Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 16, 2020

P. Mendes, Ed.
Airbus
R. Sofia
fortiss GmbH
V. Tsaoussidis
Democritus University of Thrace
C. Borrego
Autonomous University of Barcelona
March 15, 2020

Information-centric Routing for Opportunistic Wireless Networks
draft-mendes-icnrg-dabber-04

Abstract

This draft describes the Data reAchaBility BasEd Routing (DABBER) protocol, which aims to extend the operation of distributed Information Centric Networking frameworks to opportunistic wireless networks such as Delay Tolerant 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 and Data packets in a store-carry-and-forward manner, based on a combination of proactive and reactive approaches. The document presents an architectural overview of DABBER followed by the specification of the proactive approach based on the dissemination of name-prefix information, and the reactive approach based on the encounters probability.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 16, 2020.

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)[RFC7476], and Delay tolerant Networking (DTN) [RFC4838]. 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 ICN and DTN 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 aiming to support information-centric delay-tolerant networks (ICDTN) [ICN-routing-opp]. These networks are operationally located on the Internet fringes. 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.

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. The goal is to assist in better defining opportunities for the transmission of Interest and Data packets over time and space: DABBER focus on a data plane similar to the one use by CCN/NDN [NFD], since these are well established distributed ICN frameworks.

DABBER brings ICN and DTN together by combining a proactive approach to forward Interest packets based on the dissemination of name-prefix

information, with a reactive approach to forward Data packets based on information collected about custodians and based on encounters probability. The dissemination of name-prefixes and the dissemination of Data packets is done based on the context of nodes, and not just the history of contacts among wireless nodes.

1.1. Applicability

DABBER is being developed to allow the deployment of ICDTN where nodes and links can be intermittently available, such as in the case of emergency situations [NDN-emergency]. 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) 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.

1.2. Assumptions and Requirements

DABBER relies on the following assumptions:

o Mobile nodes are able of exploiting wireless connectivity.

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.

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

o In DTNs it is not possible to know the complete network topology.

o In DTNs it is not efficient to flood the network, as shown by all prior solutions based on controlled packet replication forwarding ([RFC6693][Dlife][Scorp][Dlife-draft]) instead of broadcast as used in Epidemic routing.

o 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).

In terms of requirements:

o Routing informaiton must be exchanged based on Interest / Data messages.

o Routing information 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 Routing information must be distributed to multiple next-hops based on local information that encodes data reachability.

o A synchronization mechanism my be used to exchange routing information among neighbor node.

o Forwarding of Interest packets must take into account the information stored in the Forwarding Information Base (FIB).

o Interest packets must carry information about the data consumer ID.

o Interest packets should carry information about custodians IDs.

o Forwarding of Interest must take into account the information stored in the Forwarding Information Base (FIB).

o Forwarding of Data packets must take into account the information stored in the Pending Information Table (PIT).

o The PIT must store information about the data consumer ID.

o The PIT may store information about custodians IDs.

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

1.3. 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 DABBER protocol architecture. DABBER relies on the same message formats, message exchange process, and same data structures made available by CCN/NDN: Routing Information Base (RIB); Forwarding Information Base (FIB); Pending Intent Table (PIT), while adding new elements such as two new faces (OPPFace and DTNFace), a contextual manager, and distinct forwarding strategies for Interests and Data packets. On contrary to what happens in CCN/NDN, in DTN Data packets may not be able to follow the same path followed by Interest packets.

TBD

Figure 1: DABBER Architecture.

2.1. Routing and Forwarding

DABBER aims to assist in better defining opportunities for the transmission of Interest and Data packets in a store-carry-and-forward manner, based on a combination of proactive and reactive approaches. DABBER defines a proactive routing approach based on the dissemination of name-prefix information, which are use to identifie suitable next hops to reach a certain data object. This location can be the source of data or any other custodian. The proactive routing scheme aims to reduce the time needed to reach the requested data object. Without a mechanism able of disseminating routing information, devices would need to use try and error approach based

on a broadcast forwarding strategy. Besides the extra delay in finding the requested data, such strategy will increase the amount of used networked resources. As shown in figure 2, the proposed proactive approach is 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.

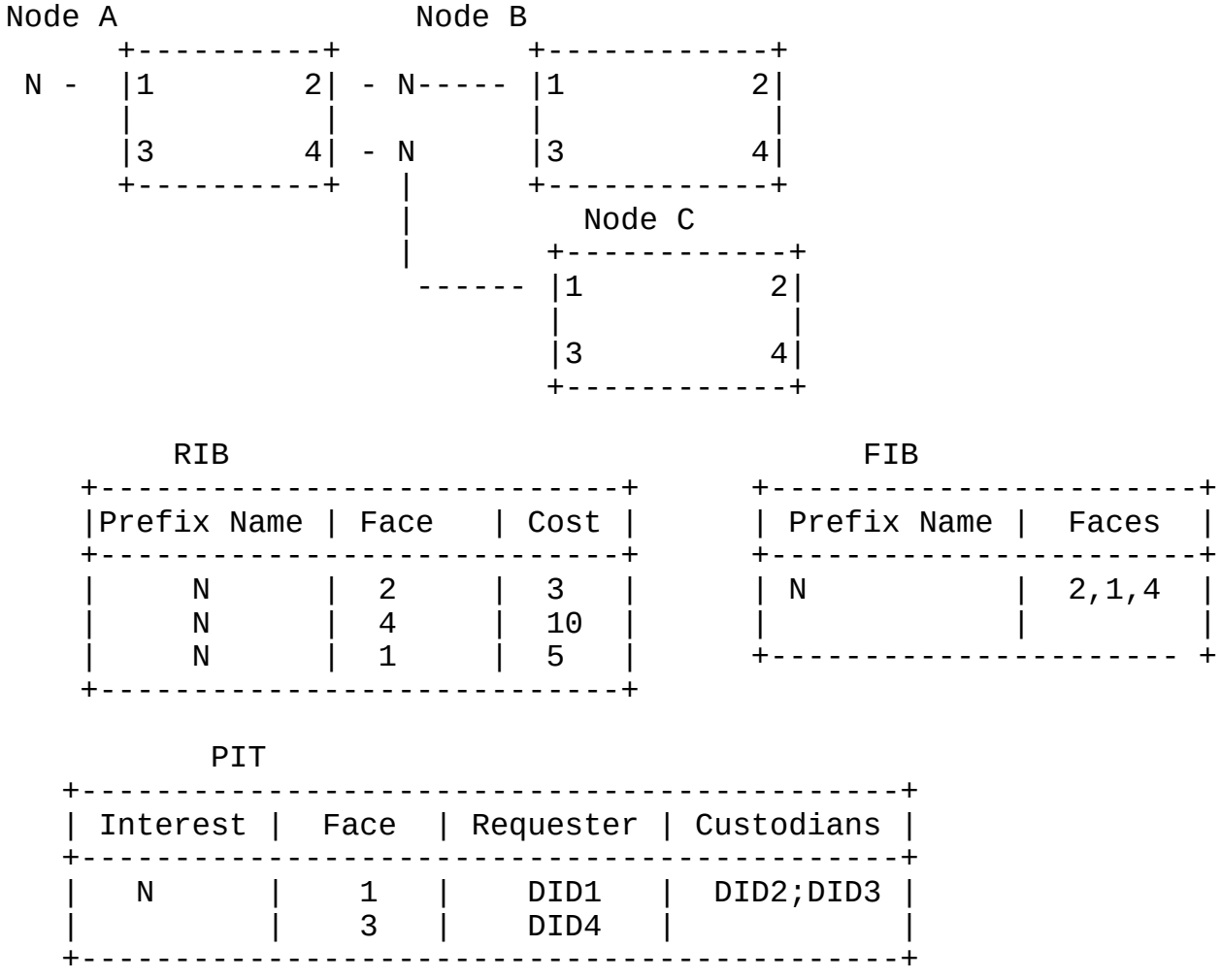

Figure 2: RIB, FIB and PIT on node A.

The FIB illustrated in Figure 2 is used by a forwarding strategy (c.f. section 4.1) used to transmit Interest packets in the direction of one of more copies of the requested data. This strategy is perfectly aligned with the current CCN/NDN architecture. However the same does not happen with the forwarding of Data packets. On CCN/NDN Data packets are transmited in the Faces listed in the PIT for the name carried in the Data packet. Although this breadcumb approach

works on a stable/fixed network, the same does not happen in a DTN, since faces from which Interest packets were received may be down. In this case DABBER forwards Data packets toward the DID of the data requester (mandatory), or to any identified custodian (Optional). This is done by using new forwarding strategy for Data packets based on the encounters probability and contextual awareness, as described in section 4.2.

The inclusion of a forwarding strategy for Data packets is already a difference from the CCN/NDN architecture. To implement such forwarding strategy some changes need to be included to handle Interest packets (c.f. section 4.1) and to the PIT structure, namely:

o The Interest packet includes the DID of the requester device, as well as of any visited custodian device. For this the ApplicationParameter optional field can be used.

o The in-record of the PIT entry related to the Interest needs to hold the following fields: DID of resquester; list of DID of custodians, as illustrated in Figure 2.

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 [Loop-free] in order to prevent forwarding 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 CCN/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.

2.2. Contextual 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 node 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, 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).

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

o Node similarity (S) provides awareness about a node's similarity towards neighbor nodes. This means that a list is kepted with the following information about each neigbour: Packet Error Rate of the wireless link towards the neighbor; Frequency of contacts with neighbor; Duration of each contact with neighbour.

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.

2.3. Device Identifiers

With DABBER, networked devices (producers, consumers, routers) are identified by variable-length identifiers, such as End-points Identifiers in DTN and hierarchical names in CCN/NDN. Using an DID, a node is able to determine the source of a Interest packet as well as a potential set of custodians that may help the data transmission

process. Each device is required to have at least one DID that uniquely identifies it.

Device ID are expressed syntactically as a Uniform Resource Identifier (URI) [RFC3986]. The URI syntax has been designed as a way to express names or addresses for a wide range of purposes, and is therefore has been used to construct names for DTN endpoints, as well as hierarchical names in CCN/NDN. In URI terminology, each URI begins with a scheme name. The scheme name is an element of the set of globally-managed scheme names maintained by IANA. Lexically following the scheme name in a URI is a series of characters constrained by the syntax defined by the scheme. This portion of the URI is called the scheme-specific part, and can be quite general.

Being based on UIRs, device IDs may be kept quite flexible. They might, for example, be constructed based on DNS names, or might look like expressions of interest or intentional names. For instance DIDs may be set up to reflect the network operator to which the mobile node belongs to and to the home site, in case the mobile operator has more than one operational site. In this case, when a mobile node is used outside its home network and some of its requests reach an access point of a visited mobile network, the latter may recognize may be able of checking if there is a roaming agreement between the home network and one of the networks of the visited operator. If so the request may be routed towards an international transit network.

Based on an URI scheme that may reflect a network operator, the information included in the DID 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 a DTN is build based on wireless direct connectivity between nodes that may belong to different operators, but that may have roaming patterns that allows them to have frequent wireless contacts.

2.4. Faces

DABBER leverages the concept of Faces in CCN/NDN to adapt its operation to the intermittent property of wireless connections. This is done by the implementation of two new type of faces, called Opportunistic Face (OPPFace) and Delay Tolerant Networking Face (DTNFace). Besides these two communication interfaces, DABBER keeps a face to the Contextual Manager (CMFace).

2.4.1. OPPFace

An OPPFace is based on a system of packet queues to hide intermittent connectivity: 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. Based on this information, the OPPFace decides whether to simply queue packets (when OPPFace is down) or flush the queue (when OPPFace is up). Since packet queuing is concealed inside OPPFaces, existing forwarding strategy do not need to be changed.

OPPFaces can be implemented by using any direct wireless communication mode. The current specification of DABBER considers Wi-Fi (Infrastructured, Ad-Hoc, and Direct mode).

The current version of the NDN port to opportunitic networks based on Android (NDN-OPP) makes usage of group communications provided by Wi-Fi Direct [NDN-OPP][NDN-opportunisticnets] (NDN-OPP GitHub code [1]). In this case 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 of updating its internal peer list (Neighbor Table as illustredted in Figure 5). 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 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.

2.4.2. DTNFace

By defining a DTNFace implemented based on the bundle layer [RFC5050] DABBER will 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 provides 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. DABBER defines a DTNFace that extends the bundle agent aiming to control the actions of the bundle agent during communication opportunities.

The new DTNFace aims to control the reception and delivery of bundles, which are placed in a queue during the forwarding of Data packets. The DTNFace 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 the DTNFace needs to provide the following interface/functionality to the forwarding process:

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

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

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

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

2.4.3. CMFace

TBD

3. Routing of Name Prefixes

Being developed to operate in DTNs, DABBER does not rely on the dissemination of Adjacency Link State Advertisements (LSAs) that reflect the status of the links towards neighbor nodes; DABBER only requires the dissemination of Prefix LSAs, and does not require the computation of shortest paths. DABBER replaces the path cost used by protocols used for fixed networks with a data reachability cost 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.

3.1. LSA Dissemination

DABBER makes use of Interest/Data packets to have neighbour devices exchanging Prefix LSAs. 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.

By using Interest/Data packets, 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.

DABBER advertises Prefix LSAs every time a new name prefix is added or deleted to the LSA Data Base (LSDB). Name prefixes are advertised with a cost metric related to the validity of the associated data, as shown in Figure 3. Each LSA used by DABBER has the name <DID>/DABBER/LSA/Prefix/<version>. The <DID> is described by a scheme based on URIs (c.f. section 2.1); 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.

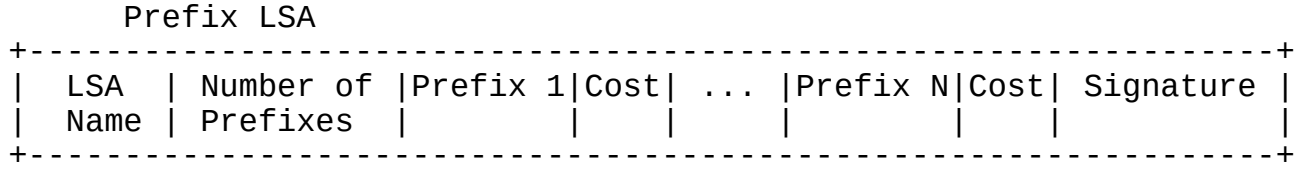

Figure 3: Prefix LSA format.

DABBER disseminates LSAs via a data synchronization mechanism (e.g. ChronoSync [ChronoSync], PartialSync [PartialSync]) 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 transfered 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 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 is 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 4 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).

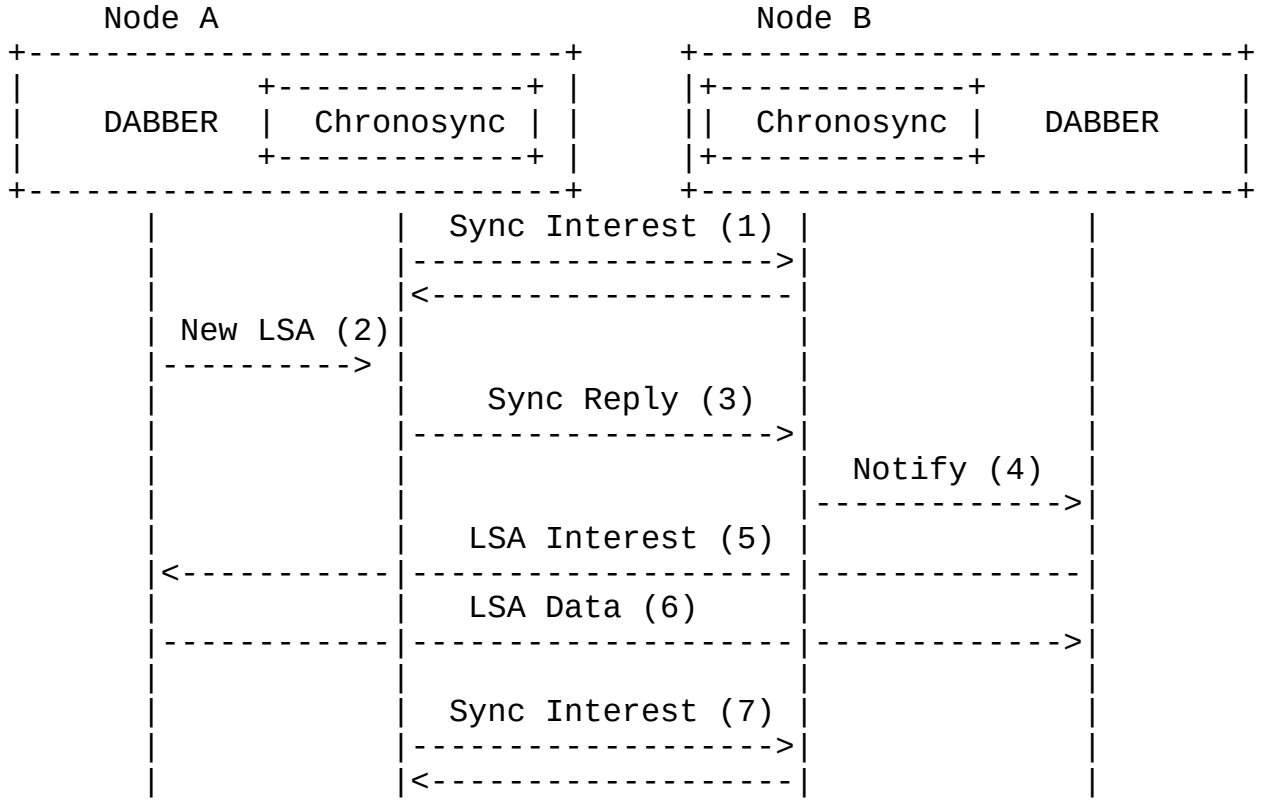

Figure 4: LSA exchange process.

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.

3.2. Multiple path Computation

By exchanging LSAs each devices 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 sequence of operations, as described in the following sub-sections are:

1) Computes a new value for the validity of a new Name Prefix in the LSDB;

2) Updates DABBER internal routing table;

3) Updates the LSDB with the data reachability information (validity) of the current node towards the new Name Prefix;

4) Updates the RIB on NDF based on the DABBER internal routing table, following a Downwards Path Criterion (FIB is updated by NFD based on the RIB content).

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.

3.2.1. 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 "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 [UmobileD45], 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 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 5: Neighbor table.

The values C, A and S provided by the contextual manager are stored in a Neighbor Table (c.f. Figure 5) 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)

3.2.2. Update of DABBER internal routing table and LSDB

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 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 neighbors by Chronosync.

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 (c.f. section 3.2.1). 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).

3.2.3. Update of RIB on NFD

After computing the new value of the cost of a name prefix (c.f. section 3.2.2), 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 is, this logic 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, 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 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, this logic includes X neighbors with cost = trunc (100/V''), such that cost is the lowest value of 1/V'' or cost <=1/ V.

3.3. Routing Operation Example

In order to illustrate the proactive routing method defined by DABBER, let's consider Figure 4, where nodes A, B, and C reside in an ICDTN running DABBER, while nodes E and F are wireless edge routers running another ICN routing protocol; G is a wireless node running DABBER.

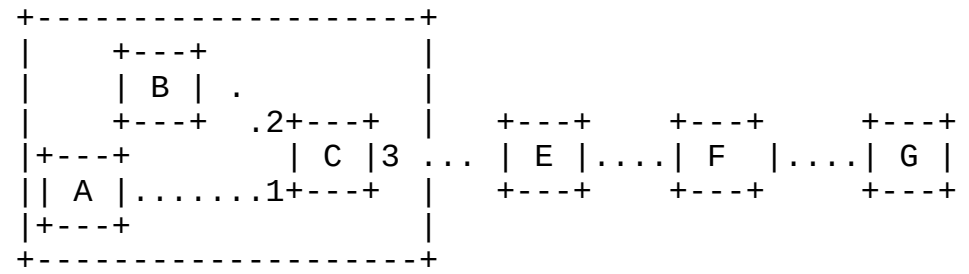

Figure 6: End-to-end operational example.

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 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 3.2.1.

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.

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.

4. Forwarding of Interest Packets

In order to support the new forward strategy for Data packets, devices need to collect information about the DID of the requester (mandatory) and of any potential custodian (optional). Therefore, when an Interest packet is received, the following operations need to be performed:

o The DIDs found in the ApplicationParameter field of the Interest packet are placed in the PIT entry corresponding to the Face over which the Interest packet was received.

o Before forwarding the Interest packet, DABBER will include the DID of the current device if this is a custodian. In this version the role of custodian is pre-configured. This may be revised to include other logics, that may consider the capabilities of the device (e.g. available storage; available energy).

Interest packets are forwarded based on the information that is stored in the FIB, which is updated by the NFD based on information stored on the RIB. Independently of the used forwarding strategy, it has to respect the ranking of faces done by DABBER on the RIB. For instance an unicast forwarding strategy will 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.

Based on the feedback provided by the wireless channel (e.g. Wi-Fi direct confirmation), the OPPFace can decide to remove the packet from the queue once it has been passed on to its intended peer. In

case the packet was not passed to the intended peers, a new attend to forward the packet will be done as soon as the OPPFace is activated: the OPPFace integrates a mechanism to automatically flush the queue whenever the face is brought up upon detection of the corresponding peer being available.

5. Forwarding of Data Packets

By following the operation of CCN/NDN, Data packets are forwarded based in the information holded in the PIT: the ID of the Faces over which a copa of the Data packet must be transmitted. In a DTN network, this setup faces two problems: i) the Face(s) stored in the PIT may not be active since neighbour devices are not in range; ii) the breadcumb path may not be available, since in a dynamic network some of the devices visited by the Interest packet may not be reachable.

To solve these two problems, DABBER makes usage of a new forwarding strategy for Data packets by making usage of information stored in the PIT (which is different from the standard information used by CCN/NDN) and by making usage of an opportunistic forwarding scheme aiming to bring the Data packet closer to the requester or to any available custodian.

The new forwarding strategy works as follows:

o First check if the Face(s) present in the PIT related to that Interest are active. The Data packet is sent to the OPPFace of each active Face. This is a procedure similar to the one used by CCN/NDN.

o For all Faces that are not active (OPPFace is down), DABBER uses an algorithm similar to dlife [Dlife][Dlife-draft] to forward the Data packet closer to the requester or any custodian.

For all OPPFaces that are not active, DABBER starts by collecting the DID of the requester of Data, as well as the DID from potential Custodian from the in-record PIT entry related to that Interest. Based on that information DABBER will forward the Data packet to any active neighbour that has high probability to meet any of these DIDs. This forwarding is done through a DTNFace, which will create a bundle based on the Data packet to be sent.

To forward Data packets, DABBER applies a social opportunistic contact paradigm to decide whether bundle replication is feasible. Its decision is based on social weight (w_(x,y)) towards the bundle’s destination or on the importance (I(x)) of the encountered node (i.e., potential next forwarder) in the system.

If the encountered node has better relationship with the bundle's destination than the carrier in a given daily sample, it receives a bundle copy, since there is a much greater chance for the encountered node to meet the destination in the future. If relationship to the bundle's destination is unknown, replication happens only if the encountered node has higher importance than the bundle's current carrier.

In order to compute the social weight between nodes and their importance, DABBER uses parameters that are determined as nodes interact in the system. A brief explanation of these parameters is given below:

o CD_(x,y): Refers to the contact duration between nodes, i.e., time nodes spent in the communication range of one another, which would allow them to exchange information. Within a given daily sample, different contacts can happen with varied lengths.

o TCT_(x,y): Refers to the total contact time between nodes within a given daily sample. It is given by the sum of all CD_(x,y) in that specific daily sample.

o AD_(x,y): Refers to the average duration of contacts for the same daily sample over different days. It is a Cumulative Moving Average (CMA) of the average duration, considering the TCT_(x,y) of the current daily sample and average duration in the same daily sample of the previous day, AD_(x,y)_old.

o w_(x,y): Refers to the social weight between nodes at a given daily sample. It reflects the level of social interaction among such nodes throughout their daily routines.

o I_(x): Refers to the importance of a node in the system. The importance is influenced by how well a node is socially related to other important nodes.

o N_(x): Refers to the neighbor set of a node x, which it encountered in the current daily sample.

o dumping factor (d): Refers the level of randomness considered by the forwarding algorithm.

o daily sample (Ti): Refers to the time period in which the contact duration will be measured to determine social weight and node importance.

As nodes interact, their CD_(x,y) is collected and used to determine TCT_(x,y), AD_(x,y), w_(x,y), and I_(x) at the end of every daily

sample. If DABBER is configured with a high number of daily samples, the social weight and node importance will be more refined. Thus, it is recommended the usage of twenty-four (24) daily samples representing each hour of the day: the first daily sample refers always to the zero hour of the day when the node is started.

Being able to identify the current daily sample allows a proper computation of social weights and importance. Hence, in the case of node failure (e.g., node crash) or node shutdown (e.g., lack of battery), nodes need to know exactly in which daily sample they stopped interaction, and more importantly how many daily samples have elapsed since then (elapsed_ds). To guarantee that, the equation below is used:

elapsed_ds = cnds * (ed - 1) + (cds - 1) + (cnds - lds) (1)

where:

"cnds" is the configured number of daily samples.

"ed"refers to the number of elapsed days.

"cds" refers to the current daily sample (the one in which the node came back on).

"lds" refers to last daily sample (in which the node failed or shut down).

With this, the node knows how many daily samples have elapsed and can proceed with the update of social weights and importance to reflect the lack of interaction that happen in reality.

5.1. Time-Evolving Contact Duration

The TECD utility function considers the duration of contacts (representing the intensity of social ties among users) and time-evolving interactions (reflecting users’ habits over different daily samples).

Regarding the notations used in the equations presented in this sub-section: sumk(...) denotes summation for k from 1 to n; sumj(...) denotes summation for j from i to i+t-1; sumy denotes summation from all y belonging to N(x).

Two nodes may have a social weight, w_(x,y), that depends on the average total contact duration they have had in that same period of time over different days. Within a specific daily sample Ti, node x has n contacts with node y, having each contact k a certain contact

duration, CD_(x,y). At the end of each daily sample, the total contact time, TCT_(x,y), between nodes x and y is given by the equation below where n is the total number of contacts between the two nodes.

TCT_(x,y) = sumk(CD_(x,y)) (2)

The Total Contact Time between users in the same daily sample over consecutive days can be used to estimate the average duration of their contacts for that specific daily sample: the average duration of contacts between users x and y during a daily sample Ti in a day j, denoted by AD_(x,y) is given by a cumulative moving average of their TCT in that same daily sample, TCT_(x,y), and the average duration of their contacts during the same daily sample Ti on the previous day, denoted by AD_(x,y)_old, as shown in the equation below.

AD_(x,y) = (TCT_(x,y)+(j-1)*AD(x,y)_old)/j (3)

The social strength between users in a specific daily sample Ti may also provide some insight about their social strength in consecutive k samples in the same day, i+k. This is what we call Time Transitive Property. This property increases the probability of nodes being capable of transmitting large data chunks, since transmission can be resumed in the next daily sample with high probability.

TECD is able to capture the social strength w_(x,y) between any pair of users x and y in a daily sample Ti based on the average duration AD_(x,y) of contacts between them in such daily sample and in consecutive t-1 samples, where t represents the total number of daily samples. When k>t, the corresponding AD_(x,y) value refers to the daily sample k-t. In the equation below the time transitive property is given by the weight t/(t+k-i), where the highest weight is associated to the average contact duration in the current daily sample, being it reduced in consecutive samples.

TECD = w_(x,y) = sumj(t/(t+k-i)*AD_(x,y)) (4)

5.2. TECD Importance

As social interaction may also be modeled to consider the node importance, TECDi computes the importance, I_(x), of a node x (cf. equation below), considering the weights of the edges between x and all the nodes y in its neighbor set, N_(x), at a specific daily sample Ti along with their importance.

TECDi = I_(x) = (1-d)+d*sumy(w_(x,y)*I_(y)/N_(x)) (5)

TECDi is based on the PeopleRank function. However, TECDi considers not only node importance, but also the strength of social ties between bundle's current carrier and potential next hops. Another difference is that, with TECDi, the neighbor set of a node x only includes the nodes which have been in contact with node x within a specific daily sample Ti, whereas in PeopleRank the neighbor set of a node includes all the nodes that ever had a link to node x. Note that the level of randomness may vary with the application scenario. Unless previously experimented, it is suggested that dumping factor be set to 0.8.

5.3. Forwarding strategy

Independently of the application scenario, each node MUST employ a forwarding strategy. The first rule is that if the encountered node is the final destination of a bundle, the carrier SHOULD prioritize such bundles by employing the prioritized forwarding strategy, described below.

We use the following notation for the description provided in this section. Nodes A and B are the nodes that encounter each other, and the strategies are described as they would be applied by node A.

5.3.1. Basic Strategy

Forward the bundle only if w_(B,D) > w_(A,D) or I_(B) > I_(A)

When two nodes A and B meet in any daily sample Ti, node A gets from node B: a) the updated list of all neighbors of B, including the social weights that B has towards each of its neighbors, as well as the importance of B; b) the list of the bundles that B is carrying (bundle identifier, plus Endpoint Identifier (EID) of the destination); c) the list of the latest set of bundles acknowledged to B (the size of the list of acknowledged bundles returned by B depends on the local cache size and policy). The information about the social weight, importance, bundle list, and acknowledged bundles received from node B are referenced in node A as w_(B,x)_recv, I_(B)_recv, bundleList(IDn, destinationEIDx)_recv, and ackedBundleList(IDn, destinationEIDx)_recv, respectively.

For every bundle that A carries in its buffer, and i) is not carried by B, ii) has not been previously acknowledged to B, and iii) B has enough buffer space to store it, node A sends a copy to B if B has already encountered the bundle's destination D and its weight in w_(B,D)_recv is greater than A's weight towards this same destination D. Otherwise, bundles are replicated if I_(B)_recv is greater than A's importance in the current daily sample Ti.

Finally, node A will update its own ackedBundleList and discard bundles that have already been acknowledged to node B.

5.3.2. Prioritized Strategy

Similar to the basic forwarding strategy, being the only difference the fact that prior to sending bundles, node A will first send those bundles that have node B as destination.

6. Protocol Addictional Functionality

6.1. Adjustment to data source 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 source mobility. Supporting data source 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 sub-section described a new feature of DABBER which follows a new reactive approach to face the challenges of the data source mobility and consistent forwarding in Mobile ICNs. To this end, DABBER is using the efficient dissemination method for Opportunistic Networks [Optimal-stopping] to efficiently discover data sources by replicating Interest messages in an efficient way that avoids network flooding.

With this new feature the prospective forwarders for a given Interest message (which are denoted as discoverers) 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 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.

A combination of these three criteria defines a reward function (discoverer suitability) of an Optimal Stopping (OS) problem. If a node finds a new node with a certain value for the discoverer suitability it is difficult to know whether this value is a good one

when compared with what a node could find in future contacts. This decision is not trivial: if a node chooses early-contacted discoverer candidates, good results are not guaranteed because selected discoverers could have a low discoverer suitability metric. On the other end of the spectrum, selecting late-contacted discoverer candidates does not guarantee either good discoverer nodes since it is likely that good candidates with high discovery suitability values would be skipped.

DABBER is so extended with the ability to perform an OS-based strategy that allows nodes to select the most suitable node among all of the contacted ones to forward the Interest message. 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. Interoperability

In this section we analyze the interoperability of DABBER with routing and forwarding mechanisms used in wired ICN networks, aiming to study how DABBER can help in ther interconnection of ICDTNs with wired ICN networks. We analyze the interoperability of DABBER with two potential configurations of an ICN access network based on: a routing protocol able of disseminating name prefix information; a broadcast based forwarding approach.

7.1. Interoperability with ICN routing

DABBER LSA dissemination mechanism provides a good interoperability with ICN routing protocols based on link state, which normally exchange information about adjacency and name prefixes. In this scenario the specification used by DABBER ensures a good level of interoperability, since DABBER follows the same message structure and sequence used by such protocols, such as the Named Data Link State Routing Protocol (NLSR).

However, when DABBER is executing the LSA dissemination procedure over a Wi-Fi face, towards an edge router it will ignore all notifications that Chronosync will send related to Adjacency LSAs.

7.2. 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 CCN/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 [Self-learning], including the announcements of name prefixes, as done by DABBER.

In any case DABBER interoperability with such CCN/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.

8. 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, and ensuring the needed levels of confidentiality. 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 remaining 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.

8.1. 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 propose, DABBER makes use of a hierarchical trust model for routing to verify the keys used to sign the routing messages.

Following the name structure described in section 2.3, DABBER can model a trust management as a five-level hierarch, 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.

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 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.

8.2. 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. devices.

8.3. 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 compromising the privacy of users. A way of preserving the privacy of the users in DABBER is to use NDN-P2F [Privacy], a privacy-preserving forwarding scheme that uses homomorphic encryption for information-centric wireless Ad Hoc Networks.

In, NDN-P2F, 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.

There are many homomorphic cryptosystems. A good choice for DABBER can be the Paillier cryptosystem because it is lightweight and, among its properties, it includes the homomorphic addition and multiplication of plaintexts and the homomorphic multiplication by a scalar. The Paillier cryptosystem, however, does not provide a way of calculating the encrypted subtraction, which is needed for metric comparisons. For these purposes, the mapping scheme proposed in [PrivHab] can be used to be able to operate with negative numbers.

9. IANA Considerations

This document has no actions for IANA.

10. 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)[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.

11. References

11.1. Normative References

[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.

11.2. 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.

[ICN-routing-opp] P. Mendes, et al, "Information-centric Routing for Opportunistic Wireless Networks", n ACM ICN, Boston, USA , September 2018.

[Loop-free]
Schneider, K. and B. Zhang, "How to Establish Loop-Free Multipath Routes in Named Data Networking", NDN Technical Report NDN-0044 , April 2017.

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

[Optimal-stopping]
Borrego, C., Borrell, J., and S. Robles, "Efficient broadcast in opportunistic networks using optimal stopping theory", Ad Hoc Networks , May 2019.

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

[Privacy] C. Borrego, et al, "Privacy-Preserving Forwarding using Homomorphic Encryption for Information-Centric Wireless Ad Hoc Networks", IEEE Communications Letters , July 2019.

[PrivHab] S.Carmona, et al, "PrivHab+: A secure geographic routing protocol for DTN", Elsevier Computer Communications , May 2016.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[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>.

[RFC6693] Lindgren, A., Doria, A., Davies, E., and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks", RFC 6693, DOI 10.17487/RFC6693, August 2012, <https://www.rfc-editor.org/info/rfc6693>.

[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.

[Self-learning] Shi, J., Newberry, E., and B. Zhang, "On Broadcast-based Self-Learning in Named Data Networking", in Proc. Of IFIP Networking, Stockholm , June 2017.

[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.

11.3. URIs

[1] https://github.com/COPELABS-SITI/ndn-opp

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