Abstract

This is a working document intended to trigger discussion and develop draft text for the CoAP protocol specification in the area of group communication. Engineering tradeoffs become more challenging in constrained environments, therefore group communication is considered within the context of adjacent topics that may impact or be impacted by design choices in the subject area. A solution based on IP multicast is proposed.

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 April 14, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Conventions and Terminology ............................... 3
2. Introduction .............................................. 3
  2.1. Background ........................................ 3
  2.2. Problem Statement and Scope .......................... 4
3. Potential Solutions ....................................... 5
  3.1. Overview ............................................ 5
  3.2. Requirements ........................................ 6
  3.3. IP Multicast ........................................ 8
    3.3.1. Multicast Listener Discovery (MLD) and Multicast
           Router Discovery (MRD) .......................... 8
    3.3.2. Group URIs and Multicast Addresses ............... 9
    3.3.3. Group Discovery ................................ 9
    3.3.4. Group Resource Manipulation ...................... 10
    3.3.5. IP Multicast Transmission Methods ............... 11
    3.3.6. Congestion Control ................................ 12
  3.4. Overlay Multicast ..................................... 13
  3.5. CoAP Application Layer Group Management ............ 13
  3.6. CoAP Multicast and HTTP Unicast Interworking ....... 16
  3.7. CoAP-Observe for Group Communication ............... 18
4. Recommended Solution ....................................... 18
  4.1. Overview ............................................. 19
  4.2. An Example Protocol Flow ............................. 19
  4.3. Implementation in Target Network Topologies ....... 22
    4.3.1. Single LLN Topology ............................ 23
    4.3.2. Single LLN with Backbone Topology ............. 25
    4.3.3. Multiple LLNs with Backbone Topology .......... 27
    4.3.4. LLN(s) with Multiple 6LBRs ...................... 27
    4.3.5. Conclusions ..................................... 27
  4.4. HTTP/CoAP Interworking Aspects ....................... 28
  4.5. Implementation Considerations ......................... 28
    4.5.1. MLD Implementation on LLNs ..................... 28
    4.5.2. 6LBR Implementation ............................ 29
    4.5.3. Backbone IP Multicast Infrastructure ............ 29
  5. Security Considerations ................................. 30
  6. IANA Considerations ..................................... 31
  7. Conclusions ............................................. 31
  8. Acknowledgements ....................................... 31
  9. References .............................................. 32
    9.1. Normative References ................................ 32
    9.2. Informative References ............................. 33
Authors’ Addresses ............................................ 35
1. Conventions and Terminology

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 [RFC2119].

The following are definitions of specific terminology used in this draft.

Group Communication: A source node sends a message to more than one destination node, where all destinations are identified to belong to a specific group. The set of source nodes and destination nodes may consist of an arbitrary mix of constrained and non-constrained nodes. Sending methods may include serial unicast, multicast, or hybrid unicast-to-multicast solutions.

Multicast: Sending a message to multiple receiving nodes simultaneously. Typically, this is done as part of a group communication process. There are various options to implement multicast including layer 2 (Media Access Control) or layer 3 (IP) mechanisms.

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

Low power and Lossy Network (LLN): LLNs are made up of constrained devices. These devices may be interconnected by a variety of links, such as IEEE 802.15.4, Bluetooth, WiFi, wired or low-power powerline communication links.

2. Introduction

2.1. Background

The CoRE working group is chartered to design and standardize a Constrained Application Protocol (CoAP) for resource constrained devices and networks [I-D.ietf-core-coap]. The requirements for CoAP are documented in [I-D.shelby-core-coap-req].

Constrained devices can be large in number, but highly correlated to each other. For example, all the light switches in a building may belong to one group and all the thermostats belong to another group. All the smart meters in the same region can belong to a group as well. Groups may be composed by function; for example, the group "all lights in building one" may consist of the groups "all lights on
floor one of building one", "all lights on floor two of building one", etc. Groups may also be configured or dynamically formed.

In this draft, we focus and expand discussions on the requirements pertaining to CoAP "group communication" and "multicast" support including:

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:

"CoAP supports sending messages to multicast destination addresses. Such multicast messages MUST be Non-Confirmable. 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).

2.2. Problem Statement and Scope

In this draft, we expand the scope from unreliable IP multicast in the current CoAP requirement to group communication, using either (reliable or unreliable) multicast or unicast or combinations thereof. We assume that all, or a substantial part of, devices participating in group communication are constrained devices (e.g. such as Low Power and Lossy Network (LLN) devices).

Machine-to-Machine (M2M) networks may contain groups of nodes that are highly correlated (e.g. by type or location). For example, all smart meters in a region may belong to one group, and all light switches in a building control system belong to another. Group communication mechanisms can improve efficiency and latency of communication and reduce bandwidth requirements for a given application.

In the following sections, we address the issues related to group communication in detail, with requirements, proposed solutions and analysis of their impact to the CoAP protocol and implementations.
3. Potential Solutions

3.1. Overview

The classic concept of group communications is that of a single source distributing content to multiple recipients that are all part of a group, as shown in the example sequence diagram in Figure 1. Also shown there is the pre-requisite step of forming the group before content can be distributed to it.

Group communication solutions have evolved from "bottom" to "top", i.e., from the network layer (IP multicast) to application layer group communication, also referred to as application layer multicast. A study published in 2005 [STUDY1] 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 [STUDY1] using metrics such as link stress and level of host complexity [STUDY2]. The results show for a realistic internet topology that IP Multicast is most resource-efficient, with the only downside being that it requires most effort to deploy in the infrastructure.

The approach adopted in this section is to begin with group communication requirements. This is followed by the solutions of IP multicast, an overlay multicast solution, and application layer group communication. Finally additional topics are covered such as group management and CoAP/HTTP proxies in group communication.
3.2. Requirements

Requirements that a group communication solution in CoRE should fulfill can be found in existing documents [RFC 5867] [draft-ietf-6lowpan-routing-requirements] [I-D.vanderstok-core-bc] [I-D.shelby-core-coap-req]. Below, a set of high-level requirements is listed that a group communication solution in CoRE should ideally fulfill. More precise requirements may depend on the chosen application (area).
A CoRE group communication solution should (ideally) offer:

REQ 1: Optional Reliability: unreliable group communication, with preferably reliable group communication as an option.

REQ 2: Efficiency: delivers messages more efficiently than a "serial unicast only" solution. Also, it should provide a right balance between group data traffic and control overhead.

REQ 3: Low latency: deliver a message (preferably) as fast as possible.

REQ 4: Synchrony: allows near-simultaneous modification of a resource on all devices in a group, providing to users a perceived effect of synchrony or simultaneity. It can be expressed as a time span D such that message m is delivered to all destinations in a time interval [t, t+D] for arbitrary t.

REQ 5: Ordering: message ordering in the reliable group communication mode.

REQ 6: Security: see Section 5 for security requirements for group communication.

REQ 7: Flexibility: support for one or many source(s), for dense and sparse networks, for high or low listener density, one or many group(s), and multi-group membership.

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

REQ 9: Network layer independence: a solution should be specified independent from specific unicast and/or IP multicast routing protocols. It should support different routing protocols and implementations thereof.

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.

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.

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).
REQ 13: CoAP Proxying support: a CoAP proxy can handle distribution of a message to a group on behalf of a (constrained) CoAP client.
REQ 14: Suitable for operation on LLNs with constrained nodes.

3.3. IP Multicast

IP Multicast protocols have been evolving for decades, resulting in proposed standards such as Protocol Independent Multicast - Sparse Mode (PIM-SM) [RFC4601]. Yet, due to various technical and marketing reasons, IP Multicast is not widely deployed on the general Internet. However, IP Multicast is popular in specific deployments such as in enterprise networks (e.g. for video conferencing or general IP multicast PC applications within a single LAN broadcast domain) and carrier IPTV deployments. Therefore, the packet economy and minimal host complexity of IP multicast make it worth investigating for group communication in constrained environments.

3.3.1. Multicast Listener Discovery (MLD) and Multicast Router Discovery (MRD)

The Multicast Listener Discovery (MLD) protocol [RFC3810] (or its IPv4 pendant IGMP) is today the method of choice used by an (IP multicast enabled) IPv6 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. It 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 a 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 switch 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 an MLD Router) if no MLD router is present.

The Multicast Router Discovery (MRD) protocol [RFC4286] defines a way to discover multicast routers, for the purpose of using this information by IGMP/MLD snooping devices. However, it appears that this protocol is not as commonly implemented in existing products as MLD is.
3.3.2. Group URIs and Multicast Addresses

An approach to map group authorities onto IP multicast addresses using DNS was proposed in [I-D.vanderstok-core-bc]. Examples of group URI naming (and scoping) for a building control application are shown below. Group URIs MUST follow the approach of the URI structure defined in [RFC3986].

<table>
<thead>
<tr>
<th>Authority</th>
<th>Targeted Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.bldg6...</td>
<td>&quot;all nodes in building 6&quot;</td>
</tr>
<tr>
<td>all.west.bldg6...</td>
<td>&quot;all nodes in west wing, building 6&quot;</td>
</tr>
<tr>
<td>all.floor1.west.bldg6...</td>
<td>&quot;all nodes in floor 1, west wing, etc.&quot;</td>
</tr>
<tr>
<td>all.bu036.floor1.west.bldg6...</td>
<td>&quot;all nodes in office bu036, floor1, etc.&quot;</td>
</tr>
</tbody>
</table>

The authority portion of the URI is used to identify a node (or group) and the resulting DNS name is bound to a unicast or multicast IP address. Each example group URI shown above can be mapped to a unique multicast IP address. This may be an address allocated according to [RFC3956], [RFC3306] or [RFC3307].

3.3.3. Group Discovery

CoAP defines a resource discovery capability but, in the absence of a standardized group communication infrastructure, it is limited to link-local scope IP multicast; examples may be found in [I-D.ietf-core-link-format]. A service discovery capability is required to extend discovery to other subnets and scale beyond a certain point, as originally proposed in [I-D.vanderstok-core-bc].

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 or alternatively 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 subtypes 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>.

3.3.4. Group Resource Manipulation

At least two forms of group resource manipulation must be supported. The first is push (multicast PUT or MPUT for short) as e.g. "turn off all the lights simultaneously". Logically, this is similar to publishing a value to multiple subscribers. The second operation is pull (multicast GET or MGET), which is essential for discovery during commissioning and can be illustrated by the example "return all the resources on all CoAP servers advertised by their .well-known/core URI". MGET to an "all-nodes" or "all-CoAP-nodes" multicast IP address should perhaps be limited in scope to link-local multicast for scaling [TBD: and possibly for security reasons, e.g. DoS attacks].

Conceptually, the result of a multicast GET or PUT should be the same as if the client had unicast them serially (that is, a set of {URI, representation} tuples). Practically, there are major benefits to avoiding serial unicast in favor of a multicast CoAP GET/PUT solution:

- packet economy on constrained networks
- M2M resource discovery (solves the "chicken-and-egg" problem)
- apparent simultaneity of events (e.g. in lighting applications)
- average lower latency per event (e.g. in lighting applications)

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 Section 3.3.3 and [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 requests.

A second solution is to impose the following restrictions, e.g. for groups not found using, or advertized in, DNS-SD:
- All CoAP multicast requests MUST be sent to the well-known CoAP port.
- All CoAP multicast requests SHOULD operate on /.well-known/core URIs

One question is whether the application (or middleboxes) need to be aware that a request is intended for a group. A separate scheme as proposed by [ID.goland-http-udp] might be useful (e.g. "corem" vs. "core"). To the extent that group membership might be implemented as a series of IP multicast, serial unicast, or some combination, having a distinct scheme for group operations might be a useful signal for a proxy receiving the request to look up the group membership and replicate serial unicasts as well as send multicast packets.

3.3.5. IP Multicast Transmission Methods

3.3.5.1. 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.

3.3.5.2. 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.

3.3.5.3. 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):

- 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.
- Integrity - destination receives m at most once from sender and only if sender sent m to a group including destination.
- Agreement - If a correct destination of g receives m, then all correct destinations of g receive m.
- 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:

- 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.
- Route redundancy
- Source node transmits multiple times (destinations do not respond)

3.3.6. Congestion Control

CoAP requests may be multicast, resulting a multitude of replies from different nodes, potentially causing congestion. [I-D.eggert-core-congestion-control] suggests to conservatively control sending multicast requests.

CoAP already addresses the congestion problem to some extent by requiring all multicast CoAP requests to be Non-Confirmable. However, as responses to multicast requests (both MGET or MPUT) are required in CoAP, using CoAP multicast still may lead to congestion issues.
Various means can be implemented to prevent congestion.

[TBD: if an MGET or MPUT request leads to the sending of a CoAP response by servers, the servers should enforce a random delay within TIMEOUT before sending their responses. More investigation required.]

Currently in the CoAP protocol, a MAX_RETRANSMIT value set by default to 4 is used for retransmission of Confirmable messages. Since CoAP multicast messages are Non-Confirmable, no retransmissions will occur in CoAP, making the effective retransmission value 0.

3.4. 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 (Section 3.3.1) 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.

3.5. 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:
- discover groups;
- query group properties (e.g. related resource descriptions);
- create a group;
- remove a group;
- add a group member;
- remove a group member;
- enumerate group members;
- 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 2:
<table>
<thead>
<tr>
<th>Type</th>
<th>C/E</th>
<th>Name</th>
<th>Data type</th>
<th>Length</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Group Join</td>
<td>String</td>
<td>1-270</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group Leave</td>
<td>String</td>
<td>1-270</td>
<td>&quot;&quot;</td>
</tr>
</tbody>
</table>

**Figure 2: CoAP Header Options for Group Management**

Figure 3 illustrates how a node can join or leave a group using the Header Options in a CoAP message:

```
+--------+--------+-----------------+--------+--------+-----------------+--------+--------+-----------------+--------+--------+-----------------+
| Ver    | T      | OC              | Code   |         | Message ID      |         |         | Message ID      |         |         | Message ID      |
|--------+--------+-----------------+--------+--------+-----------------+--------+--------+-----------------+--------+--------+-----------------+
|        | delta  | length          | Join Group A (ID or URI) |        |                  |         |         | Join Group B (ID or URI) |        |                  |         |         | Leave Group C (ID or URI) |
|        | 0      | length          | Join Group A (ID or URI) |        |                  |         |         | Join Group B (ID or URI) |        |                  |         |         | Leave Group C (ID or URI) |
|        | 2      | length          | Leave Group C (ID or URI) |        |                  |         |         | Leave Group C (ID or URI) |        |                  |         |         | Leave Group C (ID or URI) |
+--------+--------+-----------------+--------+--------+-----------------+--------+--------+-----------------+

**Figure 3: 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 3, 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 advertised by DNS-SD. For example, to join group1 and leave group2:

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

### 3.6. CoAP Multicast and HTTP Unicast Interworking

Within the constrained network, CoAP runs over UDP for which IP multicast is supported. In a non-constrained network (i.e. general Internet), HTTP over TCP is used for which IP multicast is not supported. Therefore a CoAP/HTTP Proxy node that supports group communication needs to have functionalities to support interworking of unicast and multicast. One possible way of operation of the Proxy is illustrated in Figure 4. Note that this topic is covered in more detail in [I-D.castellani-core-http-mapping].
Figure 4: CoAP Multicast and HTTP Unicast Interworking

Note that Figure 4 illustrates the case of IP multicast as the underlying group communications mechanism. However the overlay multicast group communication (Section 3.4) or CoAP application group communication (Section 3.5) can be used as the underlying mechanism and the principles of the figure would still apply (i.e. CoAP proxy needs to do interworking between HTTP unicast and CoAP multicast).

A key point in Figure 4 is that the incoming HTTP Request (from node 3) will carry a URI (with the HTTP scheme) that resolves in the general Internet to the proxy node. At the proxy node, the URI will
then possibly be mapped (as detailed in [I-D.castellani-core-http-mapping]) and again resolved (with the CoAP scheme) to an IP multicast destination. This may be accomplished, for example, by using DNS-SD (Section 3.3.3). The proxy node will then IP multicast the CoAP Request (corresponding to the received HTTP Request) to the appropriate nodes (i.e. nodes 1 and 2).

In terms of the HTTP Response, Figure 4 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-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 could be carried in the HTTP Request Line (as defined in [I-D.ietf-core-coap] Section 8) in a HTTP request sent to the IP address of the Proxy.

3.7. CoAP-Observe for Group Communication

The CoAP Observation extension [I-D.ietf-core-observe] can be directly used for group communication. A group then 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 in this case for notifications.

Group communication 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.

4. Recommended Solution
4.1. Overview

We recommend that IP multicast as outlined in Section 3.3 be adopted as the base solution for CoAP Group Communication. This approach re-uses the IP multicast suite of protocols and can operate on both constrained and non-constrained network segments. The group communication can hence work regardless of the underlying networking technology. 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.

4.2. An Example Protocol Flow

We first present an example use case to illustrate the overall steps in an IP Multicast based CoAP Group Communication solution. We assume the following network configuration for this example (see Figure 5):

1) A large room (Room-A) with three lights (Light-1, Light-2, Light-3) controlled by a Light Switch. The devices are organized into two 6LoWPAN subnets.

2) Light-1 and the Light Switch are connected to a router (Rtr-1) which is also a CoAP Proxy and a 6LoWPAN Border Router (6LBR).

3) Light-2 and the Light-3 are connected to another router (Rtr-2) which is also a CoAP Proxy and a 6LBR.

4) The routers are connected to an IPv6 network backbone which is also multicast enabled. In the general case, this means the network backbone and 6LBRs support a PIM based multicast routing protocol, and MLD for forming groups. In a limited case, if the network backbone is one link, then the routers only have to support MLD-snooping for the example use case to work.
Figure 5: Network Topology of a Large Room (Room-A)
The corresponding protocol flow for an IP Multicast based CoAP Group Communication solution for the network shown in Figure 5 is shown in Figure 6. We assume the following steps occur before the illustrated flow:

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

2) Commissioning phase (by applications): The IP multicast address of the group (Room-A-Lights) has been set in all the Lights. The URI of the group (Room-A-Lights) has been set in the Light Switch.
Figure 6: Turning on Lights in a Large Room (Room-A)

The indicated MLD Report messages are link-local multicast. In each LoWPAN, it is assumed that a multicast routing protocol in 6LRs will propagate the Join information over multiple hops to the 6LBR.

4.3. 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.)

4.3.1. 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.

4.3.1.1. 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 Section 3.3.1). 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.

4.3.1.2. RPL Multicast Routing

The RPL routing protocol for LLNs provides support for routing to multicast IP destinations (Section 12 of [I-D.ietf-roll-rpl]). 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.

4.3.1.3. 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 use link-local RPL routers for their IP connectivity. Note that the current RPL specification [I-D.ietf-roll-rpl] considers this case to be out of scope. However, it was suggested on the ROLL mailing list that RPL could potentially be run with non-RPL-aware hosts but that it is simply not specified yet. Such non-RPL hosts can’t advertise their IP multicast groups of interest via RPL DAO messages as defined above. Therefore in that case MLD can 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 4.5.1.

4.3.1.4. 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.

4.3.1.5. 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 4.5.1 for more efficient substitutes for MLD targeted towards a LLN context.

4.3.2. 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.

4.3.2.1. 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 learnt 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 4.5.1.

4.3.2.2. 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.

4.3.2.3. 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 4.3.1.3.

4.3.2.4. 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 4.3.2.1.

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 4.3.2.1) of potentially overloading the LLN with unwanted backbone IP multicast traffic appears again.

A possible solution to this is (again) to let multicast listeners advertize their interest using MLD as described in Section 4.3.2.1 or to use an MLD alternative suitable for LLNs as described in Section 4.5.1. However, following this approach requires possibly an
extension to Trickle Multicast Forwarding: the protocol should ensure
that MLD-advertized information is somehow communicated to the 6LBR,
possibly over multiple hops. MLD itself supports link-local
communication only.

4.3.2.5. 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.

4.3.3. 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.

4.3.4. 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? ]

4.3.5. 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 interwork with regular MLD on the backbone network. Such MLD
variants are further analyzed in Section 4.5.1.
4.4. HTTP/CoAP Interworking Aspects

The topic of HTTP unicast to CoAP multicast request proxying is treated in [I-D.castellani-core-http-mapping]. [TBD: only if needed more information will be added here in the future.]

4.5. Implementation Considerations

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

4.5.1. MLD Implementation on LLNs

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 learnt (“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.

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 [I-D.ietf-6lowpan-nd] 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 unicast IP address, a host "registers" its interest in a multicast IP address.
Unlike ARO, multiple MLO can be used in the same ND packet. A registration period is also defined 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 full MLD.

[ 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 ]

4.5.2. 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.

4.5.3. 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-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 & deployed IP (Ethernet/WiFi/...) infrastructure, to be shared with other IP devices (PCs)
3. Newly designed & 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.]

5. Security Considerations

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.

Note that some of the requirements are marked optional. This means that, depending on the use case, these may be required or not. For this purpose each use case can be associated to a security profile as specified in [I-D.garcia-core-security]. The security profile prescribes what requirements should be taken into account for this profile. A mapping of these requirements to these profiles has not yet been done.

REQ1- 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.

REQ2- 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.

REQ3- 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.

REQ4- 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.

REQ5- 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.

REQ6- 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 [I-D.ietf-roll-rpl]. 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.

REQ7- 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.

6. IANA Considerations

This document makes no request of IANA.

7. Conclusions

Three solutions for enabling CoAP group communications have been discussed.

Unreliable IP multicast as outlined in Section 3.3 is recommended to be adopted as the base solution for CoAP Group Communication on LLNs. This approach requires no standards changes to the IP multicast suite of protocols and it provides interoperability with IP multicast group communication on unconstrained backbone networks.

The proposals for group communication described in this draft should be considered for incorporation into the overall CoAP protocol specification.

8. Acknowledgements

Thanks to Peter Bigot, Carsten Bormann, Anders Brandt, Angelo
9. References

9.1. Normative References


9.2. Informative References

[I-D.cheshire-dnsext-dns-sd]  

[I-D.eggert-core-congestion-control]  

[I-D.ietf-6lowpan-hc]  

[I-D.ietf-6lowpan-nd]  

[I-D.ietf-core-coap]  
[I-D.ietf-core-link-format]
Shelby, Z., "CoRE Link Format",
draft-ietf-core-link-format-07 (work in progress),
July 2011.

[I-D.ietf-core-observe]
Hartke, K. and Z. Shelby, "Observing Resources in CoAP",
draft-ietf-core-observe-02 (work in progress), March 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.vanderstok-core-bc]
Stok, P. and K. Lynn, "CoAP Utilization for Building
Control", draft-vanderstok-core-bc-04 (work in progress),
July 2011.

[I-D.castellani-core-http-mapping]
Castellani, A., Loreto, S., Rahman, A., Fossati, T., and
E. Dijk, "Best practices for HTTP-CoAP mapping
implementation", draft-castellani-core-http-mapping-01
(work in progress), July 2011.

[I-D.garcia-core-security]
Garcia-Morchon, O., Keoh, S., Kumar, S., Hummen, R., and
R. Struik, "Security Considerations in the IP-based
Internet of Things", draft-garcia-core-security-02 (work
in progress), July 2011.

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J.,
Kelsey, R., Levis, P., Pister, K., Struik, R., and J.
Vasseur, "RPL: IPv6 Routing Protocol for Low power and
Lossy Networks", draft-ietf-roll-rpl-19 (work in
progress), March 2011.

[I-D.ietf-roll-trickle-mcast]
Hui, J. and R. Kelsey, "Multicast Forwarding Using
Trickle", draft-ietf-roll-trickle-mcast-00 (work in
progress), April 2011.

[1D.goland-http-udp]
Goland, Y., "Multicast and Unicast UDP HTTP Messages",
1999,


Authors’ Addresses

Akbar Rahman (editor)
InterDigital Communications, LLC

Email: Akbar.Rahman@InterDigital.com

Esko Dijk (editor)
Philips Research

Email: esko.dijk@philips.com