk ("Esko.Dijk@Philips.com")

Intended usage: COMMON

Restrictions on usage: None

Author: CoRE WG

Change controller: IETF

Provisional registration? (standards tree only): N/A

7. References

7.1. Normative References

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

   [RFC2782]   Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for
               specifying the location of services (DNS SRV)", RFC 2782,
               February 2000, <http://www.rfc-editor.org/info/rfc2782>.

   [RFC3376]   Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A.
               Thyagarajan, "Internet Group Management Protocol, Version
               3", RFC 3376, October 2002,
               <http://www.rfc-editor.org/info/rfc3376>.

   [RFC3433]   Bierman, A., Romascanu, D., and K. Norseth, "Entity Sensor
               Management Information Base", RFC 3433, December 2002,
               <http://www.rfc-editor.org/info/rfc3433>.

   [RFC3542]   Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei,
               "Advanced Sockets Application Program Interface (API) for
               IPv6", RFC 3542, May 2003,
               <http://www.rfc-editor.org/info/rfc3542>.

   [RFC3810]   Vida, R. and L. Costa, "Multicast Listener Discovery
               Version 2 (MLDv2) for IPv6", RFC 3810, June 2004,
               <http://www.rfc-editor.org/info/rfc3810>.

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

   [RFC4291]   Hinden, R. and S. Deering, "IP Version 6 Addressing
               Architecture", RFC 4291, February 2006,
               <http://www.rfc-editor.org/info/rfc4291>.

   [RFC4601]   Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas,
               "Protocol Independent Multicast - Sparse Mode (PIM-SM):
               Protocol Specification (Revised)", RFC 4601, August 2006,
               <http://www.rfc-editor.org/info/rfc4601>.

   [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6
              over Low-Power Wireless Personal Area Networks (6LoWPANs):
              Overview, Assumptions, Problem Statement, and Goals", RFC
              4919, August 2007,
              <http://www.rfc-editor.org/info/rfc4919>.

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

   [RFC5110] Savola, P., "Overview of the Internet Multicast Routing
              Architecture", RFC 5110, January 2008,
              <http://www.rfc-editor.org/info/rfc5110>.

   [RFC5771] Cotton, M., Vegoda, L., and D. Meyer, "IANA Guidelines for
              IPv4 Multicast Address Assignments", BCP 51, RFC 5771,
              March 2010, <http://www.rfc-editor.org/info/rfc5771>.

   [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6
              Address Text Representation", RFC 5952, August 2010,
              <http://www.rfc-editor.org/info/rfc5952>.

   [RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in
              Customer Premises Equipment (CPE) for Providing
              Residential IPv6 Internet Service", RFC 6092, January
              2011, <http://www.rfc-editor.org/info/rfc6092>.

   [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,
              <http://www.rfc-editor.org/info/rfc6550>.

   [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,
              <http://www.rfc-editor.org/info/rfc6636>.

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

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

   [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, <http://www.rfc-editor.org/info/rfc6775>.

   [RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data
              Interchange Format", RFC 7159, March 2014,
              <http://www.rfc-editor.org/info/rfc7159>.

   [RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol
              (HTTP/1.1): Message Syntax and Routing", RFC 7230, June
              2014, <http://www.rfc-editor.org/info/rfc7230>.

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

   [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an
              Attack", BCP 188, RFC 7258, May 2014,
              <http://www.rfc-editor.org/info/rfc7258>.

   [RFC7320] Nottingham, M., "URI Design and Ownership", BCP 190, RFC
              7320, July 2014, <http://www.rfc-editor.org/info/rfc7320>.

7.2. Informative References

   [RFC1033] Lottor, M., "Domain administrators operations guide", RFC
              1033, November 1987,
              <http://www.rfc-editor.org/info/rfc1033>.

   [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick,
              "Internet Group Management Protocol (IGMP) / Multicast
              Listener Discovery (MLD)-Based Multicast Forwarding
              ("IGMP/MLD Proxying")", RFC 4605, August 2006,
              <http://www.rfc-editor.org/info/rfc4605>.

   [RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker,
              "NACK-Oriented Reliable Multicast (NORM) Transport
              Protocol", RFC 5740, November 2009,
              <http://www.rfc-editor.org/info/rfc5740>.

   [RFC7346] Droms, R., "IPv6 Multicast Address Scopes", RFC 7346,
              August 2014, <http://www.rfc-editor.org/info/rfc7346>.

   [BLOCKWISE-CoAP]
              Bormann, C. and Z. Shelby, "Blockwise transfers in CoAP",
              Work in Progress, draft-ietf-core-block-15, July 2014.

   [CoRE-RD] Shelby, Z., Bormann, C., and S. Krco, "CoRE Resource
              Directory", Work in Progress, draft-ietf-core-resource-
              directory-01, December 2013.

   [OBSERVE-CoAP]
              Hartke, K., "Observing Resources in CoAP", Work in
              Progress, draft-ietf-core-observe-14, June 2014.

   [MCAST-MPL]
              Hui, J. and R. Kelsey, "Multicast Protocol for Low power
              and Lossy Networks (MPL)", Work in Progress, draft-ietf-
              roll-trickle-mcast-09, April 2014.

   [MCAST-SECURITY]
              Keoh, S., Kumar, S., Garcia-Morchon, O., Dijk, E., and A.
              Rahman, "DTLS-based Multicast Security in Constrained
              Environments", Work in Progress, draft-keoh-dice-
              multicast-security-08, July 2014.

   [IPSEC-PAYLOAD]
              Migault, D. and C. Bormann, "IPsec/ESP for Application
              Payload", Work in Progress, draft-mglt-dice-ipsec-for-
              application-payload-00, July 2014.

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 MLD protocol [RFC3810] (or its IPv4 equivalent, IGMP [RFC3376])
   is today the method of choice used by a (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.

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

Acknowledgements

Authors' Addresses

   Akbar Rahman (editor)
   InterDigital Communications, LLC
   1000 Sherbrooke Street West
   Montreal, Quebec H3A 3G4
   Canada

   EMail: Akbar.Rahman@InterDigital.com

   Esko Dijk (editor)
   Philips Research
   High Tech Campus 34
   Eindhoven 5656AE
   Netherlands

   EMail: esko.dijk@philips.com