Network Working Group
Internet-Draft
Intended status: Experimental
Expires: March 19, 2021

P. Mendes, Ed.
Airbus
R. Sofia
fortiss GmbH
V. Tsaoussidis
Democritus University of Thrace
C. Borrego
Autonomous University of Barcelona

September 15, 2020

Information-centric Routing for Opportunistic Wireless Networks
draft-mendes-icnrg-dabber-05

Abstract

This draft describes the Data reAchaBility BasEd Routing (DABBER) protocol. DABBER aims to provide a name-based routing solution to support the operation of Information-centric Networking frameworks in 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 in a store-carry-and-forward manner, based on a proactive approach. The document describes how to integrate DABBER in a networking node that implements some ICN approach, such as Named-Data Networking (NDN) or Content Centric Networking (CCN), along with the specification of the proactive approach based on the dissemination of name-prefix information.

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 March 19, 2021.

Table of Contents

1. Introduction

This document provides a specification for DABBER, an opportunistic wireless routing protocol that provides a name-based routing solution for the operation of Information-Centric Networking (ICN) [RFC7476] in wireless networks, including opportunistic wireless environments. The term opportunistic wireless networks primarily refers to 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, given that networking nodes availability depends upon different mobility patterns and wireless links are intermittently available due to changes in the environment or changes in the position of the nodes terminating the link. Examples of such wireless environments encompass mobile crowd sensing and Internet of Things, where things are mobile devices, or sensors, including those operating in terrestrial or non-terrestrial networks that integrate smart satellite constellations.

The deployment of information centric wireless networks by applying NDN or CCN alone, inevitably leads to network flooding, due to the lack of knowledge about the best set of paths to reach data holders. As a basic NDN/CCN approach the first set of Interest packets are broadcasted in order to identify the interfaces that lead to a data holder. DABBER aims to prevent such flooding by allowing data holders to announce the name prefix of the data sets they hold. This allows networking nodes to know which are the appropriate interfaces to reach a certain data set prior to the arrival of the corresponding Interest packets. The expected result is the reduction of both network overhead and latency of Interest packets.

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 (e.g., availability, centrality), and not just the history of contacts among wireless nodes, as it is often the case in opportunistic routing [Dlife] [Dlife-draft] [Scorp]. The goal is to better determine windows of opportunity to transmit Interest and Data packets, while opportunity reflects both time and space.

DABBER is devised to be easily integrated with the data plane of CCN/NDN [NFD], since these are well established distributed ICN frameworks.

A DABBER [DABBER18] proof-of concept implementation is available via GitHub [DABBER18-1] and via Google Play [DABBER18-2]. DABBER has been tested in emergency scenarios with intermittent connectivity based on a novel NDN instant messenger, the Oi! Application for Android [OI].

1.1. Assumptions and Design Choices

DABBER relies on the following assumptions with regard to opportunistic wireless environments:

o Nodes are mobile.

o Nodes overhear each other.

o Nodes can be data sources, data destinations, or forwarders.

o Nodes may decide to be the custodians of data transmissions based on a set of criteria, such as local available resources.

o Nodes do not have a complete topology view.

DABBER relies on the following major design choices to develop a name-based routing protocol for opportunistic wireless network:

o DABBER must avoid the network being flooded by Interest packets.

o DABBER is based on the distribution of name prefix (data reachability) only, since adjacency information may present high variability.

o DABBER is based on local information about the context of a node, in order to support the selection of the most suitable neighbours to forward Interest packets.

o DABBER avoids the definition of new messages by making usage of synchronization mechanisms already developed for NDN, such as ChronoSync [ChronoSync][PartialSync].

1.2. 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.

2. Applicability Examples

Mobile Crowd Sensing: This scenario encompasses the implementation of NDN/CCN 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 an environment without/or limited Internet access. These scenarios are also relevant in dense and remote areas.

Smart Satellite Constellation Communication: Delay Tolerant Networking (DTN) [RFC4838] is acknowledged as a relevant technology also for future smart satellite constellations. ICN adds in the support for intermittent connectivity, and relevant properties to support mobility and integrated security.

People-to-people Communication in Emergency Scenarios: A routing solution such as DABBER assists in supporting local data exchange across entities in a way that is agnostic to devices capabilities and identity [Umobile].

Internet of Things: The Pub/Sub receiver-driven nature of ICN brings in recognizable advantages to the Internet of Things, in particular involving mobile nodes, such as fleets of AGVs, UAVs.

3. DABBER Architecture

R relies on the same data structures made available by the Named Data Networking Forwarding Daemon [NFD], namely the Routing Information Base (RIB), the Forwarding Information Base (FIB), and the Pending Intent Table (PIT), as illustrated in figure 1.

DABBER brings no changes to the operation of NFD. However, augmenting NFD with a routing engine for wireless networks requires the inclusion of three new components to NFD. This extended NFD is called NDN-OPP (NDN for Opportunistic networks) [NDN-OPP], as shown in figure 1.

This section provides a description of the extensions included in NFD as well as the new components of the DABBER routing engine.

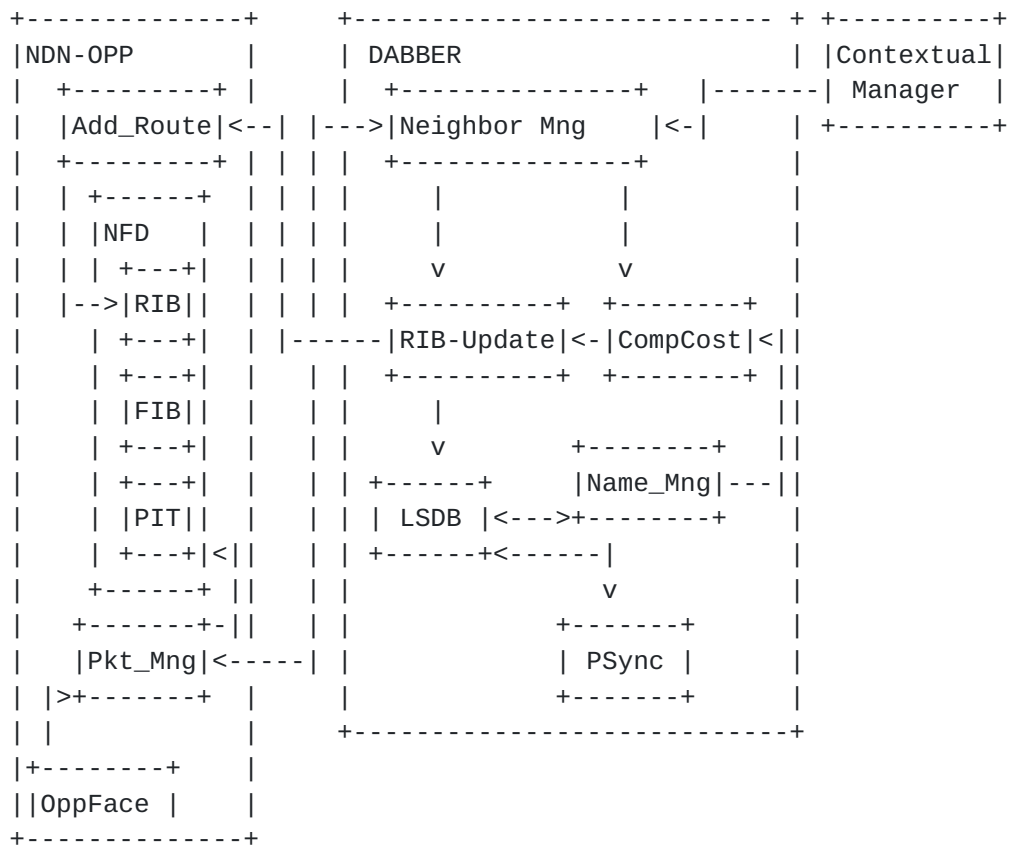

Figure 1: DABBER Architecture.

3.1. Extensions to NFD

In order to augment NFD with a routing engine to allow for the deployment of NDN/CNN in wireless networks three new elements were included in NFD: a method to add routes to the RIB (Add_route in figure 1); a method to gather information (carried names) and status (time of reception) of Interest and Data messages (Pkt_Mng in figure 1); a new face able to handle the intermittent nature of wireless links, called Opportunistic Face (OPPFace) (c.f figure 1).

An OPPFace is a virtual face based on a system of packet queues to hide intermittent connectivity. An OPPFace is created for each neighbor node, where a neighbor node is any node that the current node has ever contacted to directly (i.e. over one radio hop). OPPFaces are kept in the Face Table and their state reflects the wireless connectivity status towards the respective neighbor: they can be in an Up or Down state, depending upon the wireless reachability the corresponding neighbor node.

When packets are dispatched based on the information stored in the FIB, an OPPFace is able to delay packet transmission until there is wireless connectivity to the correspondent neighbor: the OPPFace simply queue packets, when OPPFace is Down, or flush the queue, when OPPFace is Up. Since packet queuing is concealed inside OPPFaces, existing forwarding strategies do not need to be changed.

OPPFaces can be implemented by using any direct wireless communication mode (Infrastructured, Ad-Hoc, and Direct mode). The current implementation of NDN-OPP for Android devices [NDN-OPP-Android] makes usage of group communications provided by Wi-Fi Direct [NDN-emergency][NDN-opportunisticnets]. More information about the integration of NDN-OPP with Wi-Fi Direct is provided in section 6.2.

3.2. Routing Engine

DABBER is a name-based routing protocol that allows nodes to exchange information about name reachability in order to select the most suitable neighbours to forward Interest messages.

The operation of the DABBER routing engine is performed based on the following components, illustrated in figure 1:

o Neighbor_Management: Keep updated information about neighbours in a Neighbor Table. A neighbor entry in that table has information about the neighbor availability, centrality and similarity, as well as about the delay that Interest packets suffer when forwarded via a certain neighbour. The information about neighbours is used to compute the cost of each name prefix announced by them, while the information about the delay of Interest packets is used to update the weights of the neighbors entries in the RIB. The delay of Interest packets is a measure of the time difference between sending an Interest packet and receiving the corresponding Data packet, based on information collected from the Pkt_Mng module of NDN-OPP. The information about neighbor availability, centrality and similarity is collected by an external entity, as described in section 6.1.

o Compute_Name_Cost: This module computes the cost associated with each name prefix as a function of the node similarity of the neighbor announcing the name prefix, and the cost received in the announcement. That cost is used to update the entry of that name prefix / Neighbor in the DABBER internal Routing Table.

o Name_Prefix_Management: This module is responsible to remove and add Link State Announcement (LSA) into the Link State DataBase (LSDB) used by PSync. New LSAs present in the LSDB (announced by a neighbor) are removed in order to pass the following tuples to the

module responsible to compute the costs of name prefixes: Name prefix; Cost; Neighbor.

o RIB_Update: This module keeps updated information about the cost of the name prefixes that will be used to populate the NFD RIB. The information about each name prefix is kept in a local Routing Table based on the tuple: Name prefix; Cost; Neighbor. The cost variable is a function of the cost provided by the Compute_Name_Cost module, plus the information about the availability and centrality of the neighbor, provided by the Neighbor_Management module. In defined periods of time the RIB_Update modules will use the content of the local Routing Table to populate the NFD RIB as well as to insert new LSA into the LSDB.

o LSDB: The Link State DataBase is the structure used by the PSync to keep all the information to be synchronized among neighbor nodes. By placing LSAs into the LSDB, DABBER ensures that such information is passed to all neighbours without the need to define a new set of messages for the exchange of routing information.

Between PSync and ChronoSync, DABBER makes use of the former since it is designed to support partial dataset synchronization. This means that PSync allows a node to subscribe to a subset of data streams, where a data stream is a sequence of data items under a common name prefix. Hence, by avoiding full sync operations of a LSDB in which only a few entries may be new, PSync helps to improve DABBER performance over networks in which wireless links are intermittently available.

4. DABBER Protocol Design

Being developed for wireless networks, DABBER does not rely on the dissemination of Adjacency Link State Advertisements that reflect the status of the links towards neighbor nodes; DABBER only requires the dissemination of Prefix LSAs. DABBER does not require the computation of shortest paths taking into account adjacency information. Instead DABBER replaces the path cost (sum of the weights of all the involved links) used in fixed networks with the computation of data reachability cost based on local available information, reducing the impact that topological changes would have on the stability of routing information. The assumption behind this rational is that the position and validity of the data announced by a data holder changes less than the conditions in all the links making all potential paths.

By exchanging Prefix LSAs each device becomes aware 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 devices 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.

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.

With this in mind, DABBER was designed with the following sequence of operations, as described in the following subsections:

1. Name prefix dissemination.

2. Name prefix cost computation.

3. Update of internal structures (DABBER routing table and LSDB) with the data reachability information of the current node towards the new Name Prefix.

4. Update of NFD structure (RIB) based on the DABBER internal routing table.

Periodically DABBER updates the validity values of all Name Prefixes in its internal routing table, performing the consequent updates of the local LSDB and RIB, and needed.

The updated RIB information is used by NFD to update its FIB, which is used to forward Interest packets, based on the normal operation of NFD. However, independently of the used forwarding strategy, the ranking of faces done by DABBER on the RIB should be respected. For instance an unicast forwarding strategy should use the most important face (lower cost), while a multicast forwarding strategy will use all the faces indicated for the name prefix. After selecting the best set of faces, a copy of the Interest packet is sent to the OPPFaces of the selected faces. The state of an OPPFace reflects the fact

that the corresponding neighbor device is currently reachable or not. Based on this information, the OPPFace decides whether to simply queue the packet or attempt a transmission over the associated Opportunistic Channel.

In what concerns Data packets, they are forwarded following the normal operation of NFD, that is, following the information stored in the PIT.

4.1. Name Prefix Dissemination

DABBER was designed to exploit the synchronization mechanisms made available by NDN, such as PSync to control the dissemination of name prefixes without the need to define neither new messages nor new message passing mechanisms.

This means that while IP-based routing protocols push updates to other routers, DABBER devices pull updates. DABBER can use any underlay communication channels (e.g., TCP/UDP tunnels, Link layer TXT records) to exchange LSA information via an existing mechanism, such as PSync.

By using Interest/Data packets (exchanged by PSync), DABBER benefits from CCN/NDN built-in data authenticity to exchange routing information: since a routing update is carried in an Data packet and every Data packet carries a signature, a DABBER device can verify the signature of each LSA to ensure that it was generated by the claimed origin node and was not tampered during dissemination.

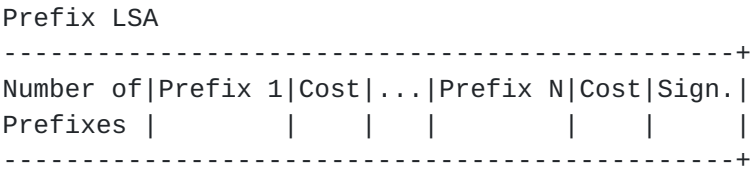

Figure 2: Prefix LSA format.

DABBER advertises Prefix LSAs every time a new name prefix is added or deleted to the LSA DataBase (LSDB). Name prefixes are advertised with a cost metric related to the validity of the associated data, as shown in Figure 2. Each LSA used by DABBER has the name DID/DABBER/LSA/Prefix/version. The DID is described by a scheme based on URIs. It can be for instance network/operator/home/node/. The version field is increased by 1 whenever a device creates a new version of the LSA.

DABBER disseminates LSAs via a data synchronization mechanism (e.g. PSync) of the local LSDB. This peer synchronization approach is

receiver-driven, meaning that a device requests LSAs only when it has CPU cycles. Thus it is less likely a device will be overwhelmed by a flurry of updates. In order to reduce the amount of transferred data, DABBER removes obsolete LSAs from the LSDB by periodically refreshing 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.

DABBER performs the dissemination of LSAs based on a process able to synchronize 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 PSync as a synchronization mechanism. PSync allows DABBER to define a collection of named data in a local repo as a slice. LSA information is synchronized among neighbor nodes, since PSync 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, PSync notifies DABBER to retrieve the corresponding LSA in order to update the local LSDB. DABBER can also request new LSAs from PSync when resources (e.g. CPU cycles) are available.

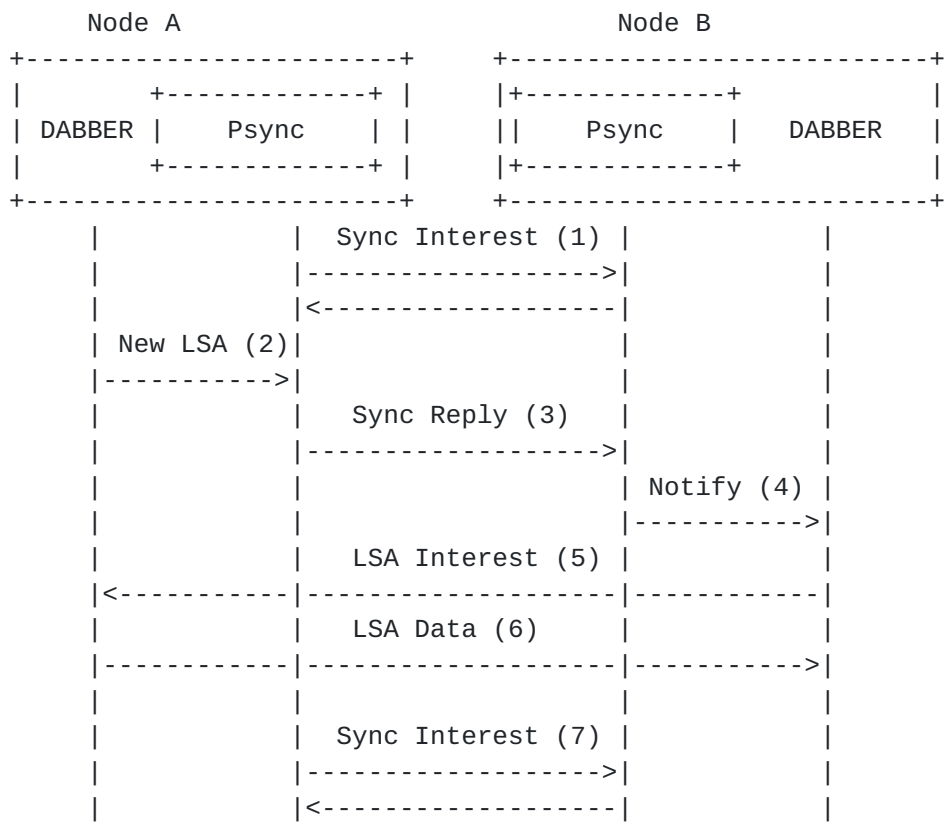

Figure 3: LSA exchange process.

Figure 3 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, PSync, 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 PSync 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 PSync 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. PSync on nodes A and B compute a new hash for updating the set and send a new Sync Interest with the new hash (step 7).

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 also be used to transport information about more than one LSA.

4.2. Name Prefix Cost Computation

When DABBER is notified that a new Prefix LSA was registered in the LSDB or an existing Prefix LSA has a new version, it computes a new cost for each name prefix in such Prefix LSA. The cost of a name prefix is given by its validity.

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 association 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 relevance of the association between two neighbor nodes can be, e.g., a measure of similarity, 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 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. 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/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 an external context-aware agent (rf. to section 6.1, 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 is 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 4: Neighbor table.

The values C, A and S provided by the contextual manager are stored in a Neighbor Table (c.f. Figure 4) 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)

The computation of V' is done by the Compute_Name_Cost module, while the execution of V'' is done by the RIB_Update module (c.f. figure 1). The value of V'' may be updated more often than V', since it is assumed that the properties of the neighbours (C,A) as well as the

delay to a data holder (T) may vary more than the new announcement from neighbours with an updated value of V.

4.3. Update of Internal Structures

After the computation of the cost of the Name Prefix taking into account the relation of the current node with the neighbor announcing it, DABBER updates its internal routing table (operation done by the RIB_Update module) and its LSDB. The information on the routing table will be used to update the RIB of the local NFD and the information of the LSDB will be announced to all neighbors by PSync.

To update the Internal routing table, DABBER adds an entry (Na, V'') for the Name Prefix received from Na, where V'' is the computed cost of the name prefix. The routing table is then ordered by name prefix in decreased order of validity.

Since the current node will also disseminate the received Name Prefix, DABBER updates the cost of the Name Prefix in the LSA stored in its local LSDB in order to consider the computed value V''. For this, DABBER can use two methods:

o Maximal method: Cost of Name Prefix = max (V'', current cost on LSA).

o Average method: Cost of Name Prefix = max (average (cost of Name Prefix entries on local routing table), current cost on LSA).

4.4. Update of NFD Structures

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

DABBER updates the RIB on NFD with the cost k based on three possible logics:

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 logic includes 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 logic includes 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 packets from looping back, there is the need to add a tiebreaker, which assures that traffic only crosses one direction of equal-cost links. This logic includes X neighbors with cost = trunc (100/V''), such that cost is the lowest value of 1/V'' or cost <=1/ V.

In any case the deployment of a DABBER network relies on having all nodes implemented with the same logic.

4.5. Packet Format

DABBER nodes exchange Prefix LSAs by means of a synchronization mechanism such as PSync, without the need to create new packets for the exchange of routing information.

5. DABBER Operational Example

In order to illustrate the proactive routing method defined by DABBER, let's consider Figure 5, where nodes A, B, and C reside in an wireless node running DABBER, while nodes E and F are wireless edge routers running another ICN routing protocol; G is a wireless node running DABBER.

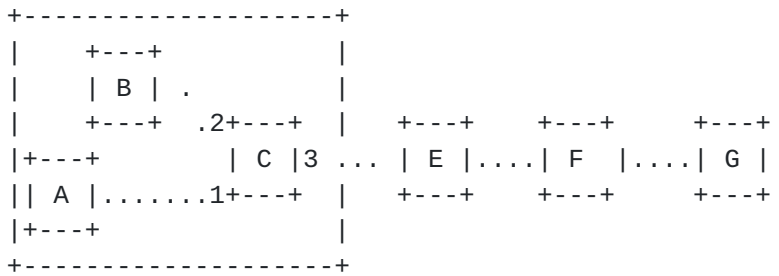

Figure 5: End-to-end operational example.

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 stores in its Content Store a data object with name /NDN/video/Lisbon/river.mpg, which node B received from a neighbor (meaning that B has already synchronized 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 of the cost of the name prefix.

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. This LSA exchange is done when the OPPFaces are up. This means that Node C, which got a new Prefix LSA from nodes A and B, will:

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

o Update 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 the two entries on the routing table of node C for prefix /NDN/video/Lisbon/ will have a validity/cost for A higher than the one for B.

o Update its LSDB with the validity of the current node towards the Name Prefix following the maximal or average methods.

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.

6. DABBER Operational Considerations

6.1. Context Awareness

DABBER defines routing and forwarding strategies that take into consideration, at a network level, the context of wireless nodes, and not just the history of contacts among wireless nodes. Contextual information is obtained in a self-learning approach, by software-based agents running in each networked device, and not based on network wide orchestration. Contextual agents are in charge of computing nodes and link related costs concerning availability and centrality metrics. Contextual agents interact with DABBER via a well-defined interface: the contextual self-learning process is not an integrating part of the DABBER routing framework.

The contextual agent (named Contextual Manager [UmobileD45]) installed in each device 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, 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):

o Node Availability (A) gives an estimate of the node availability based on the usage of internal resources over time and space, as well as the usage of physical resources (battery status; CPU status, etc.).

o Node Centrality (C) provides awareness about the node's affinity network neighborhood context. This means that a list is kept with the following information about each neighbor: neighbour's node degree (number of nodes contacted in a predefined time window); Frequency of contacts between the neighbor and other nodes; Duration of each contact between the neighbor and other nodes; Importance of encountered nodes.

The Contextual Manager keeps values for the mentioned metrics for different periods of time. Encountered nodes can be of different types, such as other mobile devices or wireless access points, for instance.

6.2. Integration with Wi-Fi Direct

When Wi-Fi Direct is used on the MAC layer, there is a one-to-one correspondence between an OPPFace and a neighbor node (for each node detected in a Wi-Fi Direct Group, a new instance of an OPPFace is created). 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). This type of communication is used to exchange small packets that require fast transmission (e.g. emergency apps, Chronosync status messages). The connection-less solution allows peers to exchange a short number of bytes without the establishment of a TCP socket.

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.

Based on the reception of notifications of Wi-Fi Direct regarding changes in the peers detected in the neighborhood, DABBER is able to update its internal peer list (Neighbor Table). 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.

By defining OPPFaces implemented based on a broadcast link layer such as ad-hoc Wi-Fi, DABBER will need to create only one OPPFace in each networked device. 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 entitled 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.

6.3. Integration with DTN

By defining a new DTNFace implemented based on the bundle layer, DABBER could make use of the end-to-end protocol, block formats, and abstract service description for the exchange of messages (bundles) described in the DTN architecture. A DTNFace could provide a robust communications platform for the transmission of Data packets towards the consumer node, making usage of any available custodian nodes.

The bundle protocol [RFC5050] introduces the concept of a "bundle agent" that manages the interface between applications and the "convergence layers" that provide the transport of bundles between nodes during communication opportunities. A DTNFace would extends the bundle agent aiming to control the actions of the bundle agent during communication opportunities.

A new DTNFace would aim to control the reception and delivery of bundles, which are placed in a queue during the forwarding of Data packets. A DTNFace would allows the routing process to be aware of the bundles placed at the node, and allows it to inform the bundle agent about the bundles to be sent to a neighbor node. Therefore, the bundle agent implemented in a DTNFace would need to provide the following interface/functionality to the forwarding process:

o Get Bundle List: Returns a list of the stored bundles and their attributes to the routing agent.

o Send Bundle: Notifies the bundle agent to send a specified bundle.

o Drop Bundle Advice: Advises the bundle agent that a specified bundle may be dropped by the bundle agent if appropriate.

o Acked Bundle Notification: Bundle agent informs routing agent whether a bundle has been delivered to its final destination and time of delivery.

6.4. Producer Mobility

As NDN uses a publish/subscribe communication model, where request resolution and data transfer are decoupled, it is especially relevant to solve the problem of data producer mobility. Supporting producer mobility requires, first of all, finding the new location of the source each time data sources move, and, second, updating the name resolution system according to the new location, in order to maintain the consistency of NDN forwarding.

This could be achieved by using a dissemination method to efficiently discover data sources by replicating Interest messages in an efficient way that avoids network flooding [Optimal-stopping].

With this new feature the prospective forwarders for a given Interest message are limited in number and carefully selected in terms of three criteria:

o Centrality: how well connected a node is in the network. The more central a node is, the bigger the chances are to find a data source.

o Reliability: the likeliness of a node does not drop messages. The more reliable a node is, the least probable is that the Interest message will be discarded.

o Similarity: how alike the contacted candidate node is in terms of shared acquaintances. The less similar, the more likely is that it will find different nodes in the future.

In the current version of DABBER, the degree of diffusion of Interest packets depends on the logic used to update the RIB on NFD (Increase diversity, Downward Path Criterion, Downward Path Criterion extension). It may happen that the diffusion degree could still lead to some overhead. In this case DABBER could be extended to allow nodes to select the most suitable node among all of the neighbors to forward the Interest message based on an optimal stopping approach. This strategy relies on the existence of an optimal set of stopping values such that the nth discoverer node for a certain Interest message is the first contacted node which is the best of all the previously explored nodes, if the node has contacted the first stopping value. If this node is not found, then it will be the first node which is the second best of all the previous nodes, if the node has contacted the second stopping value, and so on. Otherwise, if these nodes are not found, then, the nth discoverer node will be the last nth node before reaching the last contacted node. This makes the dissemination of the Interest messages in Mobile NDNs efficient, even, and pervasive all over the network, increasing the delivery ratio while decreasing the network overhead.

7. Security Considerations

DABBER follows the CCN/NDN security framework built on public-key cryptography, allows it to secure the exchange of routing messages, by being able of verifying the authenticity of routing messages. Moreover, DABBER ensures the right level of privacy of the involved entities, who provide local information to support routing decisions.

Routing security can be achieved not only by signing routing messages, but also by allowing the usage of multiple paths, as done by DABBER: when an anomaly is detected routers can retrieve the data through alternative paths.

Besides the presented security and privacy considerations, the issue of Denial of Service (DoS) needs to be properly addressed. An example is when a malicious user sends a high rate of broadcast messages aiming to exhaust available forwarding resources.

The remainder of this section provides initial insights about the methods used by DABBER to ensure the authenticity, confidentiality of the routing message exchange as well as the privacy of the involved entities.

7.1. Authenticity

DABBER routing messages are carried in Data packets containing a signature. Hence, a DABBER device 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 purpose, DABBER makes use of a hierarchical trust model for routing to verify the keys used to sign the routing messages.

DABBER can model a trust management as a five-level hierarchy, although reflecting a different administrative structure: represents the authority responsible by the international transit network allowing roaming services; represents the operator providing the mobile service; represents the network site of the mobile operator where the node is registered; represents the mobile equipment. Each node can create a DABBER process that produces LSAs.

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 Interest/Data exchange process to gather keys from all its direct neighbors. Upon the reception of an Interest of the type //broadcast/KEYS each neighbor looks up the requested keys in their local key storage and returns 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.

7.2. Confidentiality

Although being deployed under the control of an operator, DABBER allows its network to be extended beyond the reach of its infrastructure network, into scenarios where wireless communications

between involved DABBER devices/router may be spoofed. Hence, DABBER requires routing data confidentiality to ensure the setup of a secure communication topology.

DABBER basic approach relies on the usage of encryption to protect the confidentiality of routing messages. By taking advantage of the semantically meaningful names DABBER relies on approaches such as Named-based Access Control (NAC) [NAC]. NAC provides content confidentiality and access control based on a combination of symmetric and asymmetric cryptography algorithms, while using NDN's data-centric security and naming convention to automate data access control.

Being implemented in wireless devices that may energy constraint, it will be important to verify the efficiency of the cryptographic solution, namely since the generation of asymmetric key pairs as well as the symmetric and asymmetric encryption/decryption operations may be expensive in terms of the usage of resources.

7.3. Privacy

In DABBER, forwarding decisions are taken into account using different metrics such as centrality and similarity. While these metrics may be efficient in terms of node selection, they can breach privacy of network users carrying networked devices by inferring social related information such as position inside groups, as well as information about the devices themselves.

If exchanged as clear text, the information carried in routing metrics may potentially compromise the privacy of users. A way of preserving the privacy of the users in DABBER could pass by using homomorphic encryption for information-centric wireless Ad Hoc Networks.

With homomorphic encryption forwarding decisions are made by performing calculations on encrypted forwarding metric values without decrypting them first, while maintaining low overhead and delays. As a result, forwarding decisions can be taken preserving the user's privacy. For these purposes, homomorphic encryption is extremely useful. This cryptographic scheme allows computations on ciphertexts and generates encrypted results that, when decrypted, match the results of the operations as if they had been performed on plaintexts.

8. IANA Considerations

This document has no actions for IANA.

9. Acknowledgments

The research leading to these results 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).

10. References

10.1. Normative References

[DABBER18] Mendes, P., Sofia, R., Soares, J., Tsaoussidis, V., and S. Diamantopoulos, "Information-centric Routing for Opportunistic Wireless Networks", in Proc. of ACM Information Centric Networking , September 2018.

[DABBER18-1] DABBER Development Team, "DABBER Source Code", https://github.com/COPELABS-SITI/ndn-opp/tree/dabber , September 2020.

[DABBER18-2] DABBER Development Team, "DABBER Android", https://play.google.com/store/apps/details?id=pt.ulusofona.copelabs.ndn , September 2020.

[NDN-OPP] Tavares, M. and P. Mendes, "NDN-Opp: Named-Data Networking in Opportunistic Networks", Technical Report COPE-SITI-TR-18-01 , January 2018.

[NFD] A. Afanasyev, et al, "NFD Developer's Guide", NDN Technical Report NDN-001 , October 2010.

10.2. Informative References

[ChronoSync] Zhu, Z. and A. Afanasyev, "Lets ChronoSync:Decentralized Dataset State Synchronization in Named Data Networking", in Proc. IEEE ICNP , October 2013.

[Dlife] Moreira, W., Mendes, P., and S. Sargento, "Opportunistic Routing based on daily routines", in Proc. of IEEE WoWMoM workshop on autonomic and opportunistic communications, San Francisco, USA , June 2012.

[Dlife-draft] Moreira, W., Mendes, P., and E. Cerqueira, "Opportunistic Routing based on Users Daily Life Routine", IETF Internet Draft (draft-moreira-dlife-04) , May 2014.

[NAC] Z. Zhang, et al, "NAC: Automating Access Control via Named Data", in IEEE MILCOM , October 2018.

[NDN-emergency] Tavares, M., Aponte, O., and P. Mendes, "Named-data Emergency Network Services", in ACM MOBISYS, Munich, Germany , June 2018.

[NDN-OPP-Android] NDN-OPP Development Team, "NDN-OPP Android", https://github.com/COPELABS-SITI/ndn-opp , September 2020.

[NDN-opportunisticnets] Dynerowicz, S. and P. Mendes, "Named-Data Networking in Opportunistic Networks", ACM ICN, Berlin, Germany , September 2017.

[OI] Lopes, L., C. Sofia, R., Mendes, P., and W. Moreira, "Oi! Opportunistic Data Transmission Based on Wi-Fi Direct", in Proc. of IEEE INFOCOM , April 2016.

[Optimal-stopping] Borrego, C., Amado, M., Molinaro, A., Mendes, P., C. Sofia, R., Magaia, N., and J. Borrel, "Forwarding in Opportunistic Information-Centric Networks: An Optimal Stopping Approach", IEEE Communications Magazine, 58(5), 56-61 , 2020.

[PartialSync] Zhang, M., Lehman, V., and L. Wang, "PartialSync:Efficient Synchronization of a Partial Namespace in NDN", NDN Technical Report NDN-0039 , June 2016.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <https://www.rfc-editor.org/info/rfc4838>.

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, DOI 10.17487/RFC5050, November 2007, <https://www.rfc-editor.org/info/rfc5050>.

[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>.

[Scorp] Moreira, W., Mendes, P., and S. Sargento, "Social-aware Opportunistic Routing Protocol based on User's Interactions and Interests", in Proc. of AdhocNets, Barcelona, Spain , October 2013.

[Umobile] C. Sarros, et al, "Connecting the Edges: A Universal, Mobile centric and Opportunistic Communications Architecture", IEEE Communication Magazine , February 2018.

[UmobileD45] R. Sofia, et al, "UMOBILE D45 - Report on Data Collection and Inference Models", Umobile Technical Report , September 2018.

Authors' Addresses

Paulo Mendes (editor)
Airbus
Willy-Messerschmitt Strasse 1
Munich 81663
Germany

Email: paulo.mendes@airbus.com
URI: http://www.paulomilheiromendes.com

Rute C. Sofia
fortiss GmbH
Guerickestrasse 25
Munich 80805
Germany

Email: sofia@fortiss.org
URI: http://www.rutesofia.com

Vassilis Tsaoussidis
Democritus University of Thrace
University Campus
Komotini 69100
Greece

Email: vtsaousi@ee.duth.gr

Carlos Borrego
Autonomous University of Barcelona
Department of Information and Communications Engineering
Bellaterra 08193
Spain

Email: carlos.borrego@uab.cat