ICN Research Group   P.Mendes
Internet-Draft   COPELABS/University Lusofona
Intended Status: Experimental   Rute Sofia
Expires: February 28, 2019   Senception/COPELABS/University Lusofona
                                            Vassilis Tsaoussidis
                                            Sotiris Diamantopoulos
                                          Christos-Alexandros Sarros
                                          Democritus University of Thrace
                                                 August 27, 2018

        Information-centric Routing for Opportunistic Wireless Networks
                        draft-mendes-icnrg-dabber-01

Abstract

   This draft describes the Data reAchaBility BasEd Routing (DABBER)
   protocol, which has been developed to extend the reached of Named
   Data Networking based routing approaches to opportunistic wireless
   networks. By "opportunistic wireless networks" it is meant multi-hop
   wireless networks where finding an end-to-end path between any pair
   of nodes at any moment in time may be a challenge. The goal is to
   assist in better defining opportunities for the transmission of
   Interest packets towards the most suitable data source, based on
   metrics that provide information about: i) the availability of
   different data sources; ii) the availability and centrality of
   neighbor nodes; iii) the time lapse between forwarding Interest
   packets and receiving the corresponding data packets. The document
   presents an architectural overview of DABBER followed by
   specification options related to the dissemination of name-prefix
   information to support the computation of next hops, and the ranking
   of forwarding options based on the best set of neighbors to ensure a
   short time-to-completion.

Status of this Memo

material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 28, 2019.

Copyright and License Notice

Table of Contents

1. Introduction

   In a networking scenario where an increasing number of wireless
   systems, such as end-user nodes and mobile edge nodes, are being
   deployed, there are two networking paradigms that are highly
   correlated to the efficiency of pervasive data sharing: Information-
   Centric Networking (ICN), and opportunistic wireless networking. The
   latter concerns the capability of exploiting any potential wireless
   communication opportunity to exchange data in a multi-hop wireless
   networks, where it is difficult to find an end-to-end path between
   any pair of nodes at any moment in time.

   Combining opportunistic networking with ICN principles is relevant to
   efficiently extend the applicability of information-centric
   networking to novel scenarios, such as affordable pervasive access;
   low cost extension of access networks; edge computing; vehicular
   networks.

   This document describes the Data reAchaBility BasEd Routing (DABBER)
   routing protocol for information-centric wireless opportunistic
   networks [24]. These networking architectures are operationally
   located on the Internet fringes (Customer Premises). In such areas,
   networking experiences intermittent connectivity and variable
   availability of nodes due to their movement and/or due to other
   constrains, e.g., limited battery, storage, and processing.

   DABBER has been therefore designed to be compatible with the routing
   deployed within ICN access networks. Its main purpose is to assist in
   extending the reach of multi-hop transmission to opportunistic
   environments, in a seamless and fully interoperable way.

   It is our understanding that routing in such wireless environments
   needs to be done based on strategies that take into consideration, at
   a network level, the context of wireless nodes, and not just the
   history of contacts among wireless nodes. The goal is to assist in
   better defining opportunities for the transmission of Interest
   packets over time and space. Such opportunities can be better
   addressed if routing metrics take into consideration, as common in
   opportunistic environments, measures of centrality, as well as
   measures of node and data availability.

   Being NDN[1][2] a well established ICN framework, the first step
   proposed by this draft is to consider the current de facto NDN
   routing, Named-data Link State Routing protocol (NLSR)[19][20], in a
   way that allows the benefits of link-state approaches, while
   delimiting its downsize in the context of the wireless medium.

   Although DABBER specification allows an easier integration with NDN

access networks based on NSLR (e.g. the messaging systems based on
the synchronization of LSA is the same), its operation is completely
independent of NSLR. This means that DABBER can be used in any
isolated wireless opportunistic network, or by any wireless node that
frequently attaches to fix NDN networks (e.g. by using a Wi-Fi link),
which do not have NLSR as their routing protocol.

DABBER is intended as complementing existing forwarding protocols for
opportunistic networks (e.g., Prophet [12], Scorp [13], dLife
[14][18], BubbleRap [15]).

1.1. Contextual Aspects

Prior art in forwarding solutions for opportunistic networks showed
that data transmission in such wireless environments needs to be done
based on strategies that take into consideration, at a network level,
the context of wireless nodes, and not just the history of contacts
among wireless nodes.

This section provides an example on how to obtain contextual
information that defines the availability and centrality of a
wireless node, based on a specific operational example that is being
developed in the context of the H2020 UMOBILE project [17].

Contextual information is obtained in a self-learning approach, by
software-based agents running in wireless nodes, and not based on
network wide orchestration. Contextual agents are in charge of
computing node and link related costs concerning availability and
centrality metrics. Contextual agents interact with DABBER via well-
defined interfaces. This to say that the contextual self-learning
process is not an integrating part of the routing plane, as it would
add additional complexity to the simplified routing plane of NDN.

The contextual agent (named Contextual Manager, CM [7]) installed in
each wireless node can therefore be seen as an end-user background
service that seamlessly captures wireless data to characterize the
affinity network (roaming patterns and peers' context over time and
space) and the usage habits and data interests (internal node
information) of a node. Data is captured directly via the regular MAC
Layer (e.g., Wi-Fi, Bluetooth, LTE) as well as via native
applications for which the user configures interests or other type of
personal preferences. For instance, an application can request a one-
time configuration of categories of data interests (e.g., music,
food).

Based on the defined interface (cf. section 3.6), DABBER is able of
querying the local Contextual Manager about the characteristics of
neighbor nodes, based on three types of information: i) Node

availability (metric A); ii) Node centrality (metric C); iii) Node
similarity (metric S).

Node Availability (A) gives an estimate of the node availability
based on the usage of internal resources over time and space, such as
the time spent per application category (e.g. per day), as well as
the usage of physical resources (battery status; CPU status, etc).

Node Centrality (C) provides awareness about the node's affinity
network neighborhood context. For instance, the more visited networks
a node has over a period of time, the more central a node is
(increases the possibility for data transmission); The higher the
number of connections a node has over a period of time, the more
central a node is; The higher the node degree of node over a period
of time, the higher is its centrality; The lower the distances
traversed by the node are, the higher is its centrality.

Node similarity (S) provides awareness about a node's similarity
towards neighbor nodes, based on the assumption that the more similar
nodes are, the higher the probability of more robust data exchange
between those nodes. This metric relies on a cosine similarity to
provide a link cost between peer nodes. In the case of DABBER,
similarity is computed based on the number of encounters between peer
nodes and the average duration of such encounters.

The detailed specification of the contextual manager is out of scope
of this document. Nevertheless, code for such an agent is being
provided openly in the context of the H2020 UMOBILE project [7]. What
is relevant to have in mind, from a routing perspective, is that this
contextual plane provides weights (A, C and S) to assist the routing
protocol in ranking next hops, which is an aspect highly relevant in
the context of multiple path routing. We believe that contextual
awareness can assist NDN routing schemes in better dealing with
topological variability, by anticipating changes derived from prior
learning.

1.2. Applicability

DABBER is being developed to allow the deployment of wireless NDN
networks where nodes and links can be intermittently available, such
as in the case of emergency situations [25]. From an end-to-end
perspective we can consider two scenarios: the NDN wireless network
is at the fringes of the NDN core; the NDN wireless network can
interconnect different NDN fixed networks.

While the latter may support applicability scenarios typical of
Delay-Tolerant Networks (DTN)[21][22] for instance tunneling traffic
over an area lacking network deployment, the former allows the

extension of the applicability of information-centric networking to
novel scenarios such as affordable pervasive data access, low cost
extension of access networks, edge computing, and vehicular networks:

Affordable pervasive data access:
   This scenario encompasses the implementation of NDN in personal
   mobile nodes (e.g. smartphones) allowing users to share data and
   messaging services by exploiting existing intermittent wireless
   connections (e.g. Wi-Fi, Wi-Fi direct) in environment without/or
   limited Internet access.

Low cost extension of access networks:
   This scenario refers to the usage of wireless nodes (mobile or
   fix) to extend the reach of an NDN networks while reducing CAPEX
   costs.

Edge/Fog computing:
   This scenario is related to the efforts being done to bring cloud
   computing closer to the end-users. This scenario encompasses a
   large set of heterogeneous (wireless and sometimes autonomous)
   decentralized nodes able of communicating, directly or via an
   infrastructure, in order to perform storage and processing tasks
   without the intervention of third parties. This scenario deals
   with nodes that might not be continuously connected to a network,
   such as laptops, smartphones, tablets and sensors, as well as
   nodes that may be intermittently available due to scarce
   resources, such as wireless access routers and even Mobile Edge
   Computing (MEC) servers.

V2X networks:
   This scenario deals with the intermittent connectivity between
   vehicles as well as between vehicles and the infrastructure.

1.3. NFD Adjustment to Opportunistic Networks

   The main functionality of the Named-Data Networking Forwarding Daemon
   (NFD) [7] is to forward Interest and Data packets. This section
   provides a set of design considerations that need to be considered to
   allow the operation of NFD in opportunistic wireless networks. Such
   considerations have been implemented in a new branch of NDN, called
   NDN-OPP [3], which code of available on GitHub
   (https://github.com/COPELABS-SITI/ndn-opp).

   NDN-OPP introduces a few modifications in the way NFD performs its
   forwarding, by leveraging the concept of Faces in order to adapt the
   operation of the NFD to the intermittent property of wireless
   connections. This is done by the implementation of a new type of
   face, called Opportunistic Face - OPPFace.

Each OPPFace is based on a system of packet queues to hide
intermittent connectivity from NFD: instead of dispatching packets
from the FIB, the OPPFace is able of delaying packet transmission
until the wireless face is actually connected. OPPFaces are kept in
the Face Table of the forwarder and their state reflects the wireless
connectivity status: they can be in an Up or Down state, depending
upon the wireless reachability towards neighbor nodes. Since packet
queuing is concealed inside OPPFaces, no other part of the NFD or any
existing forwarding strategy needs to be changed.

OPPFaces can be implemented by using any direct wireless/cellular
communication mode (e.g., Ad-Hoc Wi-Fi, Wi-Fi Direct, D2D LTE, DTN).

The current operational version of NDN-OPP (V1.0) makes usage of
group communications provided by Wi-Fi Direct. In this case there is
a one-to-one correspondence between an OPPFace and a neighbor node.
In this peer-to-peer scenario, OPPFaces can be used in two
transmission modes: connection-oriented, in which packets are sent to
a neighbor node via a reliable TCP connection over the group owner;
connection-less, in which packets are sent directly to a neighbor
node during the Wi-Fi direct service discovery phase. In the latter
case data transmission is limited to the size of the TXT record (900
bytes for Android 5.1 and above).

In the peer-to-peer scenario of Wi-Fi direct, DABBER operates as
follows: routing information is shared among all members of a Wi-Fi
direct group, while Interest Packets are forwarded to specific
neighbors. With Dabber it is the carrier of an Interest packet that
decides which of the neighbors will get a copy of the Interest
packet. Hence, with the current implementation of NDN-OPP, DABBER
places a copy of the Interest packet in the OPPFaces of selected
neighbors. In what concerns the dissemination of routing information,
it is ensured by: i) node mobility, meaning that nodes carry such
information between Wi-Fi direct groups; ii) information is passed
between neighbor groups via nodes that belong to more than one group.

In a scenario where NDN-OPP would have OPPFaces implemented based on
a broadcast link layer, such as ad-hoc Wi-Fi, only one OPPFace would
be created in each node. Such OPPFace would be used to exchange
packets with any neighbor node, making use of the overhearing
property of the wireless medium. Since with DABBER, it is the carrier
that decides which of the neighbors are entitle to get a certain
Interest packet, DABBER would need to encode in the Interest packet
information about the ID of the neighbors that should process the
overheard Interest packet.

This means that the operation of DABBER is the same independently of
the nature of the link layer protocol, the only different being the

number of transmissions that needs to be done at the link layer to
forward Interest packets and to disseminate routing information.

Besides the OPPFaces towards neighbor wireless nodes, NDN-OPP makes
use of the Wi-Fi Face, already defined in NFD, and will integrate the
DTN Face developed in the UMOBILE project[23]. This means that DABBER
is able of exploiting any available wireless Face (OPPFace, Wi-Fi
Face, DTN Face). Future versions of NDN-OPP will allow DAGGER to
exploit interfaces to other wireless access networks, such as LTE.

A detailed specification of NDN-OPP and OPPFaces can be found in [3].
In the remainder document we will refer to OPPFaces, Wi-Fi Faces and
DTN Faces simply as Faces.

1.4. Conventions

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. In this
document, these words will appear with that interpretation   only
when in ALL CAPS. Lower case uses of these words are not to be
interpreted as carrying significance described in RFC 2119.

2. DABBER Architecture

This section presents an overview of the overall DABBER protocol
architecture. The three major considerations to architect DABBER are:

   i) In opportunistic networks it is not possible to know the
   complete network topology. Hence, there is no need to disseminate
   Adjacency information.

   ii) In opportunistic networks it is not efficient to flood the
   network, as shown by all prior solutions based on controlled
   packet replication forwarding ([12][13][14][18][15]) instead of
   broadcast as used in Epidemic routing.

   iii) Selecting the best set of neighbors to replicate packets to,
   may not be efficient if based only on connectivity based
   information (e.g. inter-contact times, contact duration).

DABBER relies on the same message formats, message exchange process,
and same data structures (RIB and FIB), made available by NDN, and
used by NLSR. As shown in figure 1, both protocols are able of
populating the FIB with a list of next hops towards each name prefix.
This is done based on the information collected from neighbor nodes
and stored in the RIB.

```
              Node A                        Node B
         +----------+                  +-----------+
   N -   |1        2|  - N----- |1        2|
         |          |           |           |
         |3        4|  - N      |3        4|
         +----------+    |      +-----------+
                         |           Node C
                         |      +-----------+
              ------ |1        2|
                         |           |
                         |3        4|
                         +-----------+
```

```
                 RIB                                    FIB
   +---------------------------+          +----------------------+
   |Prefix Name | Face   | Cost |          | Prefix Name |  Faces   |
   +---------------------------+          +----------------------+
   |     N      |   2    |  3   |          | N           |  2,1,4   |
   |     N      |   4    |  10  |          |             |          |
   |     N      |   1    |  5   |          +---------------------- +
   +---------------------------+
```
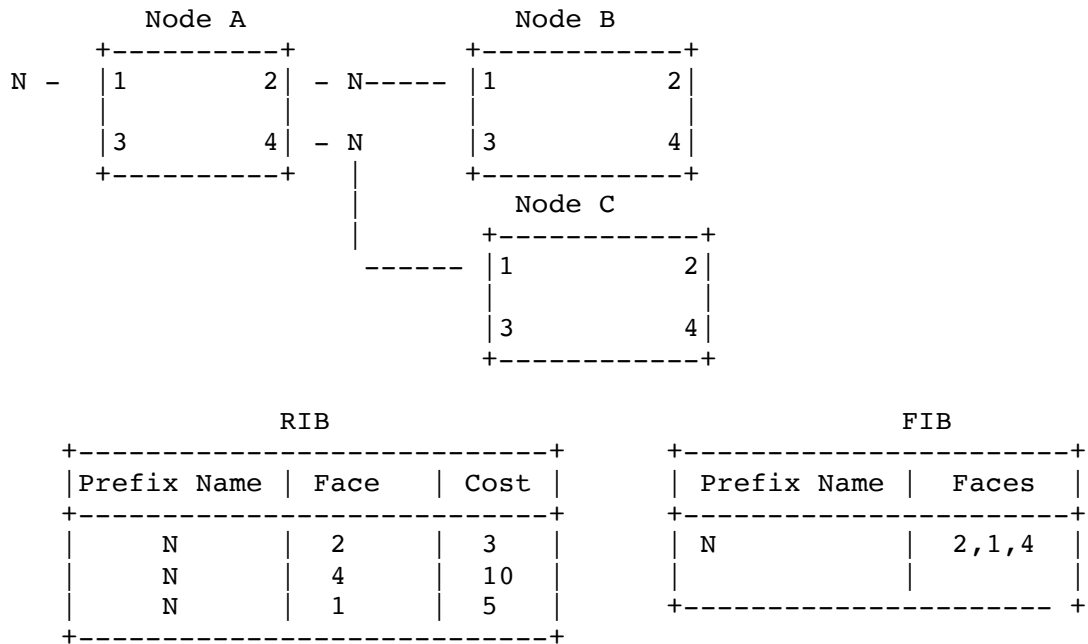
Figure 1: RIB and FIB Management, node A.

However, NLSR needs to build a full network topology, based on
Adjacency Link State Advertisements (LSA), to compute shortest paths
towards each node in the network (based on a simple extension of the
Dijkstra's algorithm). After this, NLSR computes shortest paths
towards each data source by associating each router with name
prefixes, based on the information exchanged via Prefix LSAs. Such
name prefixes are ordered in the FIB based on the distance of the
path towards the data source (shortest first).

While NLSR relies on the dissemination of Adjacency and Prefixes
LSAs, DABBER only requires the dissemination of Prefix LSAs and does
not require the computation of shortest paths: DABBER replaces the
path cost used by NSLR with a data reachability cost, as described in
section 2.4, reducing the impact that topological changes would have
on the stability of routing information.

The computation of data reachability costs towards different data
sources, based on the local dissemination of name prefixes, aims to
avoid flooding the wireless network with Interest packets that would
otherwise be broadcast to all potential data sources.

2.1. Assumptions and Requirements

DABBER relies on the following assumptions:

   o Mobile nodes are able of exploiting wireless connectivity. For
   instance having NDN-OPP installed.

   o Mobile nodes can be a source and destination of data, being able
   of operating as a router: there is not a clear distinction, in
   terms of routing process, between sources, destinations, and
   routers.

In terms of requirements:

   o LSAs must be exchanged based on Interest / Data messages, as in
   NSLR.

   o A synchronization mechanism should be used to exchange LSAs
   among neighbor node, as in NSLR.

   o LSAs should be used to distribute only name prefix reachability,
   since building a network topology based on adjacency information
   is not feasible in an opportunistic network.

   o Multiple next-hops for each name prefix must be computed based
   on local information that encodes data reachability.

   o Link failure recovery must be local and hence, the recovery
   process should be based on the operation of OPPFaces (UP/Down link
   management).

   o IP addresses or any other form of addressing a node in the
   network must not be considered, as in NLSR.

   o Selective information diffusion must be considered, in order to
   avoid network flooding.

   o Data sources must set the validity of name prefixes – validity v
   – as an integer that represents the expiration date of the data.

2.2. Naming

   DABBER makes use of NDN hierarchical naming scheme to identify each
   wireless node. This strategy is similar to the one used by NLSR. The
   difference is in the name semantics: being a routing protocol for
   wired networks, NLSR uses names that reflect network structures and
   operational practices, making it easy to identify routers belonging
   to the same network, and operator realms. In NLSR each router is
   named according to the network it resides in, the specific site it

belongs to, as well as an assigned router name, i.e.,
/<network>/<site>/<router>. For example, /ATT/AtlantaPoP1/router3.
This semantics provide additional topological information to the
routing process.

With DABBER, we assume that DABBER is installed in mobile devices
(e.g. smartphones), which have a contract with a mobile operator. In
a networking environment, a hierarchical naming scheme still makes
sense to identify to which network operator does the mobile node
belongs to and to the home site, in case the mobile operator has more
than one operational site. This naming scheme still makes sense even
if DABBER is only used to exchange traffic over 802.11, wifi direct
or 802.11 adhoc links, and never over a cellular interface.

Since DABBER is used to exchange data directly between mobile nodes
in an opportunistic networking scenario, it makes use of a
hierarchical naming scheme that reflects the way mobile roaming
works: When a mobile node is used outside its home, it attempts to
communicate with a visited mobile network. The visited network
recognizes that the node does not belong to any of its networks, and
checks if there is a roaming agreement between the home network and
one of the networks of the visited operator. If so the call is routed
towards an international transit network.

Based on the operation of a mobile network, the following semantics
is used to name DABBER nodes:/<network>/<operator>/<home>/<node>,
where <network> represents the international transit network allowing
roaming services for the mobile operator; <operator> refers to the
operator providing the mobile service; <home> is the network site of
the mobile operator where the node is registered; <node> is the
mobile equipment.

The hierarchical name is used to implement a trust model to allow
nodes to verify the signature of routing messages, as described in
section 5.

The information included in the hierarchical name may be used to
select next hops belonging to the same operator network, or nodes
that have the same home network. It is assumed that an opportunistic
wireless network is build based on wireless direct connectivity
between nodes that may belong to different operators and home
networks, but that may have roaming patterns that allows them to have
frequent wireless contacts.

2.3. LSA Dissemination

   DABBER runs on top of NDN, making use of Interest/Data packets to

exchange LSAs. This means that while IP-based routing protocols push
updates to other routers, DABBER nodes need to pull the updates.
DABBER can use any underlay communication channels (e.g., TCP/UDP
tunnels, Link layer TXT records) to exchange LSA information.

DABBER benefits from NDN built-in data authenticity: since a routing
update is carried in an NDN data packet and every NDN data packet
carries a signature, a DABBER node can verify the signature of each
LSA message to ensure that it was generated by the claimed origin
node and was not tampered during dissemination.

Similarly to what happens with NLSR, DABBER disseminates LSAs via a
data synchronization mechanism (e.g. ChronoSync [9], PartialSync
[10]) of the local LSDB.

The main differences towards NLSR are:

   o Contrary to NLSR, DABBER does not disseminate Adjacency LSAs to
   reflect the status of the links towards neighbor nodes.

   o As NSLR, DABBER advertises Prefix LSAs every time a new name
   prefix is added or deleted to the LSDB. However in the case of
   DABBER, name prefixes are advertised with a cost/metric related to
   the validity of the associated data.

This peer synchronization approach is receiver-driven, meaning that a
node will request LSAs only when it has CPU cycles. Thus it is less
likely a node will be overwhelmed by a flurry of updates. In order to
remove obsolete LSAs, every node periodically refreshes each of its
own LSAs by generating a newer version. Every LSA has a lifetime
associated with it and will be removed from the LSDB when the
lifetime expires. The LSA format is shown in Figure 2.

```
     Prefix LSA
  +----------------------------------------------------------------+
  |  LSA  | Number of |Prefix 1|Cost| ... |Prefix N|Cost| Signature |
  | Name  | Prefixes  |        |    |     |        |    |           |
  +----------------------------------------------------------------+
```
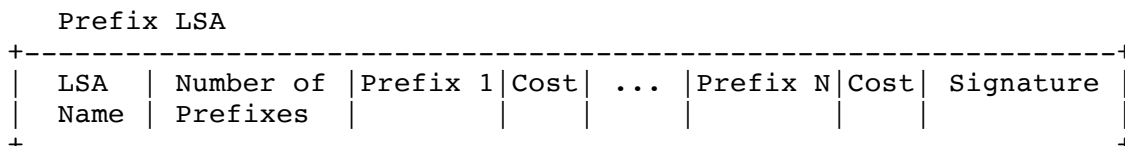
Figure 2: Prefix LSA format.

Each LSA used by DABBER has the name
<network>/<operator>/<home>/<node>/DABBER/LSA/Prefix/<version>. The
LSA <version> is increased by 1 whenever a node creates a new version
of the LSA.

A detailed description of the LSA exchange process is provided in
section 3.3.

2.4. Multiple path Computation

   As mentioned, DABBER considers that there is a set of potential next-
   hops via which a name prefix N can be reached with a certain cost k.
   This cost k represents the probability of reaching a data object
   identified by N via a Face F, and is related to the time validity of
   the name prefix (v). The rationale for this approach is that the
   selection of faces that have a lower cost k (higher validity v) will
   improve data reachability. The validity of a name prefix is set by
   the data source as an integer that represents the expiration date of
   the data.

   Since different nodes can announce the same name prefix, a certain
   name prefix may be associated with different values of k (as v shall
   differ) over different faces, depending upon the nodes announcing
   such name prefix: this lead to the identification of multiple next
   hops, each one with a different cost.

   The computation of multiple next hops is performed every time DABBER
   has a new Name Prefix LSA (or a new version of an existing Name
   Prefix LSA) in its LSDB (c.f. section 2.3). The sequence of
   operations, as described in the following sub-sections are: Computes
   a new value for the validity of a new Name Prefix in the LSDB;
   Updates DABBER internal routing table; Updates the LSDB with the data
   reachability information (validity) of the current node towards the
   new Name Prefix; Updates the RIB based on the DABBER internal routing
   table, following a Downwards Path Criterion (FIB is updated by NFD
   based on the RIB content). Periodically DABBER will update the
   validity values of all Name Prefixes in its internal routing table,
   performing the consequent updates of the local LSDB and RIB, and
   needed.

2.4.1. Name Prefix Validity Computation

   When DABBER is notified that a new Prefix LSA was entered in the LSDB
   or an existing Prefix LSA has a new version, it computes a new
   validity for each name prefix in such Prefix LSA.

   DABBER starts by computing a new validity v for a prefix N depending
   upon the validity announced by the neighbor, and the relevancy of the
   "relation" between the two neighbor nodes (e.g., node A and node B).
   The cost of the Name Prefix, passed to NFD, will then be computed as
   an inversely proportional value to its validity.

   The relevancy of the "relation" between two neighbor nodes can be,
   e.g., a measure of similarity [7], where similarity is seen as a link
   measure, i.e., it provides a correlation cost between a node and its
   neighbors. Or such relation can be weighted based, as is common in

opportunistic environments, on metrics derived from average contact
duration thus allowing a node to adjust the Name Prefix validity
based on the probability of meeting the respective neighbor again. In
the current implementation of DABBER, the "relation" between two
neighbor nodes is computed based on the three metrics (A, C, and S)
provided by the local contextual manager, plus a metric computed by
DABBER itself: the time reachability.

The variable considered by DABBER for the computation of the validity
(and cost) of a Name prefix passed by a neighbor Na are:

   o Validity (V) - Represents the expiration date of the data
   associated with the Name Prefix. Currently it is the UNIX
   Timestamp (10 digit integer).

   o Similarity metric (S) towards the neighbor Na, as passed by the
   contextual manager (S >= 0), aiming to select neighbors with whom
   the current node has a good communication link.

   o Availability metric (A) towards the neighbor Na, as passed by
   the contextual manager (0 < A < 1), aiming to select neighbors
   able to process Interest packets with high probability.

   o Centrality metric (C) towards the neighbor Na, as passed by the
   contextual manager (C >= 0), aiming to select neighbors with high
   probability of successfully forwarding Interest packets.

   o Time reachability (T) which corresponds to the RTT between
   sending an Interest packet towards the source of such Name Prefix
   and receiving a data packet. (0 < T < 1). Currently the value of T
   is computed as (current time when data packet of received - time
   when Interest packet was sent) / Validity of Name Prefix. Time
   reachability allows DABBER to select next hops that lead to closer
   sources.

        Neighbor table
     +--+
     | Face | Status | Metric C | Metric A | Metric S |
     +--+
     | 1 | UP | 6 | 0.3 | 12 |
     | 2 | DOWN | 4 | 0.8 | 8 |
     | 3 | UP | 1 | 0.8 | 9 |
     +--+

Figure 3: Neighbor table.

The values C, A and S provided by the contextual manager are stored

in a Neighbor Table as shown in figure 3. Since an OPPFace is created
by NDN-OPP (c.f. section 1.3) by NDN-OPP, the table is indexed by the
number of faces. The higher the values of C, A and S, the most
preferential a neighbor is.

T is measured by observing the flow of Interest and Data packets.
Thus, the lowest the T, the most preferential a Face is. Although
different nodes may have a different implementation of a face ranking
logic, the relevancy of T in comparison to C and A should be higher,
since T reflects the measured delay to reach a data source, while C
and A are indicators of the neighbors potential as relays.

Based on the above mentioned metrics the Validity of a new Name
Prefix (V) is updated based on two operations:
   o V' = f (V, S') = trunc (V * S'), where:

      S' = (S - Smin) / (Smax - Smin); Smin = 0 and Smax = max (Smax,
      C)

   o V'' = f (V', C', A, T) = 0.3* trunc (V' * (C'+A)) + 0.7 * trunc
   (V' * T), where:

       C' = (C - Cmin) / (Cmax - Cmin); Where Cmin = 0 and Cmax = max
      (Cmax, C)

2.4.2. Update of DABBER internal information

   After the computation of the validity of the Name Prefix taking into
   account the relation of the current node with the neighbor announcing
   it (c.f. section 2.4.1), DABBER updates its internal routing table
   and its LSDB. The information on the routing table will be used to
   updated the RIB of the local NFD and the information of the LSDB will
   be announced to all the neighbor by Chronosync.

   To update the Internal routing table, DABBER adds an entry Na for the
   Name Prefix with a validity V''. The routing table is then ordered by
   name prefix in decreased order of validity.

   To update its local LSDB updated with validity of current node
   towards the Name Prefix, DABBER can use two methods:
      o Maximal method: Name Prefix entry on current node LSA updated
      with max (V'', current value on LSA).

      o Average method: Name Prefix entry on current node LSA updated
      with max (average (cost of Name Prefix entries on local routing

table), current value on LSA).

2.4.2. Update of RIB of the local NFD

   After computing the new value of the validity V'' of a name prefix,
   as described in section 2.4.1, DABBER updates the RIB entry of that
   name prefix with the face over which the Name Prefix LSA was received
   and the new cost of such Name prefix. The cost (k) of the Name Prefix
   is computed based on its validity as k = trunc (100/V'').

   DABBER assigns selection logics to name prefix, such as NDN assigns
   forwarding strategies to name prefixes.

   There may be different available logics to choose from:

      o Increase diversity - The new Face is included in the RIB entry,
      if the computed cost k helps to increase diversity of the name
      prefix. For instance the new cost k is higher than the average
      costs already stored for that name prefix, affected by a
      configured diversity constant. This is include all neighbors with
      cost = trunc (100/V''), such that 1/V'' - Avr (Costs in RIB for N)
      > X (predefined value).

      o Downward Path Criterion - It is a non-equal cost multi-path
      logic that is guaranteed to be loop-free. Based on the Downward
      Path Criterion, the X faces (the maximum number X of desirable
      faces can be defined by configuration) to be considered for a name
      prefix include the one with the lowest cost k plus X-1 faces that
      have a cost k lower than the cost that the current node has itself
      to the name prefix. This is include X neighbors with cost =
      trunc(100/V''), such that cost is the lowest value of 1/V'' or
      cost < 1/ V.

      o Downward Path Criterion extension - Also considers any face over
      which the name prefix can be reached with a cost k equal to the
      cost that the current node has itself to the name prefix. To avoid
      packet from looping back, there is the need to add a tiebreaker,
      which assures that traffic only crosses one direction of equal-
      cost links. This is, include X neighbors with cost = trunc
      (100/V''), such that cost is the lowest value of 1/V'' or cost
      <=1/ V.

   The FIB is then updated by the NFD from RIB (multicast forwarding
   strategy used).

2.5. Loop Prevention

Given the multi-path nature of DABBER, the incoming Face might appear among the potential next-hops for a given name prefix. For this reason, DABBER applies the Incoming Face Exclusion principle [11] in order to prevent forwarding Interest packets back though the Face them came from, thus removing two-hop loops.

Furthermore, in order to detect longer forwarding loops (more than two hops), DABBER relies on the nonce-based detection scheme available in NDN in order to drop a looping packet as soon as it is received the second time.

In addition, DABBER considers a loop removal mechanism, which takes care of disabling the Face responsible for the looping once it is detected, as described in section 3.4.

3. Protocol Overview

3.1. Overall Operation Example

We consider the scenario in Figure 4 to assist in the protocol operation overview: namely to understand to role of DABBER to allow extension of NDN operation towards wireless dynamic networks. In Figure 4, nodes A, B, and C reside in an opportunistic network and run DABBER, while nodes E and F are wireless edge routers running another NDN routing/forwarding protocol, such as NLSR. G is a wireless node running DABBER.

```
        +-------------------+
        |    +---+          |
        |    | B |  .       |
        |    +---+  .2+---+  |    +---+     +---+      +---+
        |+---+       | C |3 ... | E |....| F |....| G |
        || A |......1+---+  |    +---+     +---+      +---+
        |+---+              |
        +-------------------+
```
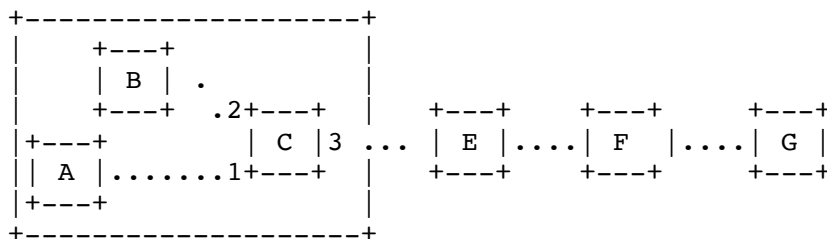
Figure 4: End-to-end operational example.

In our example, Node A starts producing some content derived, for instance, from the use of an application (such as a data sharing application). The produced content is stored in its local Content Store with the name /NDN/video/Lisbon/nighview.mpg. Node B stored in its Content Store a data object with name /NDN/video/Lisbon/river.mpg, which node B received from a neighbor (meaning that B have already synchronize its LSDB with such a neighbor).

Due to the update of the Content Store, the name prefix

/NDN/video/Lisbon/ is stored in the LSDB of node A and B with a
validity of 864000 and 518400 in the case of node A and B
respectively. In the case of node A, the cost k of the name prefix
equals the validity v of the data object, e.g., v=864000 seconds (10
days) stipulated by the application. In the case of node B the
validity is the result of the computation process described in
section 2.4.

From a routing perspective, storing a name prefix in the local LSDB
allows the node to share the respective Prefix LSA on all its Faces,
excepting on the Face over which the LSA was previously received, as
explained in section 3.3. This LSA exchange is done when the OPPFaces
are up, as described in section 3.2. Node C, which got a new Prefix
LSA from nodes A and B, will:

   o Updates its LSDB with the Prefix LSAs received from node A and
   node B.

   o Updates its internal routing table with two new entries for the
   name prefix /NDN/video/Lisbon/, associated with the face towards A
   (face1) and with the face towards B (face2), after computing the
   values of V' and V'' for the received validity values:

      o The validity of the name prefix is updated based on the
      metric configured for node C: average inter-contact time.

      o Let's assume that A and C encounter each other frequently,
      while B and C do not meet frequently. This means that node C
      stores 2 new entries for prefix /NDN/video/Lisbon/ in its
      internal routing table related to face2 with a validity for A
      higher than the one for B.

   o Updates its LSDB with the validity of the current node towards
   the Name Prefix following the maximal or average methods (c.f.
   section 2.4.2).

   o Update the RIB with one new entry for the name prefix
   /NDN/video/Lisbon/ with two faces (face 1 and face 2) with a cost
   inversely proportional to the validity of the Name Prefix.

Based on the status of the FIB (updated based on the status of the
RIB) all interest packets that node C gets for the name prefix
/NDN/video/Lisbon/ will be forwards to the number of faces associated
to the used forwarded strategy, but respecting the ranking of faces
done by DABBER.

When node C gets in the range of router E (wireless edge router) it
will exchange disseminate routing information, based on some

interoperability issues need to be considered, as described in
section 4.

3.2. Peer Discovery and Face Setup

In an opportunistic network DABBER needs to manage the dynamic
connectivity among neighbor nodes. For this proposes the DABBER
protocol relies on a background process, the Connectivity Manager.

The current version of DABBER comes with a Connectivity Manager for
Wi-Fi Direct. However, such connectivity manager can be easily
extended to integrate other type of wireless or cellular support. The
description here provided is adjusted to the case of Wi-Fi Direct.

When booted, the Connectivity Manager starts reacting to changes in
the peers available within scanning range of the current node. It
oversees managing the connection to a Wi-Fi Direct Group and
automatically joins a Group if it is not part of one.

Upon the reception of notifications regarding changes in the peers
detected in the neighborhood, the Connectivity Manager updates its
internal peer list. If it is not currently connected to a Wi-Fi
Direct Group, it performs a selection heuristic to determine which
node to connect to. The motivation behind this selection process is
to attempt to minimize the number of Wi-Fi Direct Groups in a certain
area given that nodes can only transmit packets within the Group they
are currently connected to.

The heuristic simply favors whichever Group Owner is already detected
among the available peers. In the case there is exactly one Group
Owner, the current node attempts to join its Group. If more than one
or no Group Owners are available, the heuristic selects the non-
client node with the highest UUID. If the selected node is not the
current node, a connection is attempted. This heuristic guarantees
that the current node will never attempt to connect to a Client, thus
breaking an existing Group. Also, all nodes located in an area and
have the same view of available peers will all select the same node
as the Group Owner to which connection should be attempted.

For each node detected in a Wi-Fi Direct Group, a new instance of an
OPPFace is created. The status of an OPPFace tells us if the
connectivity link towards a specific node is up or down. Based on
this information, the OPPFace decides whether to simply queue a
packet (when OPPFace is down) or flush the queue (when OPPFace is
up).

In order to achieve this, whenever the node joins a Wi-Fi Direct
Group, it gets registered in the Group so that other nodes can send

packets to it. After this setup, all service changes detected within
the Group (connectivity up or down) are reflected into the Neighbor
Table (c.f. Figure 3). Upon disconnection from the Group, the node is
unregistered and the node returns to a state of waiting for a Group
to be joined.

3.3. LSA Exchange

DABBER performs the dissemination of LSAs based on a process able of
synchronizing the content of LSDBs. In this sense, all LSAs are kept
in the LSDB as a name set, and DABBER uses a hash of the LSA name set
as a compact expression of the set. Neighbor nodes use the hashes of
their LSA name sets to detect inconsistencies in their sets. For this
reason, neighbor nodes exchange hashes of the LSDB as soon as
OPPFaces are UP.

Current version of DABBER makes use of ChronoSync as synchronization
mechanism. Chronosync allows DABBER to define a collection of named
data in a local repo as a slice. LSA information are synchronized
among neighbor nodes, since Chronosync keeps the repo slice
containing the LSA information in sync with identically defined
slices in neighboring repositories.

If a new LSA name is detected in a repo, ChronoSync notifies DABBER
to retrieve the corresponding LSA in order to update the local LSDB.
DABBER can also request new LSAs from Chronosync when resources (e.g.
CPU cycles) are available.

Figure 5 shows how an LSA is disseminated between two neighbor nodes
A and B, when the OPPFace is UP. To synchronize the slice
representing the LSDB information in the repo, ChronoSync, on each
node, periodically sends Sync Interests with the hash of its LSA name
set / slice (step 1). When Node A has a new Prefix LSA in its LSDB,
DABBER writes it in the Chronosync slice (step 2). At this moment,
the hash value of the LSA slide of node A becomes different from that
of node B. As a consequence, the Chronosync in node A replies to the
Sync Interest of node B with a Sync Reply with the new hash value of
its local LSA slice (step 3). The Chronosync in node B identifies the
LSA that needs to be synchronized and notifies DABBER about the
missing LSA, and updates its LSA name set (step 4). Since DABBER on
node B has been notified of the missing LSA, DABBER sends an LSA
Interest message to retrieve the missing LSA (step 5). DABBER on node
A sends the missing data in a LSA Data message (step 6). When DABBER
on node B receives the LSA data, it inserts the LSA into its LSDB.
Chronosync on nodes A and B compute a new hash for updated the set
and send a new Sync Interest with the new hash (step 7).

          Node A   Node B

```
     +---------------------------+      +----------------------------+
     |           +-------------+ |      |+-------------+             |
     |  DABBER   | Chronosync  | |      || Chronosync  |   DABBER    |
     |           +-------------+ |      |+-------------+             |
     +---------------------------+      +----------------------------+
          |                 |    Sync Interest (1) |               |
          |                 |------------------->|                 |
          |                 |<-------------------|                 |
          |  New LSA (2)    |                    |                 |
          |--------->       |                    |                 |
          |                 |    Sync Reply (3)  |                 |
          |                 |------------------->|                 |
          |                 |                    |   Notify (4)    |
          |                 |                    |------------->   |
          |                 |    LSA Interest (5)|                 |
          |<-----------|----|------------------->|-------------|   |
          |                 |    LSA Data (6)    |                 |
          |------------|----|------------------->|------------->   |
          |                 |                    |                 |
          |                 |  Sync Interest (7) |                 |
          |                 |------------------->|                 |
          |                 |<-------------------|                 |
```
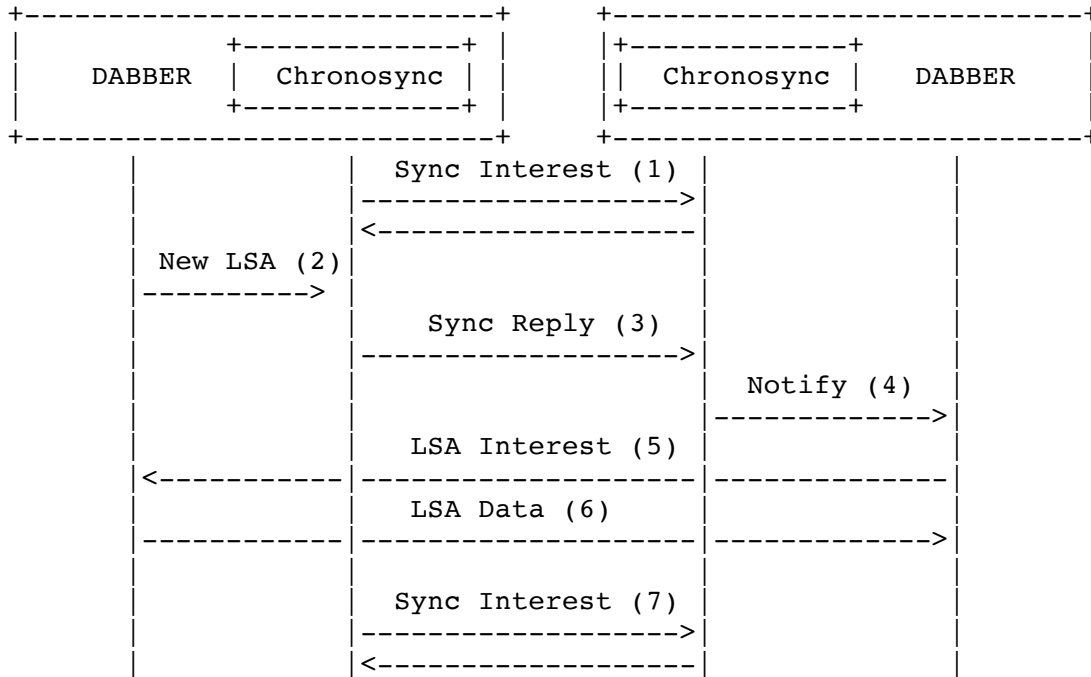
   Figure 5: LSA exchange process.

   When more than one LSA needs to be synchronized, the issued LSA
   Interest packet will contain information about as many LSAs as
   allowed by the Link maximum transmission unit. In the same sense one
   LSA Data packet may include also be used to transport information
   about more than one LSA.

3.4. Loop Avoidance

   In addition to the loop avoidance mechanism of NDN, DABBER considers
   a loop removal mechanism, which takes care of disabling the Face
   responsible for the looping once it is detected.

3.5. Failure and Recovery

   As described in section 3.2, DABBER relies on a connectivity manager
   that is able to react to changes in the peers available within the
   wireless scanning range of the current node.

   Upon detection of a Wi-Fi Direct Group, the connectivity manager
   automatically joins that Group, if it is not part of one.

   Upon the reception of notifications regarding changes in the peers
   detected in the neighborhood, the Connectivity Manager updates its

internal peer list.

3.6. Interface towards a Contextual Agent

The interface between DABBER and CM provides the former with periodic
information concerning a node's centrality (C) and a node's
availability (A), as well as with a similarity weight (S) between
peers (link relevancy).

This interface integrates premises to perform specific requests to
get the computed values C and A for a list of peers provided by
DABBER. The peers are identified by hashed MACs.

The interface integrates also a premise to provide a similarity
weight (S) between two peers passed by DABBER to the CM. For
instance, if DABBER requests similarity between node A (sender) and
node B (potential successor), then the CM computes similarity for
both nodes based on a specific period of time. Such analysis can
assist in a better selection of peers for data transmission, for
instance.

4. Interoperability

As mentioned in section 1.2 DABBER is being developed to allow the
deployment of wireless NDN networks where nodes and links can be
intermittently available. In this section we analyze the
interoperability of DABBER in two scenarios: the NDN wireless network
is at the fringes of a wired NDN core; the NDN wireless network can
interconnect topologically separated NDN networks or hosts, via a
DTN.

4.1. Interoperability with NDN operation over DTNs

In this sub-section, we review the deployment of DABBER over existing
DTNs. We only consider deployment scenarios where NDN is deployed as
an overlay over a DTN. In this case, the existing DTN infrastructure
and implementation are leveraged to extend NDN operation in
challenged networks. We consider scenarios such as data mulling,
services to remote locations, and interconnecting different NDN hosts
(fixed or mobile)[23].

In such challenged network topologies, OPPFaces may not be able to
cope well with long delays or disruption due to frequent
disconnections and node mobility, severely hampering network
operations. A DTN face integrated into NDN-OPP provides the latter
with a robust communications platform supporting communications in
these conditions, by providing the option to propagate Interests to,
and return Data from, remote NDN hosts or networks. These are assumed

to typically reside in access points and wireless edge routers, or
mobile devices and have a corresponding DTN face implementation.

DABBER will employ the DTN face, either in a hop-by-hop or a multi-
hop fashion, when it senses, through the connectivity manager, that
the OPPFaces do not provide a high probability of successful data
delivery (e.g. Time-to-completion is too high). As DTN faces operate
as regular faces, multiple path computation is performed using the
procedure described in section 2.4.

4.2. Interoperability with NDN operation in wired networks

In this sub-section we analyze the interoperability of DABBER with
two potential configurations of an NDN access network based on: a
routing protocol able of disseminating name prefix information, such
as NLSR; a broadcast based forwarding approach.

4.2.1. Interoperability with NLSR

The LSA dissemination mechanism described in section 3.3 is used to
ensure interoperability with NLSR. Such mechanism ensures the
interoperability between a DABBER node and a NLSR edge router, since
the specification used by DABBER follows the same message structure
and sequence of the mechanism used by NLSR [19][20].

However, when DABBER is executing the LSA dissemination procedure
over a Wi-Fi face (towards a NLSR edge router), the following updates
to the procedure described in section 3.3 need to be done in order to
account for the changes between DABBER and NLSR as stated in section
2.3:
    o DABBER will ignore all notifications that Chronosync will send
      related to Adjacency LSAs.

4.2.2. Interoperability with broadcast based forwarding

Broadcast-based forwarding is a common mechanism in the design of
some networks, such as switched Ethernet and mobile ad-hoc networks.
In NDN networks this means that NFD broadcasts Interest packets that
do not match an entry in the FIB, inserting then into the FIB the
forwarding path learned through observation of Data return paths. The
main challenge in broadcast based forwarding schemes is the prefix
granularity problem: determine the name prefix of an inserted FIB
entry from the Data name. Several solutions exist [16], including the
announcements of name prefixes, as done by DABBER.

In any case DABBER interoperability with such NDN networks relies on
the following considerations:

o When in contact with a wireless edge router, DABBER always
forward Interest packet towards the Wi-Fi Face, even when the
Interest packet does not match an entry in the FIB. o Interest
packets received from a wireless edge router will not be
broadcast. Interest packets will be forwarded if they match an
entry in the FIB, or dropped otherwise.

5. Security Considerations

As happens with NLSR, DABBER routing messages are carried in NDN data
packets containing a signature. Hence, a DABBER node can verify the
signature of each routing message to ensure that it was generated by
the claimed origin node and was not tampered with during
dissemination. For this propose, DABBER makes use of a hierarchical
trust model for routing, as used by NLSR within a single domain, to
verify the keys used to sign the routing messages.

Following the name structure described in section 2.2, DABBER models
the trust management as a five-level hierarch, as in NLSR, although
reflecting a different administrative structure: <network> represents
the authority responsible by the international transit network
allowing roaming services; <operator> represents the operator
providing the mobile service; <home> represents the network site of
the mobile operator where the node is registered; <node> represents
the mobile equipment. Each node can create a DABBER process that
produces LSAs.

With this hierarchical trust model, one can establish a chain of keys
to authenticate LSAs. Specifically, a LSA must be signed by a valid
DABBER process, which runs on the same node where the LSA was
originated. To become a valid DABBER process, the process key must be
signed by the corresponding node key, which in turn should be signed
by the registered home network of the network operator. Each home
network key must be signed by the operator key, which must be
certified by the network authority using the network key, which is
called trust anchor in NDN.

Since keys must be retrieved in order to verify routing updates,
DABBER allows each node to retrieve keys from its neighbors. This
means that a DABBER node will use the NDN Interest/Data exchange
process to gathers keys from all its direct neighbors. Upon the
reception of an Interest of the type /<network>/broadcast/KEYS each
neighbor looks up the requested keys in their local key storage and
return the key if it is found. In case a neighbor does not have the
requested key, the neighbor can further query its neighbors for such
key. The used key retrieval process makes use of a broadcast
forwarding strategy, stopping at nodes who either own or cache the
requested keys.

6. Implementation and Deployment Experience

   DABBER is implemented as the routing scheme for the NDN framework for
   Opportunistic Networks (NDN-OPP) [3]. NDN-OPP is an extension of the
   NDN Android implementation, aiming to support NDN communication in
   wireless networks by exploiting direct communication between wireless
   nodes, as well as intermittent Wi-Fi connectivity to the Internet
   (NDN global test-bed).

   NDN-OPP has been demonstrated in ACM ICN 2017 in Berlin [4], as well
   as in the NDNComm in Memphis [5]. NDN-OPP code is available in
   GitHub: https://github.com/COPELABS-SITI/ndn-opp

7. Acknowledgments

   The research leading to these results has received funding from the
   European Union (EU) Horizon 2020 research and innovation programmer
   under grant agreement No 645124(Action full title: Universal, mobile-
   centric and opportunistic communications architecture, Action
   Acronym: UMOBILE).

   We thank all contributors, as well as the valuable comments offered
   by Lixia Zhang (UCLA) and Lan Wang (University of Memphis) to improve
   this draft.

8. References

8.1 Normative References

   [1] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James
       D. Thornton, Diana K. Smetters, Beichuan Zhang, Gene Tsudik, KC
       Claffy, Dmitri Krioukov, Dan Massey, Christos Papadopoulos,
       Tarek Abdelzaher, Lan Wang, Patrick Crowley, Edmund Yeh "Named
       Data Networking", NDN Technical Report NDN-001, October 2010.

   [2] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, I. Moiseenko, Y. Yu, W.
       Shang, Y. Li, S. Mastorakis, Y. Huang, J. P. Abraham, E.
       Newberry, S. DiBenedetto, C. Fan, C. Papadopoulos, D. Pesavento,
       G. Grassi, G. Pau, H. Zhang, T. Song, H. Yuan, H. B. Abraham, P.
       Crowley, S. O. Amin, V. Lehman, M. Chowdhury, and L. Wang, "NFD
       Developer's Guide", NDN, Technical Report NDN-0021, February
       2018.

   [3] Miguel Tavares, Paulo Mendes, "NDN-Opp: Named-Data Networking in
       Opportunistic Networks", Technical Report COPE-SITI-TR-18-01,
       January 2018.

8.2 Informative References

   [4] Seweryn Dynerowicz, Paulo Mendes, "Named-Data Networking in
       Opportunistic Networks", in ACM ICN, Berlin, Germany, September
       2017.

   [5] Seweryn Dynerowicz, Omar Aponte, Paulo Mendes, "NDN Operation in
       Opportunistic Wireless Networks", in NDNcomm, Memphis, USA,
       March 2017

   [6] Christos-Alexandros Sarros, Sotiris Diamantopoulos, Sergi Rene,
       Ioannis Psaras, Adisorn Lertsinsrubtavee, Carlos Molina-Jimenez,
       Paulo Mendes, Rute Sofia, Arjuna Sathiaseelan, George Pavlou,
       Jon Crowcroft, Vassilis Tsaoussidis, "Connecting the Edges: A
       Universal, Mobile centric and Opportunistic Communications
       Architecture", IEEE Communication Magazine, February 2018

   [7] Rute C. Sofia, Igor Santos, Jose Soares, Sotiris Diamantopoulos,
       Christos-Alexandro Sarros, Dimitris Vardalis, Vassilis
       Tsaoussidis, Angela; d'Angelo, "UMOBILE D4.5 - Report on Data
       Collection and Inference Models" Technical Report, September
       2018.

   [8] NDN Project, "NFD Developer's Guide", Technical Report NDN-0021,
       October 2016.

   [9] Zhenkai Zhu and Alexander Afanasyev, "Let's ChronoSync:
       Decentralized Dataset State Synchronization in Named Data
       Networking", in Proc. IEEE ICNP, Goettingen, Germany, Oct 2013

   [10] Minsheng Zhang, Vince Lehman, and Lan Wang, "PartialSync:
       Efficient Synchronization of a Partial Namespace in NDN", NDN
       Technical Report NDN-0039, June 2016.

   [11] Klaus Schneider, Beichuan Zhang, "How to Establish Loop-Free
       Multipath Routes in Named Data Networking", NDN Technical Report
       NDN-0044, April 2017.

   [12] A. Lindgren, A. Doria, E. Davies, S. Grasic, "Probabilistic
       Routing Protocol for Intermittently Connected Networks, IETF RFC
       6693, Aug 2012.

   [13] Waldir Moreira, Paulo Mendes, Susana Sargento, "Social-aware
       Opportunistic Routing Protocol based on User's Interactions and
       Interests", in Proc. of AdhocNets, Barcelona, Spain, October
       2013

   [14] Waldir Moreira, Paulo Mendes, Susana Sargento, "Opportunistic

Routing based on daily routines", in Proc. of IEEE WoWMoM
        workshop on autonomic and opportunistic communications, San
        Francisco, USA, June, 2012

   [15] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based
        forwarding in delay tolerant networks," Mobile Computing, IEEE
        Transactions on, vol. 10, pp. 1576-1589, November, 2011.

   [16] Junxiao Shi, Eric Newberry, Beichuan Zhang, "On Broadcast-based
        Self-Learning in Named Data Networking", in Proc. Of IFIP
        Networking, Stockholm, Sweden, June 2017

   [17] The H2020 UMOBILE project. Grant number 645124, 2015-2018.
        Available via http://www.umobile-project.eu/

   [18] Waldir Moreira, Paulo Mendes and Eduardo Cerqueira,
        "Opportunistic Routing based on Users Daily Life Routine", IETF
        Internet Draft (draft-moreira-dlife-04), May 2014

   [19] Vince Lehman, A K M Mahmudul Hoque, Yingdi Yu, Lan Wang,
        Beichuan Zhang, Lixia Zhang "A Secure Link State Routing
        Protocol for NDN", NDN Technical Report NDN-0037, January 2016.

   [20] Vince Lehman, Muktadir Chowdhury, Nicholas Gordon, Ashlesh
        Gawande, "NLSR Developer's Guide", November 2017.

   [21] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K.
        Scott, K. Fall, H. Weiss, "Delay-Tolerant Networking
        Architecture", IETF RFC 4838, April 2007

   [22] K. Scott, S. Burleigh, "Bundle Protocol Specification", IETF RFC
        5050, November 2007

   [23] C.A. Sarros, A. Lertsinsrubtavee, C. Molina-Jimenez, K.
        Prasopoulos, S. Diamantopoulos, D. Vardalis, A. Sathiaseelan,
        "ICN-based Edge Service Deployment in Challenged Networks"
        (demo), in Proceedings of the 4th ACM Conference on Information-
        Centric Networking, Berlin, Germany, September 26-28, 2017

   [24] Paulo Mendes, Rute Sofia, Vassilis Tsaoussidis, Sotiris
        Diamantopoulos, Christos-Alexandros Sarros, "Information-centric
        Routing for Opportunistic Wireless Networks", in ACM ICN,
        Boston, USA, September 2018.

   [25] Miguel Tavares, Omar Aponte, Paulo Mendes, "Named-data Emergency
        Network Services", in ACM MOBISYS, Munich, Germany, June 2018.

Authors' Addresses

    Paulo Mendes
    COPELABS, Universidade Lusofona
    Campo Grande, 376
    1749-024 Lisboa
    Portugal
    Email: paulo.mendes@ulusofona.pt
    URI: http://www.paulomilheiromendes.com

    Rute Sofia
    Senception
    Av. da Republica 6, 7 Esq
    1050-191 Lisboa
    Portugal
    Email: rute.sofia@senception.com
    URI: http://www.rutesofia.com

    Vassilis Tsaoussidis
    Democritus University of Thrace
    University Campus
    69100 Komotini
    Greece
    Email: vtsaousi@ee.duth.gr

    Sotiris Diamantopoulos
    Democritus University of Thrace
    University Campus
    69100 Komotini
    Greece
    Email: diamantopoulos.sotiris@gmail.com

    Christos-Alexandros Sarros
    Democritus University of Thrace
    University Campus
    69100 Komotini
    Greece
    csarros@ee.duth.gr