       Publish-Subscribe Deployment Option for NDN in the Constrained Internet
                                of Things
                   draft-gundogan-icnrg-pub-iot-02

Abstract

   Constrained IoT devices often operate more efficiently in a loosely
   coupled environment without maintaining end-to-end connectivity
   between nodes. Information Centric Networking naturally supports
   this demand by replicated data distribution and hop wise forwarding.
   This document outlines a deployment option for NDN in low-power and
   lossy networks (LLNs) that follows a publish-subscribe pattern. The
   proposed protocol scheme simplifies name-based routing significantly
   and facilitates even large off-duty cycles for constrained nodes.

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 https://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 September 6, 2018.

Table of Contents

1. Introduction

   In the emerging Internet of Things (IoT), it is expected that large
   quantities of very constrained sensors and actuators collect,
   communicate, and process massive amounts of machine data. Early
   experiments with constrained nodes show promising results for
   different deployments of ICN communication [NDN-EXP]. NDN stacks for
   the IoT tend to be less resource-consuming and more lightweight, the
   hop wise caching and forwarding indicate higher reliability and lower
   sensitivity to disruptions than end-to-end IP solutions. Most
   notably, security and resiliance of nodes and networks are enhanced
   by the NDN request-response scheme, where data delivery remains
   receiver-initiated.

   The following properties are characteristic for constrained IoT
   environments:

   o Battery-powered nodes with significant sleep cycles

   o Low memory reserves, but potentially large (flash) storage

o Failing nodes due to various external or internal reasons

o Low power lossy wireless networks

These various constraints pose significant challenges on a real-world
deployment of NDN networks, among them [RFC7927]:

o The complexity of name-based routing may overburden link and
   memory capacities.

o State management at nodes may drain batteries or memory resources.

o Clear separation between control and data plane potentially
   increases

o An adaptation to the constrained wireless link and transmission
   characteristics will often be required (see
   [I-D.gundogan-icnrg-ccnlowpan]).

o Mobility management will need to aid non-stationary deployments.

o Intermittent connectivity may occur from mobility or temporarily
   fragmenting networks.

1.1. Baseline Scenarios

Multiple scenarios have been discussed in [RFC7476] and [IWMT] that
evaluate the applicability of ICN in IoT.

We consider two characteristic constrained IoT scenarios with the
enumerated challenges below.

Stationary IoT nodes within reach of fixed uplinks for home,
   building, and factory automation, stationary monitoring, and
   related stable environments:

   * Reliability, resilience of operation

   * Radio coordination, coverage

   * Energy constraints, device lifetime

   * Interference with rivaling appliances

Mobile IoT nodes with sparse coverage or intermittent connectivity
   for urban or rural mobility and sensing, industrial Internet in
   widespread environments, disaster recovery and rescue, and related
   unstable environments:

* Exploit connectivity when available

* Large off-duty cycles of nodes

* Partitioned networks

* Limited dependability

* Environmental impact and disturbance

IoT scenarios usually impose routing requirements to support mobile
nodes, handle failing links and to be resilient against attacks. A
secure and autonomous bootstrapping is essential, especially for
large-scale IoT deployments.

1.2. Benefits of Loose Coupling in the IoT

ICN decouples content consumers from data producers (decoupling in
space). A more sophisticated decoupling can be provided with the
publish-subscribe messaging pattern that further adds a decoupling in
time and synchronization. Constrained devices in LLNs can leverage
this loose coupling to increase sleep cycles and delegate the
authority over as much information as possible to more powerful
devices that act as content proxies. In Figure 1, once content is
published to the content proxy (CP) by a producer (P), consumers (C)
can retrieve this content from (CP) without interacting with the
producer. This indirection when retrieving information allows (P) to
align sleep cycles accordingly to the period of generating new sensor
readings, instead of handling content requests from any consumers
(C).

```
                    (CP)
                   / | \
                  /  |   `-----.
                 /   | | |   \
                (P) (C) ...... (C)
```
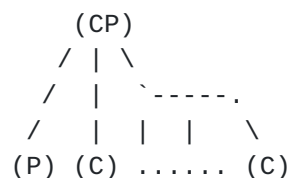
Figure 1: Content Proxy (CP) - Producer (P) - Consumer (C)

The NDN communication follows a request-response pattern and prevents
an unsolicited push of data. While this design must be seen as a
vital security feature that prevents common DDoS attacks, no
straight-forward mechanism for issuing an unsolicited alert between
nodes has been foreseen which is desired in many use cases of sensors
and actuators. Several extensions have been proposed to enable an
unsolicited push of data, among them Interest-follows-Interest,
Interest notification, and a dedicated push packet. All these push
messages are sent immediately to a prospective consumer node, which

not only conflicts with the ICN paradigm of naming content instead of hosts, but re-open the DoS attack surface.

In this memo, we introduce the different approach of a (local) 'Hop and Pull' [HoPP]. We assume the presence of a link-local control message for alerting next-hop neighbors. Such single-hop alerts advertise names (NAM-messages), or prefixes (PAM-mesage). Nodes that receive such local control information may either update their control plane (routing tables) or decide to pull content from its neighboring nodes. In this way, the NDN request-response pattern for data transmission remains fully intact, while control functions and time-critical update alerts become available as first class citizens in the ICN scheme.

2. 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 RFC 2119 [RFC2119]. The use of the term, "silently ignore" is not defined in RFC 2119. However, the term is used in this document and can be similarly construed.

This document uses the terminology of [RFC7476], [RFC7927], and [RFC7945] for ICN entities.

The following terms are used in the document and defined as follows:

Converge Cast Many-to-One communication pattern, where multiple
   devices send sensor readings to one gateway.

Content Proxy Stable node for replicating content.

Cloud Gateway A Gateway that enables content transfer to and from a
   remote cloud storage, possibly by performing some kind of protocol
   translation.

PAM Prefix Advertisement Message.

NAM Name Advertisement Message.

3. Publish-Subscribe in IoT Edge Networks

The publish-subscribe system is centered around prefix-specific content proxies (CPs) that are deployed in IoT edge networks. Such proxy function can be hosted on the Cloud- or Internet Gateway, or may reside on a stable, less constrained node within the IoT

infrastructure. It is assumed that a CP is present for each prefix
covering publishable content.

Implementing a pub-sub NDN involves several steps that are bound to
the tight requirements of resource-constrained devices. These steps
include:

1. Building the prefix-specific routing topology tailored to
    constrained networks

2. Mapping _Publish_ to NDN semantics

3. Mapping _Subscribe_ to NDN semantics

3.1. Topology Maintenance and Routing

A (sensor) node that wants to publish a data item needs to rely on
path information towards the Content Proxy. Following the approach
of PANINI [PANINI], default routes will be established as follows.

Each CP in the local IoT sub-network advertises the prefix(es) it
represents to the routing system. It does so by broadcasting Prefix
Advertisement Messages (PAMs) on the link layer (see Section 4 for
the corresponding protocol details). Nodes that newly receive PAM
advertisements will add or refresh a prefix-specific default route in
their FIB. Intermediate nodes in a multi-hop environment also re-
broadcast PAMs, so that the entire sub-network is flooded and default
route entries build a shortest path tree (SPT) towards the CP as
shown in Table 1 (alternatively, a DODAG w.r.t. a gateway for
redundant CPs).

```
                    (CP)
                    /  \
                PAM /    \ PAM
                 (A)    (B)
                 /|\    /|\
                : : :  : : : PAM
```
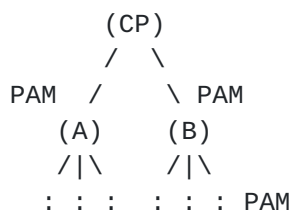
    Figure 2: SPT building by Prefix Advertisement Messages (PAMs)

Information flowing from constrained sensor nodes towards a gateway
is the prevalent communication pattern in the IoT (converge cast).
The publish-subscribe system hence establishes a default routing (see
sample FIB in Table 1) and uses the tree topology with default routes
towards the CP as a first step of content aggregation. Content
replication towards other CPs, an Internet gateway, or into a cloud
can follow subsequently.

```
+--------+------+----------+
| Prefix | Face | Lifetime |
+--------+------+----------+
| /P/    | Fx   |    Ft    |
| ...    | ...  |   ...    |
+--------+------+----------+
```

Table 1: FIB with a prefix-specific default route

It is noteworthy that the role of the new PAM message remains
orthogonal to the existing Interest or Data semantics. A PAM never
carries data nor requests, but resides on the control plane of name-
based routing. User applications stay unaffected, and continue to
rely on the NDN-specific request-response paradigm.

Each node in the tree calculates a rank based on the rank of its
parent and a routing metric. Hence, the rank is an indication for
the position within the tree and is strictly monotonically increasing
in the downwards direction from root to leaf. For simplicity, this
document uses the hop-count routing metric to calculate the rank.
Future work can focus on more elaborate routing metrics, e.g., to
reduce packet retransmissions or improve load balancing for publish
operations.

The Routing Information Base (RIB) contains the following
information:

SPT

   Prefix:
      Variable length prefix to configure a prefix-specific default
      route

   Rank:
      16-bit unsigned integer indicating a node's rank

   Flags:
      8-bit unsigned integer to store SPT related flags

   Parent
      The appropriate face towards the parent node is stored.

NAM Cache (NC)
   Entries in the NAM Cache are _<name,downstream_face>_ tuples. The
   NC is consolidated when propagating name advertisements.

3.2. Mapping Publish to NDN

In classical publish-subscribe systems, a Publish is typically
implemented as a push mechanism on the data plane. However, this
contradicts the request-response paradigm employed by NDN. To adapt
the Publish operation to NDN semantics, it is split into two phases
and the required push mechanism is performed on the control plane
with a link-local scope. The two phases consist of:

1. Announcing names of Named Data Objects (NDO) to neighbors on the
    control plane

2. Requesting NDOs on the data plane

In the first phase, the name of the NDO to publish is advertised to
the prefix-specific upstream neighbor by encoding it with TLV format
in a unicast link-local Name Advertisement Message (NAM) adopted from
PANINI [PANINI]. Once an upstream neighbor receives an unsolicited
NAM, the name is extracted and along with the incoming face stored in
the NAM Cache (NC) as a _<name,downstream_face>_ tuple. This link-
local content signaling does not establish any data paths in the PIT,
nor does it modify the FIB or the Content Store (CS).

In the second phase and as a result of an unsolicited NAM, the
content is requested from the downstream neighbor according to the
standard NDN scheme with an Interest message for the name recorded in
the NC on the recorded face (see [HoPP]). When a downstream neighbor
replies with the content (Data message), this neighbor removes the
corresponding NC entry and the NDO to publish is successfully
replicated one hop closer towards the content proxy. NC entries are
thus short-lived for hop-wise replication only.

Both phases are iteratively repeated hop-wise until the content proxy
is reached. Being the root of this prefix-specific spanning tree,
the content proxy has no further prefix-specific upstream neighbor
and the replication terminates. It is noteworthy, that both phases
can be interleaved, so that NAMs are signaled to the upstream
neighbor, while the content is requested from the downstream
neighbor.

Figure 3 (a) depicts the hop-wise replication of a published content
on the gradient towards the (CP). In this example, the name _/HAW/
temp_t_ is advertised by the producer (P) to its parent node (A).
(A) extracts the name from the NAM and stores it along with the
incoming face in its NC. (A) then propagates the NAM further to its
parent (CP) and simultaneously requests the content from (P) as
depicted in Figure 3 (b). Likewise in Figure 3 (c), (CP) reacts to

   the incoming NAM by requesting the content from (A). NC states from
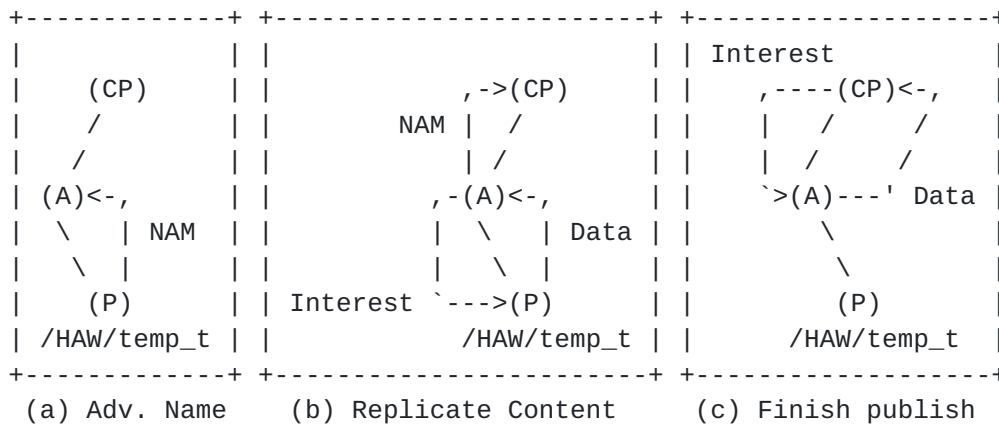   (A) and (P) are removed as soon as they respond to the request.

```
      +-------------+ +-----------------------+ +-------------------+
      |             | |                       | | Interest          |
      |    (CP)     | |            ,->(CP)     | |    ,----(CP)<-,   |
      |     /       | |       NAM |   /        | |    |  /    /     |
      |    /        | |           |  /         | |    | /    /      |
      |  (A)<-,      | |        ,-(A)<-,        | |    `>(A)---' Data |
      |   \   | NAM  | |        |  \   | Data  | |       \           |
      |    \  |      | |        |   \  |       | |        \          |
      |     (P)     | | Interest `--->(P)      | |        (P)        |
      |  /HAW/temp_t | |             /HAW/temp_t | |     /HAW/temp_t  |
      +-------------+ +-----------------------+ +-------------------+
       (a) Adv. Name    (b) Replicate Content    (c) Finish publish
```

                            Figure 3: Publish

3.3. Mapping Subscribe to NDN

   In the proposed publish-subscribe system, the _Subscribe_ operation
   is equivalent to an Interest-based request of previously learned
   content names. A device can learn about new content in various ways:

   a. Name Advertisements by the CP via a dedicated prefix path or
       broadcast.

   b. By requesting a named topic from the CP and receiving a specific
       name of available content in return.

   c. By polling a name inventory / a dedicated data structure from the
       CP that records publishings and updates. Such a data structure
       may contain indicators about the periodicity of sensor readings
       to align periodic polling schemes to sensor reading intervals.

   d. By (offline) encoding topics into a hierarchical naming scheme of
       the form _routing prefix / topic / unique_part_.

   Consumers request topics as well as named content by regurlar
   Interest requests. Thereby, Inerest timeouts for these requests
   serve as subscription periods, may be selected of a specific (long)
   timespan, and must be refreshed or retriggered for every publishing
   to adhere to the flow control approach of NDN.

3.4. Content Replication in partitioned networks

   The hop-wise replication described in Section 3.2 transparently
   supports publish operations in partitioned networks. When a publish
   operation fails to replicate content and no backup parent is in the
   vicinity (Figure 4 (b)), the node marks its sub-DAG as _floating_ by
   propagating PAMs with the floating bit set and the NC entry is
   preserved. Once a sub-DAG becomes connected to another parent, the
   publishing is resumed (Figure 4 (c)).

```
         +-------------+     +-------------+     +-------------+
         |             |     |             |     |             |
         |    (CP)     |     |    (CP)     |     |    (CP)<-,   |
         |             |     |             |     |    /    /   |
         |             |     |         X   |     |   /    /    |
         |   (A)<-,    |     |   (A)---'   |     |  (A)---'    |
         |    \   | PUB |    |    \    PUB  |    |   \    PUB   |
         |     \  |    |     |     \       |     |    \        |
         |     (P)    |     |     (P)     |     |    (P)       |
         | /HAW/temp_t |     | /HAW/temp_t |     | /HAW/temp_t |
         +-------------+     +-------------+     +-------------+
          (a) Publish content (b) Publish fails (c) Finish Publish
```
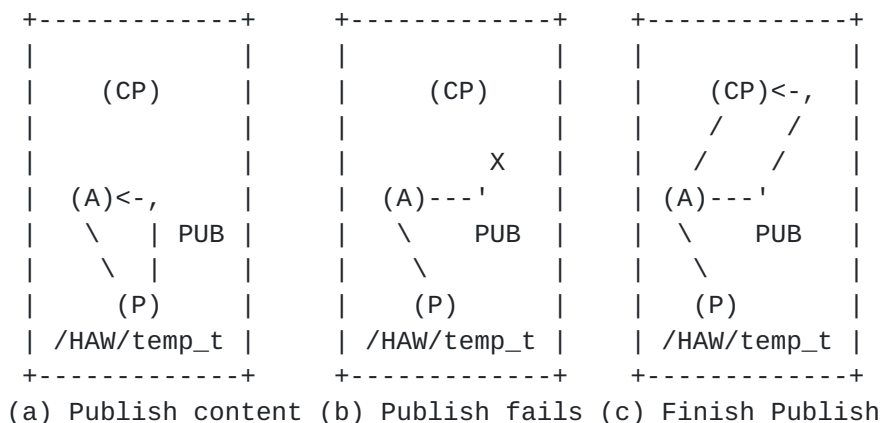
                  Figure 4: Publish in partitioned network

   A node receiving a PAM from its parent with the floating bit set may
   rejoin the tree using another parent in the neighborhood that is
   connected to the content proxy. Rejoining the tree may result in a
   worse rank.

3.5. Content Replication between Proxy Instances

   Content Proxies within a network domain MAY advertise prefixes from
   an overlapping prefix range and thereby implement multiple upstreams
   in (some of) the FIBs. According to Section 3.2, content will be
   published to these multiple CPs upstream and thus populate a
   replicated environment.

   Content Proxies MAY equally establish a subscription hierarchy
   between neighboring domains. This requires an exchange of
   corresponding FIB entries between the involved domain. It is
   noteworthy that a global exchange of FIB tables (e.g., Inter-provider
   peering) is beyond the scope of this document.

3.6. Alerting and group communication

    TODO

4. Control Plane Messaging

    Control plane messaging is currently under separate discussion.
    Details will be added after consensus is found.

    TODO: add mappings of PAM and NAM to the NDN and CCNx dialects

4.1. Prefix Advertisement Message (PAM)

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Type = PAM    |F|   Reserved  |            Rank             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |        Prefix Length          |            Prefix ...
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

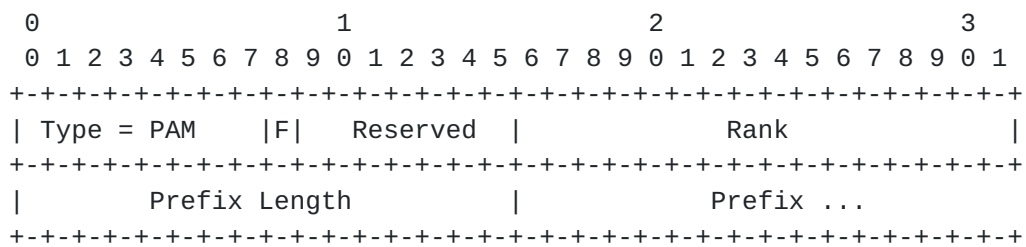   Figure 5: Prefix Advertisement Message (PAM)

    Message type:
              8-bit unsigned integer. Indicates a PAM.

    Floating (F):
              1-bit floating flag. Indicates that a sub-tree is not
              connected to the content proxy, when set.

    Reserved:
              7-bit unused field. It MUST be initialized to zero by
              the sender and MUST be ignored by the receiver.

    Rank:
              16-bit unsigned integer. Indicates a node's position in
              the SPT.

    Prefix Length:
              16-bit unsigned integer. Indicates the length of the
              following prefix in bytes.

    Prefix:
              Variable length prefix used to configure a default route
              within the SPT.

4.2. Name Advertisement Message (NAM)

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = NAM    |   Reserved    |        Options Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                            Options                            :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
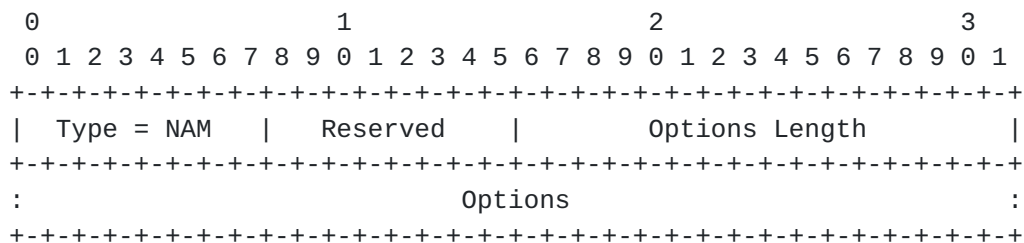
Figure 6: Name Advertisement Message (NAM)

Type:
        8-bit unsigned integer. Indicates a NAM.

Reserved:
        8-bit unused field. It MUST be initialized to zero by
        the sender and MUST be ignored by the receiver.

Length:
        16-bit unsigned integer. Indicates the accumulated
        length of all options.

4.2.1. Name Option

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 0x00   |         Name Length         |   Name ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
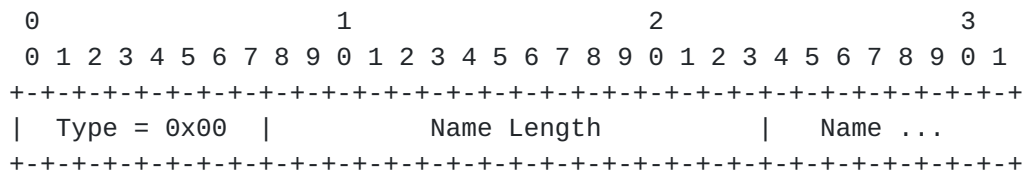
Figure 7: Name option format

Type:
        8-bit unsigned integer. Indicates the name option
        (0x00).

Name Length:
        16-bit unsigned integer. Indicates the length of the
        name, excluding the type and length field.

Name:
        Variable length name.

5. Security Considerations

   TODO

6. References

6.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

6.2. Informative References

   [HoPP]   Cuendogan, C., Kietzmann, P., Schmidt, TC., and M.
              Waehlisch, "HoPP: Robust and Resilient Publish-Subscribe
              for an Information-Centric Internet of Things", Technical
              Report No. arXiv:1801.03890, January 2018.

   [I-D.gundogan-icnrg-ccnlowpan]
              Gundogan, C., Schmidt, T., Waehlisch, M., Scherb, C.,
              Marxer, C., and C. Tschudin, "CCN Packet Adaptation to
              IEEE 802.15.4 Networks", draft-gundogan-icnrg-ccnlowpan-00
              (work in progress), September 2017.

   [IWMT]   Kutscher, D. and S. Farrell, "Towards an Information-
              Centric Internet with more Things", Position Paper,
              Interconnecting Smart Objects with the Internet
              Workshop IAB, 2011.

   [NDN-EXP] Baccelli, E., Mehlis, C., Hahm, O., Schmidt, TC., and M.
              Waehlisch, "Information Centric Networking in the IoT:
              Experiments with NDN in the Wild", Proc. of 1st ACM Conf.
              on Information-Centric Networking (ICN-2014) ACM DL, pp.
              77-86, September 2014.

   [PANINI]   Schmidt, TC., Woelke, S., Berg, N., and M. Waehlisch,
              "Let's Collect Names: How PANINI Limits FIB Tables in Name
              Based Routing", Proc. of 15th IFIP Networking
              Conference IEEE Press, pp. 458-466, Mai 2016.

   [RFC7476] Pentikousis, K., Ed., Ohlman, B., Corujo, D., Boggia, G.,
              Tyson, G., Davies, E., Molinaro, A., and S. Eum,
              "Information-Centric Networking: Baseline Scenarios",
              RFC 7476, DOI 10.17487/RFC7476, March 2015,
              <https://www.rfc-editor.org/info/rfc7476>.

   [RFC7927] Kutscher, D., Ed., Eum, S., Pentikousis, K., Psaras, I.,
              Corujo, D., Saucez, D., Schmidt, T., and M. Waehlisch,
              "Information-Centric Networking (ICN) Research
              Challenges", RFC 7927, DOI 10.17487/RFC7927, July 2016,
              <https://www.rfc-editor.org/info/rfc7927>.

   [RFC7945] Pentikousis, K., Ed., Ohlman, B., Davies, E., Spirou, S.,
              and G. Boggia, "Information-Centric Networking: Evaluation
              and Security Considerations", RFC 7945,
              DOI 10.17487/RFC7945, September 2016,
              <https://www.rfc-editor.org/info/rfc7945>.

Authors' Addresses

   Cenk Gundogan
   HAW Hamburg
   Berliner Tor 7
   Hamburg D-20099
   Germany

   Phone: +4940428758067
   EMail: Cenk.Guendogan@haw-hamburg.de

   Thomas C. Schmidt
   HAW Hamburg
   Berliner Tor 7
   Hamburg D-20099
   Germany

   EMail: t.schmidt@haw-hamburg.de
   URI: http://inet.haw-hamburg.de/members/schmidt

   Matthias Waehlisch
   link-lab & FU Berlin
   Hoenower Str. 35
   Berlin D-10318
   Germany

   EMail: mw@link-lab.net
   URI: http://www.inf.fu-berlin.de/~waehl