CoRE Working Group   E. Dijk, Ed.
Internet-Draft   Philips Research
Intended status: Informational   A. Rahman, Ed.
Expires: June 23, 2013   InterDigital Communications, LLC
                                                  December 20, 2012

              Miscellaneous CoAP Group Communication Topics
                   draft-dijk-core-groupcomm-misc-03

Abstract

   This document contains miscellaneous text around the topic of group
   communication for the Constrained Application Protocol (CoAP). The
   first part contains, for reference, text that was removed from the
   Group Communication for CoAP draft. The second part describes group
   communication and multicast functionality that may be input to future
   standardization in the CoRE WG.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on June 23, 2013.

Table of Contents

1. Introduction

   This document contains miscellaneous text around the topic of group
   communication for the Constrained Application Protocol, CoAP
   [I-D.ietf-core-coap]. The first part of the document (Section 5)
   contains, for reference, text that was removed from the Group
   Communication for CoAP [I-D.ietf-core-groupcomm] draft and its
   predecessor [I-D.rahman-core-groupcomm]. The second part of the
   document (Section 9) contains text and/or functionality that may be
   considered for inclusion in [I-D.ietf-core-groupcomm] or otherwise
   may be input to future standardization in the CoRE WG.

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [RFC2119].

2. Potential Solutions for Group Communication

   The classic concept of group communications is that of a single
   source distributing content to multiple destination recipients that
   are all part of a group. Before content can be distributed, there is
   a separate process to form the group. The source may be either a
   member or non-member of the group.

   Group communication solutions have evolved from "bottom" to "top",
   i.e., from layer 2 (Media Access Control broadcast/multicast) and
   layer 3 (IP multicast) to application layer group communication, also
   referred to as application layer multicast. A study published in
   2005 [Lao05] identified new solutions in the "middle" (referred to as
   overlay multicast) that utilize an infrastructure based on proxies.

   Each of these classes of solutions may be compared [Lao05] using
   metrics such as link stress and level of host complexity
   [Banerjee01]. The results show for a realistic internet topology
   that IP Multicast is the most resource-efficient, with the downside
   being that it requires the most effort to deploy in the
   infrastructure. IP Multicast is the solution adopted by this draft
   for CoAP group communication.

3. Use Cases

   CoAP group communication can be applied in the context of the
   following use cases:

   o Discovery of Resource Directory: discovering the local CoRE RD
      which contains links (URIs) to resources stored on other servers

[RFC6690].

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

o Parameter Update: updating parameters/settings simultaneously in a
   large group of devices in a building/campus control
   ([I-D.vanderstok-core-bc]) application.

o Firmware Update: efficiently updating firmware simultaneously in a
   large group of devices in a building/campus control
   ([I-D.vanderstok-core-bc]) application. Here, the use of CoAP
   group communication could be realized via a multicast extension of
   CoAP blockwise transfer [I-D.ietf-core-block]. This use case and
   use of multicast is especially valuable if there are time
   constraints related to the software update for large groups of
   devices.

o Group Status Report: requesting status information or event
   reports from a group of devices in a building/campus control
   application. In this use case, conditional reporting is required:
   only device that have events to report (as indicated by the
   request query) respond, others remain silent. This use case
   requires reliable CoAP group communication, which is currently not
   in CoRE WG scope.

4. Requirements

   Requirements that a CoAP group communication solution should fulfill
   can be found in existing documents ([RFC5867],
   [I-D.ietf-6lowpan-routing-requirements], [I-D.vanderstok-core-bc],
   and [I-D.shelby-core-coap-req]). Below, a set of high-level
   requirements is listed that a group communication solution should
   ideally fulfill. In practice, all these requirements can never be
   satisfied at once in an LLN context. Furthermore, different use
   cases will have different needs i.e. an elaboration of a subset of
   below requirements.

4.1. Background

   The requirements for CoAP are documented in
   [I-D.shelby-core-coap-req]. In this draft, we focus and expand
   discussions on the requirements pertaining to CoAP "group
   communication" and "multicast" support as stated in
   [I-D.shelby-core-coap-req]:

REQ 9: CoAP will support a non-reliable IP multicast message to be sent to a group of Devices to manipulate a resource on all the Devices simultaneously. The use of multicast to query and advertise descriptions must be supported, along with the support of unicast responses.

Currently, the CoAP protocol [I-D.ietf-core-coap] supports unreliable IP multicast using UDP. It defines the unreliable multicast operation as follows in Section 4.5:

"CoAP supports sending messages to multicast destination addresses. Such multicast messages MUST be Non-Confirmable. Some mechanisms for avoiding congestion from multicast requests are being considered in [I-D.eggert-core-congestion-control]."

Additional requirements were introduced in [I-D.vanderstok-core-bc] driven by quality of experience issues in commercial lighting; the need for large numbers of devices to respond with near simultaneity to a command (multicast PUT), and for that command to be received reliably (reliable multicast).

4.2. General Requirements

A CoAP group communication solution should (ideally) meet the following general requirements:

GEN-REQ 1:   Optional Reliability: the application can select between unreliable group communication and reliable group communication.

GEN-REQ 2:   Efficiency: delivers messages more efficiently than a "serial unicast" solution. Provides a balance between group data traffic and control overhead.

GEN-REQ 3:   Low latency: deliver a message as quickly as possible.

GEN-REQ 4:   Synchrony: allows near-simultaneous modification of a resource on all devices in a target group, providing a perceived effect of synchrony or simultaneity. For example a specified time span D such that a message is delivered to all destinations in a time interval [t,t+D].

GEN-REQ 5:   Ordering: message ordering may be required for reliable group communication use cases.

GEN-REQ 6:   Security: see Section 4.3 for security requirements for group communication.

GEN-REQ 7:   Flexibility: support for one or many source(s), both dense and sparse networks, for high or low listener density, small or large number of groups, and multi-group membership.

GEN-REQ 8:   Robust group management: functionality to join groups, leave groups, view group membership, and persistent group membership in failure or sleeping node situations.

GEN-REQ 9:   Network layer independence: a solution is independent from specific unicast and/or IP multicast routing protocols.

GEN-REQ 10:   Minimal specification overhead: a group communication solution should preferably re-use existing/established (IETF) protocols that are suitable for LLN deployments, instead of defining new protocols from scratch.

GEN-REQ 11:   Minimal implementation overhead: e.g. a solution allows to re-use existing (software) components that are already present on constrained nodes such as (typical) 6LoWPAN/CoAP nodes.

GEN-REQ 12:   Mixed backbone/LLN topology support: a solution should work within a single LLN, and in combined LLN/backbone network topologies, including multi-LLN topologies. Both the senders and receivers of CoAP group messages may be attached to different network links or be part of different LLNs, possibly with routers or switches in between group members. In addition, different routing protocols may operate on the LLN and backbone networks. Preferably a solution also works with existing, common backbone IP infrastructure (e.g. switches or routers).

GEN-REQ 13:   CoAP Proxying support: a CoAP proxy can handle distribution of a message to a group on behalf of a (constrained) CoAP client.

GEN-REQ 14:   Suitable for operation on LLNs with constrained nodes.

4.3. Security Requirements

   Security for group communications at the IP level has been studied
   extensively in the IETF MSEC (Multicast Security) WG, and to a lesser
   extent in the IRTF SAMRG (Scalable Adaptive Multicast Research
   Group). In particular, [RFC3740], [RFC5374] and [RFC4046] are very
   instructive. A set of requirements for securing group communications
   in CoAP were derived from a study of these previous investigations as
   well as understanding of CoAP specific needs. These are listed
   below.

   A CoAP group communication solution should (ideally) meet the
   following security requirements:

   SEC-REQ 1:   Group communications data encryption: Important CoAP
                group communications shall be encrypted (using a group
                key) to preserve confidentiality. It shall also be
                possible to send CoAP group communications in the clear
                (i.e. unencrypted) for low value data.

   SEC-REQ 2:   Group communications source data authentication:
                Important CoAP group communications shall be
                authenticated by verifying the source of the data (i.e.
                that it was generated by a given and trusted group
                member). It shall also be possible to send
                unauthenticated CoAP group communications for low value
                data.

   SEC-REQ 3:   Group communications limited data authentication: Less
                important CoAP group communications shall be
                authenticated by simply verifying that it originated
                from one of the group members (i.e. without explicitly
                identifying the source node). This is a weaker
                requirement (but simpler to implement) than REQ2. It
                shall also be possible to send unauthenticated CoAP
                group communications for low value data.

   SEC-REQ 4:   Group key management: There shall be a secure mechanism
                to manage the cryptographic keys (e.g. generation and
                distribution) belonging to the group; the state (e.g.
                current membership) associated with the keys; and other
                security parameters.

   SEC-REQ 5:   Use of Multicast IPSec: The CoAP protocol
                [I-D.ietf-core-coap] allows IPSec to be used as one
                option to secure CoAP. If IPSec is used as a way to
                security CoAP communications, then multicast IPSec
                [RFC5374] should be used for securing CoAP group

communications.

SEC-REQ 6:   Independence from underlying routing security: CoAP
              group communication security shall not be tied to the
              security of underlying routing and distribution
              protocols such as PIM [RFC4601] and RPL [RFC6550].
              Insecure or inappropriate routing (including IP
              multicast routing) may cause loss of data to CoAP but
              will not affect the authenticity or secrecy of CoAP
              group communications.

SEC-REQ 7:   Interaction with HTTPS: The security scheme for CoAP
              group communications shall account for the fact that it
              may need to interact with HTTPS (Hypertext Transfer
              Protocol Secure) when a transaction involves a node in
              the general Internet (non-constrained network)
              communicating via a HTTP-CoAP proxy.

5. Group Communication Solutions

This section includes the text that describes the solutions of IP
multicast, overlay multicast, and application layer group
communication which were removed from [I-D.rahman-core-groupcomm]
version 07 when the text was transferred to
[I-D.ietf-core-groupcomm].

5.1. IP Multicast Transmission Methods

5.1.1. Serial unicast

Even in systems that generally support IP Multicast, there may be
certain data links (or transports) that don't support IP multicast.
For those links a serial unicast alternative must be provided. This
implies that it should be possible to enumerate the members of a
group, in order to determine the correct unicast destinations.

5.1.2. Unreliable IP Multicast

The CoRE WG charter specified support for non-reliable IP multicast.
In the current CoAP protocol design [I-D.ietf-core-coap], unreliable
multicast is realized by the source sending Non-Confirmable messages
to a multicast IP address. IP Multicast (using UDP) in itself is
unreliable, unless specific reliability features are added to it.

5.1.3. Reliable IP Multicast

   [TBD: This is a difficult problem. Need to investigate the benefits
   of repeating MGET and MPUT requests (saturation) to get "Pretty Good
   Reliability". Use the same MID or a new MID for repeated requests?
   Carsten suggests the use of bloom filters to suppress duplicate
   responses.

   One could argue that non-idempotent operations (POST) cannot be
   supported without a *truly* reliable multicast protocol. However, is
   this the case? If a multicast POST request is sent repeatedly with
   the same Message ID (MID), then CoAP nodes that already received it
   once will ignore duplicates. Sending with Message ID is supported in
   CoAP for Non-Confirmable messages (thus including multicast messages)
   as per [I-D.ietf-core-coap] section 4.2.]

   Reliable multicast supports guaranteed delivery of messages to a
   group of nodes. The following specifies the requirements as was
   proposed originally in version 01 of [I-D.vanderstok-core-bc]:

   o Validity - If sender sends a message, m, to a group, g, of
      destinations, a path exists between sender and destinations, and
      the sender and destinations are correct, all destinations in g
      eventually receive m.

   o Integrity - destination receives m at most once from sender and
      only if sender sent m to a group including destination.

   o Agreement - If a correct destination of g receives m, then all
      correct destinations of g receive m.

   o Timeliness - For real-time control of devices, there is a known
      constant D such that if m is sent at time t, no correct
      destination receives m after t+D.

   There are various approaches to achieve reliability, such as

   o Destination node sends response: a destination sends a CoAP
      Response upon multicast Request reception (it SHOULD be a Non-
      Confirmable response). The source node may retry a request to
      destination nodes that did not respond in time with a CoAP
      response.

   o Route redundancy

   o Source node transmits multiple times (destinations do not respond)

5.2. Overlay Multicast

   An alternative group communication solution (to IP Multicast) is an
   "overlay multicast" approach. We define an overlay multicast as one
   that utilizes an infrastructure based on proxies (rather than an IP
   router based IP multicast backbone) to deliver IP multicast packets
   to end devices. MLD ([RFC3810]) has been selected as the basis for
   multicast support by the ROLL working group for the RPL routing
   protocol. Therefore, it is proposed that "IGMP/MLD Proxying"
   [RFC4605] be used as a basis for an overlay multicast solution for
   CoAP.

   Specifically, a CoAP proxy [I-D.ietf-core-coap] may also contain an
   MLD Proxy function. All CoAP devices that want to join a given IP
   multicast group would then send an MLD Join to the CoAP (MLD) proxy.
   Thereafter, the CoAP (MLD) proxy would be responsible for delivering
   any IP multicast message to the subscribed CoAP devices. This will
   require modifications to the existing [RFC4605] functionality.

   Note that the CoAP (MLD) proxy may or may not be connected to an
   external IP multicast enabled backbone. The key function for the
   CoAP (MLD) proxy is to distribute CoAP generated multicast packets
   even in the absence of router support for multicast.

5.3. CoAP Application Layer Group Management

   Another alternative solution (to IP Multicast and Overlay Multicast)
   is to define CoAP application level group management primitives.
   Thus, CoAP can support group management features without need for any
   underlying IP multicast support.

   Interestingly, such group management primitives could also be offered
   even if there is underlying IP multicast support. This is useful
   because IP multicast inherently does not support the concept of a
   group with managed members, while a managed group may be required for
   some applications.

   The following group management primitives are in general useful:

   o discover groups;

   o query group properties (e.g. related resource descriptions);

   o create a group;

   o remove a group;

o add a group member;

o remove a group member;

o enumerate group members;

o security and access control primitives.

In this proposal a (at least one) CoAP Proxy node is responsible for group membership management. A constrained node can specify which group it intends to join (or leave) using a CoAP request to the appropriate CoAP Proxy. To Join, the group name will be included in optional request header fields (explained below). These header fields will be included in a PUT request to the Proxy. The Proxy-URI is set to the Group Management URI of the Proxy (found previously through the "/.well-known/" resource discovery mechanism). Note that in this solution also CoAP Proxies may exist in a network that are not capable of CoAP group operations.

Group names may be defined as arbitrary strings with a predefined maximum length (e.g. 268 characters or the maximum string length in a CoAP Option), or as URIs.

[TBD: how can a client send a request to a group? Does it only need to know the group name (string or URI) or also an IP multicast address? One way is to send a CoAP request to the CoAP Proxy with a group URI directly in the Proxy-URI field. This avoids having to know anything related to IP multicast addresses.]

This solution in principle supports both unreliable and reliable group communication. A client would indicate unreliable communication by sending a CoAP Non-Confirmable request to the CoAP Proxy, or reliable communication by sending a CoAP Confirmable request.

It is proposed that CoAP supports two Header Options for group "Join" and "Leave". These Options are Elective so they should be assigned an even number. Assuming the Type for "join" is x (value TBD), the Header Options are illustrated by the table in Figure 1:

```
+------+-----+--------------+-------------+--------+-------------+
| Type | C/E | Name         | Data type   | Length | Default     |
|------+-----+--------------+-------------+--------+-------------+
|      |     |              |             |        |             |
|  x   |  E  | Group Join   | String      | 1-270  | ""          |
|      |     |              |             | B      |             |
| x+2  |  E  | Group Leave  | String      | 1-270  | ""          |
|      |     |              |             | B      |             |
+------+-----|--------------+-------------+--------+-------------+
```

            Figure 1: CoAP Header Options for Group Management

   Figure 2 illustrates how a node can join or leave a group using the
   Header Options in a CoAP message:


```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Ver| T |   OC  |     Code      |          Message ID           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | delta |length |   Join Group A (ID or URI)
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   0   |length |   Join Group B (ID or URI)
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   2   |length |   Leave Group C (ID or URI)
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
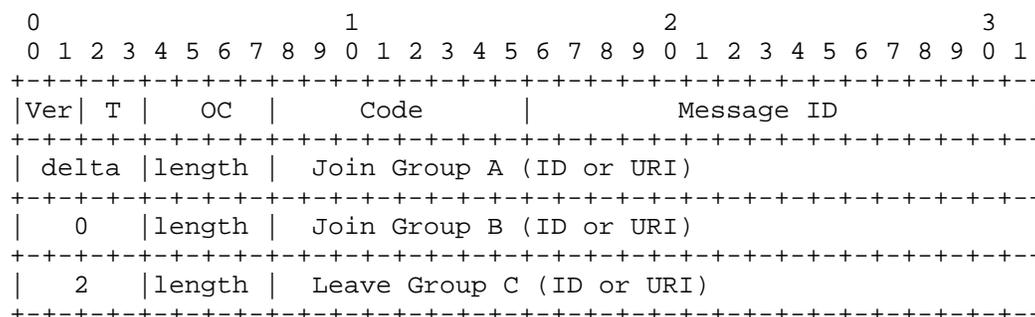
             Figure 2: CoAP Message for Group Management

   Header Fields for the above example:

   Ver: 2-bit unsigned integer for CoAP Version. Set to 1 by
   implementation as defined by the CoAP specification.

   T: 2-bit unsigned integer for CoAP Transaction Type. Either '0'
   Confirmation or '1' Non-Confirmable can be used for group "join" or
   "leave" request.

   OC: 4-bit unsigned integer for Option Count. For this example, the
   value should be "3" since there are three option fields.

   Code: 8-bit unsigned integer to indicate the Method in a Request or a
   Response Code in a Response message. Any Code can be used so the

group management can be piggy-backed in either Request or Response
message.

Message ID: 16-bit value assigned by the source to uniquely identify
a pair of Request and Response.

CoAP defines a delta encoding for header options. The first delta is
the "Type" for group join in this specific example. If the type for
group join is x as illustrated in Figure 2, delta will be x. In the
second header option, it is also a group join so the delta is 0. The
third header option is a group leave so the delta is 2.

An alternative solution to using Header Options (explained above) is
to use designated parameters in the query part of the URI in the
Proxy-URI field of a POST (TBD: or PUT?) request to a Proxy's group
management service resource advertized by DNS-SD. For example, to
join group1 and leave group2:

  coap://proxy1.bld2.example.com/groupmgt?j=group1&l=group2

6. DNS-SD Based Group Resource Manipulation

   Ideally, all nodes in a given group (defined by its multicast IP
   address) must receive the same request with high probability. This
   will not be the case if there is diversity in the authority port
   (i.e. a diversity of dynamic port addresses across the group) or if
   the targeted resource is located at different paths on different
   nodes. Extending the definition of group membership to include port
   and path discovery is not desirable.

   Therefore, some measures must be present to ensure uniformity in port
   number and resource name/location within a group.

   A first solution in this respect is to couple groups to service
   descriptions in DNS (using DNS-SD as in [I-D.vanderstok-core-bc]). A
   service description for a multicast group may have a TXT record in
   DNS defining a schema X (e.g. "schema=DALI"), which defines by
   service standard X (e.g. "DALI") which resources a node supporting X
   MUST have. Therefore a multicast source can safely refer to all
   resources with corresponding operations as prescribed by standard X.
   For port numbers (which can be found using DNS-SD also) the same
   holds. Alternatively, only the default CoAP port may be used in all
   CoAP multicast requests.

7. Group Discovery and Member Discovery

   CoAP defines a resource discovery capability, but does not yet
   specify how to discover groups (e.g. find a group to join or send a
   multicast message to) or to discover members of a group (e.g. to
   address selected group members by unicast). These topics are
   elaborated in more detail in [I-D.vanderstok-core-dna] including
   examples for using DNS-SD and CoRE Resource Directory.

7.1. DNS-SD

   DNS-based Service Discovery [I-D.cheshire-dnsext-dns-sd] defines a
   conventional way to configure DNS PTR, SRV, and TXT records to enable
   enumeration of services, such as services offered by CoAP nodes, or
   enumeration of all CoAP nodes, within specified subdomains. A
   service is specified by a name of the form
   <Instance>.<ServiceType>.<Domain>, where the service type for CoAP
   nodes is _coap._udp and the domain is a DNS domain name that
   identifies a group as in the examples above. For each CoAP end-point
   in a group, a PTR record with the name _coap._udp and/or a PTR record
   with the name _coap._udp.<Domain> is defined and it points to an SRV
   record having the <Instance>.<ServiceType>.<Domain> name.

   All CoAP nodes in a given subdomain may be enumerated by sending a
   query for PTR records named _coap._udp to the authoritative DNS
   server for that zone. A list of SRV records is returned. Each SRV
   record contains the port and host name (AAAA record) of a CoAP node.
   The IP address of the node is obtained by resolving the host name.
   DNS-SD also specifies an optional TXT record, having the same name as
   the SRV record, which can contain "key=value" attributes. This can
   be used to store information about the device, e.g. schema=DALI,
   type=switch, group=lighting.bldg6, etc.

   Another feature of DNS-SD is the ability to specify service sub-types
   using PTR records. For example, one could represent all the CoAP
   groups in a subdomain by PTR records with the name
   _group._sub._coap._udp or alternatively
   _group._sub._coap._udp.<Domain>.

7.2. CoRE Resource Directory

   CoRE Resource Directory [I-D.shelby-core-resource-directory] defines
   the concept of a Resource Directory (RD) server where CoAP servers
   can register their resources offered and CoAP clients can discover
   these resources by querying the RD server. RD syntax can be mapped
   to DNS-SD syntax and vice versa [I-D.lynn-core-discovery-mapping],
   such that the above approach can be reused for group discovery and
   group member discovery.

Specifically, the Domain (d) parameter can be set to the group URI by an end-point registering to the RD. If an end-point wants to join multiple groups, it has to repeat the registration process for each group it wants to join.

8. Deployment Guidelines

8.1. Overview

We recommend to use IP multicast as the base solution for CoAP Group Communication, provided that the use case and network characteristics allow this. It has the advantage that it re-uses the IP multicast suite of protocols and can operate even if group members are distributed over both constrained and un-constrained network segments. Still, this approach may require specifying or implementing additional IP Multicast functionality in an LLN, in a backbone network, or in both - this will be evaluated in more detail in this section.

8.2. Implementation in Target Network Topologies

This section looks in more detail how an IP Multicast based solution can be deployed onto the various network topologies that we consider important for group communication use cases. Note that the chosen solution of IP Multicast for CoAP group communication works mostly independently from the underlying network topology and its specific IP multicast implementation.

Starting from the simplest case of a single LLN topology, we move to more complex topologies involving a backbone network or multiple LLNs. With "backbone" we refer here typically to a corporate LAN or VLAN, which constitutes a single broadcast domain by design. It could also be an in-home network. A multi-link backbone is also possible, if there is proper IP multicast routing or forwarding configured between these links. (The term 6LoWPAN Border Router or "6LBR" is used here for a border router, though our evaluation is not necessarily restricted to 6LoWPAN networks.)

8.2.1. Single LLN Topology

The simplest topology is a single LLN, where all the IP multicast source(s) and destinations are constrained nodes within this same LLN. Possible implementations of IP multicast routing and group administration for this topology are listed below.

8.2.1.1. Mesh-Under Multicast Routing

   The LLN may be set up in either a mesh-under or a route-over
   configuration. In the former case, the mesh routing protocol should
   take care of routing IP multicast messages throughout the LLN.

   Because conceptually all nodes in the LLN are attached to a single
   link, there is in principle no need for nodes to announce their
   interest in multicast IP addresses via MLD (see Appendix A). A
   multicast message to a specific IP destination, which is delivered to
   all 6LoWPAN nodes by the mesh routing algorithm, is accepted by the
   IP network layer of that node only if it is listening on that
   specific multicast IP address and port.

8.2.1.2. RPL Multicast Routing

   The RPL routing protocol for LLNs provides support for routing to
   multicast IP destinations (Section 12 of [RFC6550]). Like regular
   unicast destinations, multicast destinations are advertised by nodes
   using RPL DAO messages. This functionality requires "Storing mode
   with multicast support" (Mode Of Operation, MOP is 3) in the RPL
   network.

   Once all RPL routing tables in the network are populated, any RPL
   node can send packets to an IP multicast destination. The RPL
   protocol performs distribution of multicast packet both upward
   towards the DODAG root and downwards into the DODAG.

   The text in Section 12 of the RPL specification clearly implies that
   IP multicast packets are distributed using link-layer unicast
   transmissions, looking at the use of the word "copied" in this
   section. Specifically in 6LoWPAN networks, this behavior conflicts
   with the requirement that IP multicast packets MUST be carried as
   link-layer 802.15.4 broadcast frames [RFC4944].

   Assuming that link-layer unicast is indeed meant, this approach seems
   efficient only in a balanced, sparse tree network topology, or in
   situations where the fraction of nodes listening to a specific
   multicast IP address is low, or in duty cycled LLNs where link-layer
   broadcast is a very expensive operation.

8.2.1.3. RPL Routers with Non-RPL Hosts

   Now we consider the case that hosts exist in a RPL network that are
   not RPL-aware themselves, but rely on RPL routers for their IP
   connectivity beyond link-local scope. Note that the current RPL
   specification [RFC6550] leaves this case for future specification
   (see Section 16.4). Non-RPL hosts cannot advertise their IP

multicast groups of interest via RPL DAO messages as defined above. Therefore in that case MLD could be used for such advertisements (State Change Report messages), with all or a subset of RPL routers acting in the role of MLD Routers as defined in [RFC3810]. However, as the MLD protocol is not designed specifically for LLNs it may be a burden for the constrained RPL router nodes to run the full MLD protocol. Alternatives are therefore proposed in Section 8.3.1.

8.2.1.4. Trickle Multicast Forwarding

Trickle Multicast Forwarding [I-D.ietf-roll-trickle-mcast] is an IP multicast routing protocol suitable for LLNs, that uses the Trickle algorithm as a basis. It is a simple protocol in the sense that no topology maintenance is required. It can deal especially well with situations where the node density is a-priori unknown.

Nodes from anywhere in the LLN can be the multicast source, and nodes anywhere in the LLN can be multicast destinations.

Using Trickle Multicast Forwarding it is not required for IP multicast destinations (listeners) to announce their interest in a specific multicast IP address, e.g. by means of MLD. Instead, all multicast IP packets regardless of IP destination address are stored and forwarded by all routers. Because forwarding is always done by multicast, both hosts and routers will be able to receive all multicast IP packets. Routers that receive multicast packets they are not interested in, will only buffer these for a limited time until retransmission can be stopped as specified by the protocol. Hosts that receive multicast packets they are not interested in, will discard multicast packets that are not of interest. Above properties seem to make Trickle especially efficient for cases where the multicast listener density is high and the number of distinct multicast groups relatively low.

8.2.1.5. Other Route-Over Methods

Other known IP multicast routing methods may be used, for example flooding or other to be defined methods suitable for LLNs. An important design consideration here is whether multicast listeners need to advertise their interest in specific multicast addresses, or not. If they do, MLD is a possible option but also protocol-specific means (as in RPL) is an option. See Section 8.3.1 for more efficient substitutes for MLD targeted towards a LLN context.

8.2.2. Single LLN with Backbone Topology

A LLN may be connected via a Border Router (e.g. 6LBR) to a backbone network, on which IP multicast listeners and/or sources may be

present. This section analyzes cases in which IP multicast traffic
needs to flow from/to the backbone, to/from the LLN.

8.2.2.1. Mesh-Under Multicast Routing

Because in a mesh routing network conceptually all nodes in the LLN
are attached to a single link, a multicast IP packet originating in
the LLN is typically delivered by the mesh routing algorithm to the
6LBR as well, although there is no guaranteed delivery. The 6LBR may
be configured to accept all IP multicast traffic from the LLN and
then may forward such packets onto its backbone link. Alternatively,
the 6LBR may act in an MLD Router or MLD Snooper role on its backbone
link and decide whether to forward a multicast packet or not based on
information learned from previous MLD Reports received on its
backbone link.

Conversely, multicast packets originating on the backbone network
will reach the 6LBR if either the backbone is a single link (LAN/
VLAN) or IPv6 multicast routing is enabled on the backbone. Then,
the 6LBR could simply forward all IP multicast traffic from the
backbone onto the LLN. However, in practice this situation may lead
to overload of the LLN caused by unnecessary multicast traffic.
Therefore the 6LBR SHOULD only forward traffic that one or more nodes
in the LLN have expressed interest in, effectively filtering inbound
LLN multicast traffic.

To realize this "filter", nodes on the LLN may use MLD to announce
their interest in specific multicast IP addresses to the 6LBR. One
option is for the 6LBR to act in an MLD Router role on its LLN
interface. However, this may be too much of a "burden" for
constrained nodes. Light-weight alternatives for MLD are discussed
in Section 8.3.1.

8.2.2.2. RPL Multicast Routing

For RPL routing within the 6LoWPAN, we first consider the case of an
IP multicast source on the backbone network with one or more IP
multicast listeners on the RPL LLN. Typically, the 6LBR would be the
root of a DODAG so that the 6LBR can easily forward the IP multicast
packet received on its backbone interface to the right RPL nodes in
the LLN down along this DODAG (based on previously DAO-advertized
destinations).

Second, a multicast source may be in the RPL LLN and listeners may be
both on the LLN and on the backbone. For this case RPL defines that
the multicast packet will propagate both up and down the DODAG,
eventually reaching the DODAG root (typically a 6LBR) from which the
packet can be routed onto the backbone in a manner specified in the

previous section.

8.2.2.3. RPL Routers with Non-RPL Hosts

For the case that a RPL LLN contains non-RPL hosts, the solutions
from the previous section can be used if in addition RPL routers
implement MLD or "MLD like" functionality similar to as described in
Section 8.2.1.3.

8.2.2.4. Trickle Multicast Forwarding

First, we consider the case of an IP multicast source node on the LLN
(where all 6LRs support Trickle Multicast Forwarding) and IP
multicast listeners that may be on the LLN and on the backbone. As
Trickle will eventually deliver multicast packets also to a 6LBR,
which acts as a Trickle Multicast router as well, the 6LBR can then
forward onto the backbone in the ways described earlier in
Section 8.2.2.1.

Second, for the case of an IP multicast source on the backbone and
multicast listeners on both backbone and/or LLN, the 6LBR needs to
forward multicast traffic from the backbone onto the LLN. Here, the
aforementioned problem (Section 8.2.2.1) of potentially overloading
the LLN with unwanted backbone IP multicast traffic appears again.

A possible solution to this is (again) to let multicast listeners
advertise their interest using MLD as described in Section 8.2.2.1 or
to use an MLD alternative suitable for LLNs as described in
Section 8.3.1. However, following this approach requires possibly an
extension to Trickle Multicast Forwarding: the protocol should ensure
that MLD-advertised information is somehow communicated to the 6LBR,
possibly over multiple hops. MLD itself supports link-local
communication only.

8.2.2.5. Other Route-Over Methods

For other multicast routing methods used on the LLN, there are
similar considerations to the ones in sections above: the strong need
to filter IP multicast traffic coming into the LLN, the need for
reporting multicast listener interest (e.g. with MLD or a to-be-
defined MLD alternative) by constrained (6LoWPAN) nodes, and the need
for LLN-internal routing as identified in the previous section such
that the MLD communicated information can reach the 6LBR to be used
there in multicast traffic filtering decisions.

8.2.3. Multiple LLNs with Backbone Topology

   Now the case of a single backbone network with two or more LLNs
   attached to it via 6LBRs is considered. For this case all the
   considerations and solutions of the previous section can be applied.

   For the specific case that a source on a backbone network has to send
   to a very large number of destination located on many LLNs, the use
   of IGMP/MLD Proxying [RFC4605] with a leaf IGMP/MLD Proxy located in
   each 6LBR may be useful. This method only is defined for a tree
   topology backbone network with the IP multicast source at the root of
   the tree.

8.2.4. LLN(s) with Multiple 6LBRs

   [TBD: an LLN with multiple 6LBRs may require some additional
   consideration. Any need to synchronize mutually on multicast
   listener information?]

8.2.5. Conclusions

   For all network topologies that were evaluated, CoAP group
   communication can be in principle supported with IP Multicast, making
   use of existing protocols. For the case of Trickle Multicast
   Forwarding, it appears that an addition to the protocol is required
   such that information about multicast listeners can be distributed
   towards the 6LBR. Opportunities were identified for an "MLD-like" or
   "MLD-lightweight" protocol specifically suitable for LLNs, which
   should inter-work with regular MLD on the backbone network. Such MLD
   variants are further analyzed in Section 8.3.1.

8.3. Implementation Considerations

   In this section various implementation aspects are considered such as
   required protocol implementations, additional functionality of the
   6LBR and backbone network equipment.

8.3.1. MLD Implementation on LLNs and MLD alternatives

   In previous sections, it was mentioned that the MLDv2 protocol
   [RFC3810] may be too costly for use in a LLN. MLD relies on periodic
   link-local multicast operations to maintain state. Also it is
   optimized to fairly dynamic situations where multicast listeners may
   come and go over time. Such dynamic situations are less frequently
   found in typical LLN use cases such as building control, where
   multicast group membership can remain constant over longer periods of
   time (e.g. months) after commissioning.

Hence, a viable strategy is to implement a subset of MLD functionality in 6LoWPAN nodes which is just enough for the required functionality. A first option is that 6LoWPAN Routers, like MLD Snoopers, passively listen to MLD State Change Report messages and handle the learned ("snooped") IP multicast destinations in the way defined by the multicast routing protocol they are running (e.g. for RPL, Routers advertise these destinations using DAO messages).

A second option is to use MLD as-is but adapt the recommended parameter values such that operation on a LLN becomes more efficient. [RFC6636] could be a guideline in this case.

A third option is to standardize a new protocol, taking a subset of MLD functionality into a "MLD for 6LoWPAN" protocol to support constrained nodes optimally.

A fourth option is now presented, which seems attractive in that it minimizes standardization, implementation and network communication overhead all at the same time. This option is to specify a new Multicast Listener Option (MLO) as an addition to the 6LoWPAN-ND [RFC6775] protocol communication that is anyway ongoing between a 6LoWPAN host and router(s). This MLO is preferably designed to be maximally similar to the Address Registration Option (ARO), which minimizes the need for additional program code on constrained nodes. With an MLO, instead of registering a hosts's unicast IP address as with ARO, a host "registers" its interest in a multicast IPv6 address. Unlike the ARO, multiple MLO can be used in the same ND packet. A registration period is also defined in the MLO just like in the ARO. MLO allows a host to persistently register as a listener to IP multicast traffic and to avoid the overhead of periodic multicast communication which is required for the regular MLD protocol.

[TBD: consider what aspects are needed/not needed for CoAP/LLN applications. Will MLDv1 suffice? What to do with options like 'source specific' and include/exclude. Source-specific can also be dealt with at the destination host by filtering? Do we need limits on number of records per packet? Do we need a higher MLD reliability setting - see the parameters in the MLD RFC]

8.3.2. 6LBR Implementation

To support mixed backbone/LLN scenarios in CoAP group communication, it is RECOMMENDED that a 6LowPAN Border Router (6LBR) will act in an MLD Router role on the backbone link. If this is not possible then the 6LBR SHOULD be configured to act as an MLD Multicast Address Listener and/or MLD Snooper on the backbone link.

8.3.3. Backbone IP Multicast Infrastructure

   For corporate/professional applications, most routing and switching
   equipment that is currently on the market is IPv6 capable. For that
   reason backbone infrastructure operating IPv4 only is considered out
   of scope in this document, at least for the backbone network
   segment(s) where IP multicast destinations are present. What is
   still in scope is for example an IPv4-only HTTP client that wants to
   send a group communication message via a HTTP-CoAP proxy as
   considered in [I-D.castellani-core-advanced-http-mapping].

   The availability of, and requirements for, IP multicast support may
   depend on the specific installation use case. For example, the
   following cases may be relevant for new IP based building control
   installations:

   1. System deployed on existing IP (Ethernet/WiFi/...)
       infrastructure, shared with existing IP devices (PCs)

   2. Newly designed and deployed IP (Ethernet/WiFi/...)
       infrastructure, to be shared with other IP devices (PCs)

   3. Newly designed and deployed IP (Ethernet/WiFi/...)
       infrastructure, exclusively used for building control.

   Besides physical separation the building control backbone can be
   separated from regular (PC) infrastructure by using a different VLAN.
   A typical corporate installation will have many LAN switches and/or
   routing switches, which pass through IP multicast traffic but on the
   other hand do not support acting in the Router role of MLD/IGMP.
   Perhaps for case 2) and 3) above it is acceptable to add a MLD/IGMP
   capable router somewhere in the network, while for case 1) this may
   not be the case.

   [TBD: consider the influence of WiFi based backbone networks. What
   if 6LBRs are at the same time also WiFi routers? What if 6LBRs have
   an Ethernet connection to legacy WiFI routers? Check if equivalent
   with Ethernet backbone.]

9. Miscellaneous Topics

   This section collects miscellaneous text, topics or proposals related
   to CoAP group communication which do not directly fit into any of the
   preceding sections.

9.1. CoAP Multicast and HTTP Unicast Interworking

   CoAP supports operation over UDP multicast, while HTTP does not. For
   use cases where it is required that CoAP group communication is
   initiated from an HTTP end-point, it would be advantageous if the
   HTTP-CoAP Proxy supports mapping of HTTP unicast to CoAP group
   communication based on IP multicast. One possible way of operation
   of such HTTP-CoAP Proxy is illustrated in Figure 3. Note that this
   topic is covered in more detail in
   [I-D.castellani-core-advanced-http-mapping].


```
        CoAP     Mcast    CoAP     Mcast   HTTP-CoAP           HTTP
        Node 1   Rtr1     Node 2   Rtr2    Proxy               Node 3
         |        |        |        |        |                   |
         |MLD REQUEST      |        |        |                   |
         |(Join Group X)   |        |        |                   |
         |--LL-->|         |        |        |                   |
         |        |        |MLD REQUEST      |                   |
         |        |        |(Join Group X)   |                   |
         |        |        |--LL-->|         |                   |
         |        |        |        |        |   HTTP REQUEST    |
         |        |        |        |        |     (URI to       |
         |        |        |        |        |    unicast addr)  |
         |        |        |        |        |< -----------------|
         |        |        |        |        |                   |
         |        |        |       Resolve HTTP Request-Line URI |
         |        |        |        to Group X multicast address |
         |        |        |        |        |                   |
         | CoAP REQUEST (to multicast addr)| |                   |
         |< ------<---------<-------<------| |                   |
         |        |        |        |        |                   |
         |        |        |        |        |                   |
         |     (optional) CoAP RESPONSE(s) | |                   |
         |        |        |-------------->| |                   |
         |-----------------|-------------->|   Aggregated        |
         |        |        |        |        |  HTTP RESPONSE     |
         |        |        |        |        |------------------>|
         |        |        |        |        |                   |
```
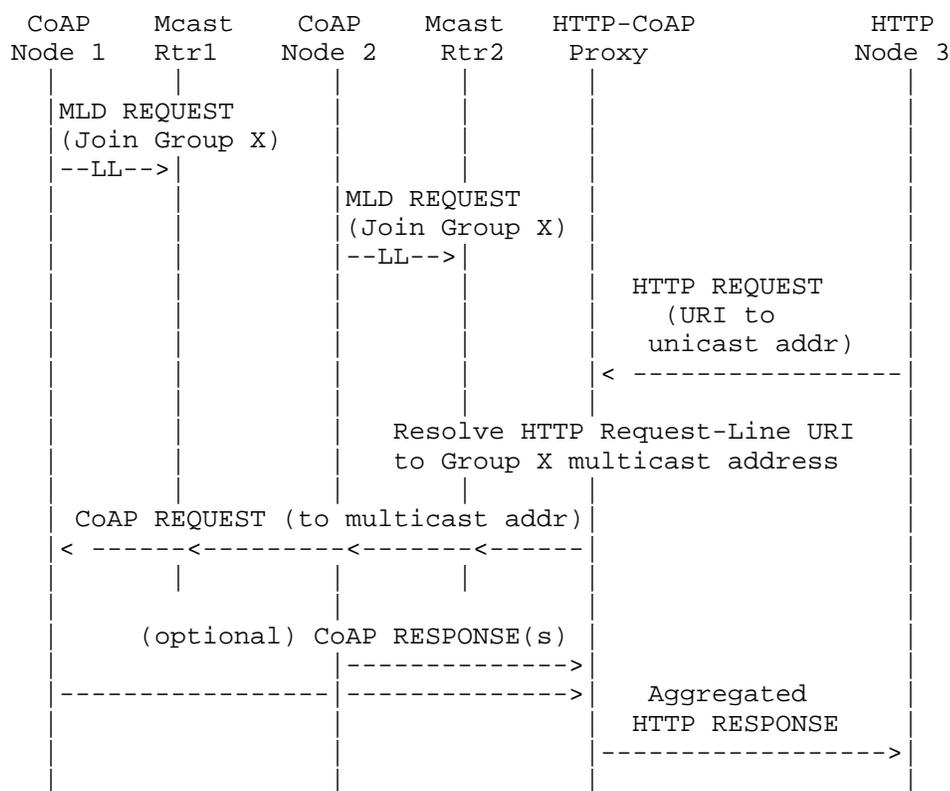

              Figure 3: CoAP Multicast and HTTP Unicast Interworking

   Note that Figure 3 illustrates the case of IP multicast as the
   underlying group communications mechanism. MLD denotes the Multicast
   Listener Discovery protocol ([RFC3810], Appendix A) and LL denotes a
   Link-Local multicast.

A key point in Figure 3 is that the incoming HTTP Request (from node
3) will carry a Host request-header field that resolves in the
general Internet to the proxy node. At the proxy node, this hostname
and/or the Request-Line URI will then possibly be mapped (as detailed
in [I-D.castellani-core-advanced-http-mapping]) and again resolved
(with the CoAP scheme) to an IP multicast address. This may be
accomplished, for example, by using DNS or DNS-SD (Section 7). The
proxy node will then IP multicast the CoAP Request (corresponding to
the received HTTP Request) via multicast routers to the appropriate
nodes (i.e. nodes 1 and 2).

In terms of the HTTP Response, Figure 3 illustrates that it will be
generated by the proxy node based on aggregated responses of the CoAP
nodes and sent back to the client in the general Internet that sent
the HTTP Request (i.e. node 1). In
[I-D.castellani-core-advanced-http-mapping] the HTTP Response that
the Proxy may use to aggregate multiple CoAP responses is described
in more detail. So in terms of overall operation, the CoAP proxy can
be considered to be a "non-transparent" proxy according to [RFC2616].
Specifically, [RFC2616] states that a "non-transparent proxy is a
proxy that modifies the request or response in order to provide some
added service to the user agent, such as group annotation services,
media type transformation, protocol reduction or anonymity
filtering."

An alternative to the above is using a Forward Proxy. In this case,
the CoAP request URI is carried in the HTTP Request-Line (as defined
in [I-D.ietf-core-coap] Section 10.2) in a HTTP request sent to the
IP address of the Proxy.

10. Acknowledgements

   Thanks to all CoRE WG members who participated in the IETF 82
   discussions, which was the trigger to initiate this document.

11. IANA Considerations

   This memo includes no request to IANA.

12. Security Considerations

   Security aspects of group communication for CoAP are discussed in
   [I-D.ietf-core-groupcomm]. The current document contains no new
   proposals yet, for which security considerations have to be analyzed
   here.

13. References

13.1. Normative References

   [I-D.ietf-core-coap]
              Shelby, Z., Hartke, K., Bormann, C., and B. Frank,
              "Constrained Application Protocol (CoAP)",
              draft-ietf-core-coap-13 (work in progress), December 2012.

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

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

   [RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security
              Architecture", RFC 3740, March 2004.

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

   [RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm,
              "Multicast Security (MSEC) Group Key Management
              Architecture", RFC 4046, April 2005.

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

   [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.

   [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler,
              "Transmission of IPv6 Packets over IEEE 802.15.4
              Networks", RFC 4944, September 2007.

   [RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast
              Extensions to the Security Architecture for the Internet
              Protocol", RFC 5374, November 2008.

   [RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen,
              "Building Automation Routing Requirements in Low-Power and
              Lossy Networks", RFC 5867, June 2010.

   [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R.,

               Levis, P., Pister, K., Struik, R., Vasseur, JP., and R.
               Alexander, "RPL: IPv6 Routing Protocol for Low-Power and
               Lossy Networks", RFC 6550, March 2012.

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

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

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

13.2. Informative References

   [Banerjee01]
               Banerjee, B. and B. Bhattacharjee, "A Comparative Study of
               Application Layer Multicast Protocols", 2001, <http://
               wmedia.grnet.gr/P2PBackground/
               a-comparative-study-ofALM.pdf>.

   [I-D.castellani-core-advanced-http-mapping]
               Castellani, A., Loreto, S., Rahman, A., Fossati, T., and
               E. Dijk, "Best Practices for HTTP-CoAP Mapping
               Implementation",
               draft-castellani-core-advanced-http-mapping-00 (work in
               progress), July 2012.

   [I-D.cheshire-dnsext-dns-sd]
               Cheshire, S. and M. Krochmal, "DNS-Based Service
               Discovery", draft-cheshire-dnsext-dns-sd-11 (work in
               progress), December 2011.

   [I-D.eggert-core-congestion-control]
               Eggert, L., "Congestion Control for the Constrained
               Application Protocol (CoAP)",
               draft-eggert-core-congestion-control-01 (work in
               progress), January 2011.

   [I-D.ietf-6lowpan-routing-requirements]
               Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem
               Statement and Requirements for 6LoWPAN Routing",
               draft-ietf-6lowpan-routing-requirements-10 (work in
               progress), November 2011.

   [I-D.ietf-core-block]
              Bormann, C. and Z. Shelby, "Blockwise transfers in CoAP",
              draft-ietf-core-block-10 (work in progress), October 2012.

   [I-D.ietf-core-groupcomm]
              Rahman, A. and E. Dijk, "Group Communication for CoAP",
              draft-ietf-core-groupcomm-03 (work in progress),
              October 2012.

   [I-D.ietf-core-observe]
              Hartke, K., "Observing Resources in CoAP",
              draft-ietf-core-observe-07 (work in progress),
              October 2012.

   [I-D.ietf-roll-trickle-mcast]
              Hui, J. and R. Kelsey, "Multicast Protocol for Low power
              and Lossy Networks (MPL)",
              draft-ietf-roll-trickle-mcast-02 (work in progress),
              October 2012.

   [I-D.lynn-core-discovery-mapping]
              Lynn, K. and Z. Shelby, "CoRE Link-Format to DNS-Based
              Service Discovery Mapping",
              draft-lynn-core-discovery-mapping-02 (work in progress),
              October 2012.

   [I-D.rahman-core-groupcomm]
              Rahman, A. and E. Dijk, "Group Communication for CoAP",
              draft-rahman-core-groupcomm-07 (work in progress),
              October 2011.

   [I-D.shelby-core-coap-req]
              Shelby, Z., Stuber, M., Sturek, D., Frank, B., and R.
              Kelsey, "CoAP Requirements and Features",
              draft-shelby-core-coap-req-02 (work in progress),
              October 2010.

   [I-D.shelby-core-resource-directory]
              Shelby, Z., Krco, S., and C. Bormann, "CoRE Resource
              Directory", draft-shelby-core-resource-directory-04 (work
              in progress), July 2012.

   [I-D.vanderstok-core-bc]
              Stok, P. and K. Lynn, "CoAP Utilization for Building
              Control", draft-vanderstok-core-bc-05 (work in progress),
              October 2011.

   [I-D.vanderstok-core-dna]

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

[Lao05]   Lao, L., Cui, J., Gerla, M., and D. Maggiorini, "A
            Comparative Study of Multicast Protocols: Top, Bottom, or
            In the Middle?", 2005, <http://www.cs.ucla.edu/NRL/hpi/
            AggMC/papers/comparison_gi_2005.pdf>.

Appendix A. Multicast Listener Discovery (MLD)

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

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

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

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

Appendix B. CoAP-Observe Alternative to Group Communication

   The CoAP Observation extension [I-D.ietf-core-observe] can be used as

a simple (but very limited) alternative for group communication. A
group in this case consists of a CoAP server hosting a specific
resource, plus all CoAP clients observing that resource. The server
is the only group member that can send a group message. It does this
by modifying the state of a resource under observation and
subsequently notifying its observers of the change. Serial unicast
is used for sending the notifications. This approach can be a simple
alternative for networks where IP multicast is not available or too
expensive.

The CoAP-Observe approach is unreliable in the sense that, even
though Confirmable CoAP messages may be used, there are no guarantees
that an update will be received. For example, a client may believe
it is observing a resource while in reality the server rebooted and
lost its listener state.

Authors' Addresses

    Esko Dijk (editor)
    Philips Research

    Email: esko.dijk@philips.com

    Akbar Rahman (editor)
    InterDigital Communications, LLC

    Email: Akbar.Rahman@InterDigital.com