NDN-OPP
Named-Data Networking in Opportunistic Networks

(Version 1.0)

Technical Report COPE-SITI-TR-18-01
January 31st, 2018

Editor
Miguel Tavares (COPELABS/ULHT)

Authors
Paulo Mendes (COPELABS/ULHT)
Miguel Tavares (COPELABS/ULHT)

Executive Summary

This technical report describes the architecture of NDN-OPP, an extension of the Named-Data Networking (NDN) framework for opportunistic networking scenarios.

NDN-OPP was developed based on the NDN Android implementation. NDN-OPP code is open and is available as source code on https://github.com/COPELABS-SITI/ and will be released as APK on Google play.

The research leading to these results has received funding from the European Union (EU) Horizon 2020 research and innovation programmer under grant agreement No 645124 (Action full title: Universal, mobile-centric and opportunistic communications architecture, Action Acronym: UMOBILE). This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein.

Table of Contents

List of Figures

1 Introduction

Information-Centric Networking (ICN) [1] is a new networking paradigm, aiming to evolve the Internet infrastructure by supporting accessing to named data instead of named hosts. In the ICN framework networking becomes independent from the location of data objects, which copies can be stored in several locations within the network, allowing for affordable and pervasive in-network caching and replication. The expected benefits are improved efficiency and security, better scalability with respect to information/bandwidth demand, and better robustness in challenging communication scenarios.

Within ICN, Named-Data Networking (NDN) [2], aims to develop a new Internet architecture that can capitalize on strengths — and address weaknesses — of the Internet's current host-based, point-to-point communication architecture in order to naturally accommodate emerging patterns of communication. By naming data instead of their locations, NDN transforms data into a first-class entity. The current Internet secures the data container. NDN secures the contents, a design choice that decouples trust in data from trust in hosts, enabling several radically scalable communication mechanisms such as automatic caching to optimize bandwidth.

Next-generation wireless networks are expected to support a large set of novel mobile services that may require large data rates and very low latencies. Such requirements bring new requirements for the physical and link layers of novel wireless systems. On the other hand a large set of such services are expected to be pervasively available at low costs. This raises requirements for affordable pervasive access to data services.

Such affordable pervasive access to information is an expected property on the next generation Internet: although the recognized impact of the Internet in areas such as health, education, and agriculture, there are still, some 4 billion people – more than 55% of the world's population – that does not have access to the Internet. The reason is twofold: many live in hard-to-reach areas or do not have access to digital or other basic infrastructure, due to its cost even when living in urban areas.

The barriers towards a bigger Internet usage fall into two major categories: infrastructure, affordability. Lac of proper infrastructure is a big problem for many regions, especially those that are poor or with large rural or remote populations. On the other hand affordability remains a major constraint for the almost 13% of people worldwide who live below the international poverty line, and for those who find Internet access too expensive or do not perceive sufficient value for money from Internet use.

In this context, this technical report aims to describe an extension of NDN for operation in opportunistic networking scenarios, allowing users to use the NDN framework even in the absence of Internet access. The proposed NDN extension for opportunistic networks (NDN-OPP) is able of exploiting any intermittent wireless connectivity: at the moment it makes use of Wi-Fi direct and Wi-Fi infrastructure. NDN-OPP was developed in a modular way, aiming to allow the inclusion of other types of wireless direct communication, such as Ad-Hoc Wi-Fi and D2D-LTE or DTN [12].

2 Operation under Intermittent Wireless Connectivity

NDN-OPP was developed in a modular way aiming to allow the exploitation of any types of wireless direct communication (e.g. Ad-Hoc Wi-Fi, Wi-Fi Direct, D2D) and wireless infrastructure (e.g. Wi-Fi and 3G). The current version, described in this technical report, was developed to operate in a wireless environment where devices can communicate with open Wi-Fi access points (when available), as well as directly via Wi-Fi direct.

Figure 1 illustrate a case of intermittent wireless connectivity, with two groups of wireless devices which are not able to communicate directly among them, although communication is possible within each group, and are also not able to communicate with the Internet via a close by (but not in wireless range) Wi-Fi access point.

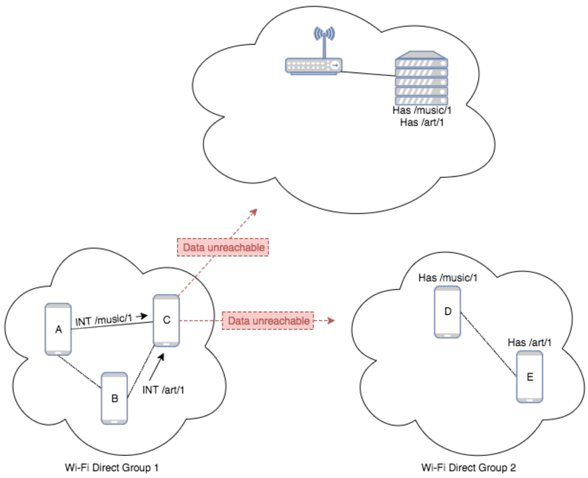

Figure 1 – Failure of Wi-Fi Direct support in NFD-Android.

In the envisioned scenarios, devices should be able of leveraging any opportunity for retrieving data and carry interests beyond the boundaries of the current wireless group or Infrastructure point. For example, when a device expresses an interest within a wireless group, any device in that group should be able of carrying that Interest after it leaves the group being able of retrieving the requested Data from source that gets into its wireless range. Figure 2 shows how Device C can carry an Interest along between two different wireless groups, being able of retrieving the corresponding Data (step 1), and eventually delivering it to interested devices when it returns to the original wireless group (step 2).

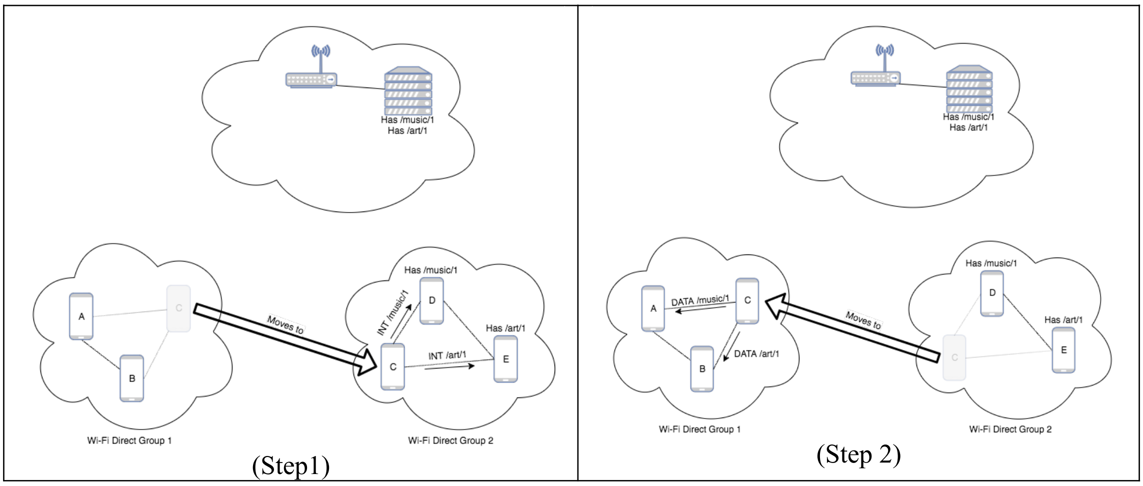

Figure 2 – Leveraging opportunities for Interest/Data exchanges

Alternatively, if Device C comes within range of an Infrastructure point, (c.f. Figure 3) it will be able to pass on the Interests (step 1) to perform normal NDN retrieval of Data (step 2) and carry it back to the original requester (step 3).

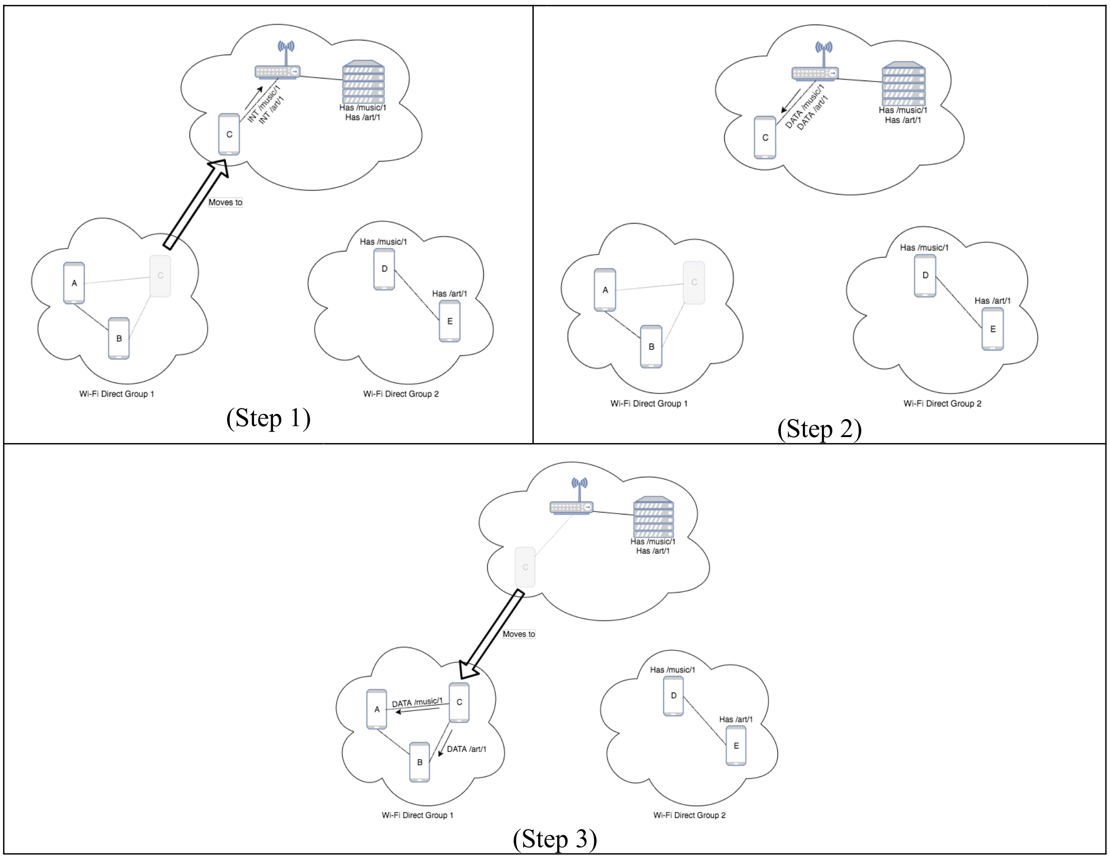

Figure 3 – Leveraging opportunities for Interest/Data exchanges

3 NDN-OPP General Description

NDN-OPP is implemented as an Android application, which combines some software components which are illustrated in figure 4.

First of all, we have a library (JNDN) which works as an interface between applications and NDN-OPP. That library communicates directly with NFD where the NDN logic is implemented.

All components with green background were developed during this project. Those components will be explained in this technical report.

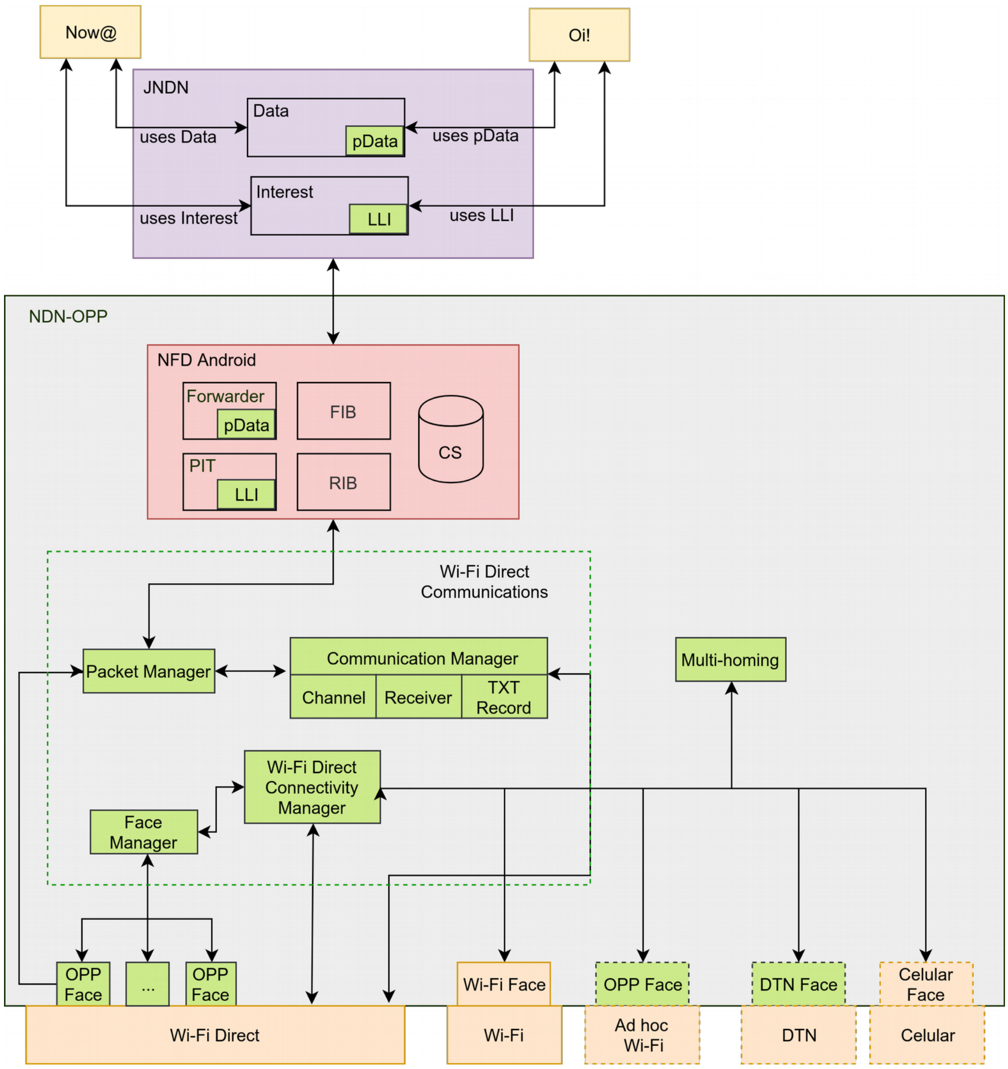

Figure 4 – General Component and their interfaces

NDN-OPP is an extension of the Android implementation of the NDN Forwarding Daemon (NFD), providing the following novel features:

- The concept of an Opportunistic Face (OPPFace), which enables opportunistic communication in an intermittent connected wireless environment. Currently NDN-OPP implements OPPFace on top of Wi-Fi direct, allowing the exploitation of device-to-device communications.
- Two communication models within a Wi-Fi Direct group:

- Connection oriented, in which all packets exchanged between NDN-OPP devices go over the group owner via TCP connections.
- Connectionless, in which case NDN-OPP devices exchange data directly during the Wi-Fi direct service discovery phase (until the maximum allowed by Wi-Fi direct TX records). This form of communication allows packets to be exchange between devices within wireless range, even if not belonging to the same Wi-Fi direct group.

- It implements two methods of push communication:
  - A mechanism based on Long-Lived Interests (LLI), which persist in the network for a predefined time, allowing several data packets to flow based on the same PIT entry. This mechanism can only be used within a wireless network supported by NDN-OPP, since LLI are not supported by NDN basic implementation. LLI are used to send emergence messages to specific authorities within the same wireless network, independent of the topology distance.
  - A mechanism based on pData, which allows a device to send data to a specific name prefix, without the need to previously generate Interest packets. This communication mechanism is only used to send data related to specific emergency name prefixes within a close topological distance (e.g. a citizen requesting help from any nearby person).
- It implements a routing protocol, named Data reAchaBility BasEd Routing (DABBER), to deploy NDN in wireless multi-hop networks without the need to keep the basic requirement of Ad-hoc networks.

The two mentioned methods to implement a push communication models are described in an Internet draft [3], while the DABBER routing protocol is described in another Internet draft [4].

Also out-of-scope of this report is the specification of the Contextual Manager [5], a software component that provides notifications regarding the peers available in its neighbourhood as well as information regarding those peers. This information is to be used in the routing process to compute cost regarding which peer is best suitable for satisfying a certain Interest.

Further information about NDN-OPP can be found in the description of two demonstrations performed in the ACM ICN 2017 conference and NDNComm meeting [6,7].

4 System Components

The NDN Opportunistic Daemon is the encapsulation of a modified version of the NDN Forwarding Daemon, allowing its operation in wireless environments with intermittent connectivity. Tolerance towards intermittent wireless connectivity is built into NDN-OPP based on the implementation of the opportunistic faces (OPPFaces), channels for opportunistic transmission and receivers of intermittent transmitted packets.

As illustrated in Figure 5, when it is started, the NDN Opportunistic Daemon retrieves its generated UUID and reads the NFD configuration file (nfd_conf), after which it stores the starting time and starts the daemon thread. This value is later used to compute how long the daemon has been running. This information is used to show the daemon run time.

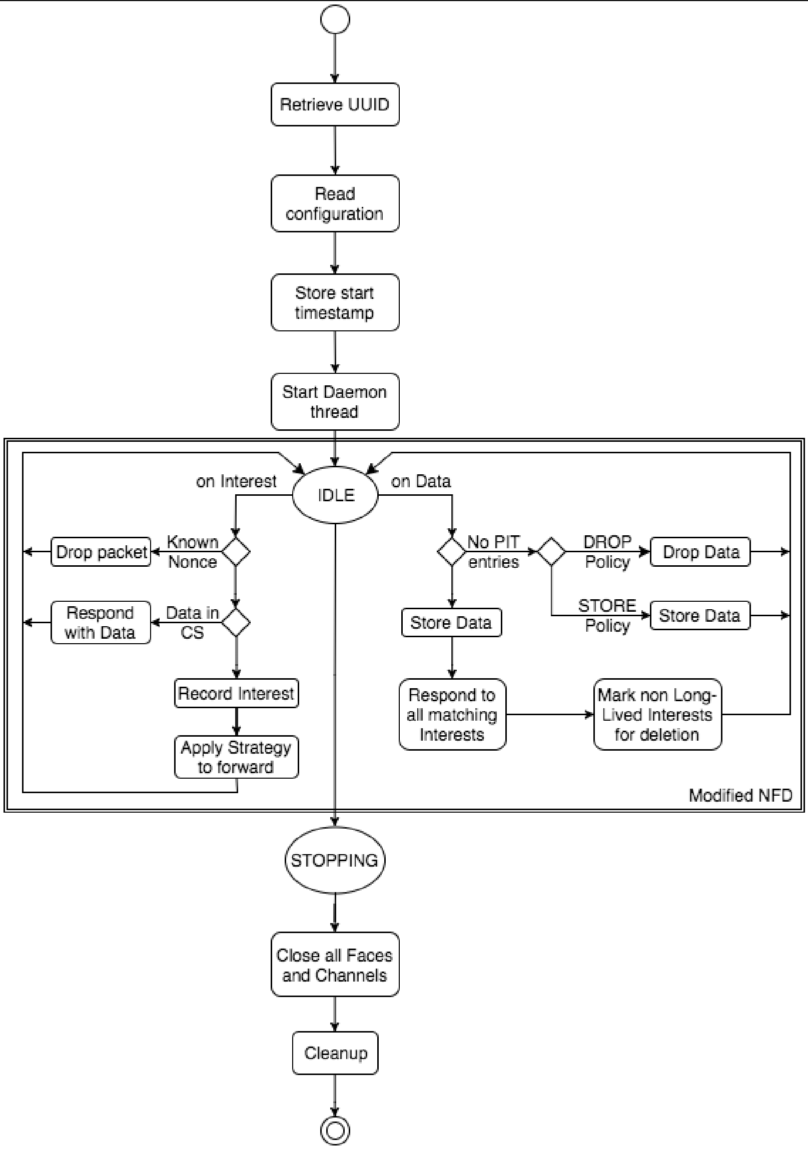

Figure 5 – Flowchart of NDN Opportunistic Daemon

NDN-OPP performs the regular operation of NFD: Upon receiving an Interest packet, the Opportunistic Daemon performs a loop detection check by verifying that the Nonce carried in the packet has not been seen recently. If the Nonce is already known, the packet is dropped to break the forwarding loop. Then, the daemon checks if there is no Data matching the received Interest and if it is the case, responds with the corresponding Data packet retrieved from the Content Store (CS). Otherwise, the daemon performs the standard forwarding process; first, it performs a Longest-Match Prefix lookup on the Strategy Choice Table (SCT) for the name carried in the Interest and applies the corresponding strategy. The strategy may or may not decide to forward the Interest further. Upon receiving a Data packet, the Opportunistic Daemon verifies if there are matching pending Interests stored in the PIT. If not, based on the configured policy, the data packet is dropped. In the event some matching Interests are found, the received Data is forwarded in response on the faces on which those Interests were found.

In addition to the regular pull communication operation of NFD based on the reception of Interests and Data packets, NDN-OPP implements two methods of push communication (for emergency situations) based on the reception of special data packets (pData) and Interest packets (Long Lived Interest – LLI), as described in section 7.

In order to support the operation of NDN in intermittently connected networks, NDN-OPP extended

NFD with a new type of Faces, Opportunistic Faces (OPPFaces), which allows forwarded packets to be queued, being dispatched as soon as a wireless channel is established. The implementation of the OPPFaces depends upon the specific link layer protocols based on two basic policies: In the presence of a peer-to-peer link layer protocol, such as Wi-Fi Direct or D2D LTE, one OPPFace should be created for each wireless neighbour; In the present of broadcast link layer protocols, such as Ad-Hoc Wi-Fi, a unique OPPFace should be created.

The current implementation of NDN-OPP integrates the implementation of OPPFaces for Wi-Fi direct communications, as well as the management of multi-homing towards infrastructure Wi-Fi. Future versions will include the implementation of OPPFaces for Ad-Hoc Wi-Fi as well as to DTN and 3G.

This section aims to provide a detailed description of the components of NDN-OPP, which are grouped into three modules: Opportunistic Faces; Wi-Fi direct communications; Multi-homing.

4.1 Opportunistic Face

In order to support the operation of NDN in intermittently connected networks, NDN-OPP extended NFD with a new type of Faces, Opportunistic Faces (OPPFaces), which allows forwarded packets to be queued, being dispatched as soon as a wireless channel is established. The new concept of Opportunistic Faces allows NDN-OPP to provide support for intermittent communications without requiring any other changes to NFD itself. This makes co-existence of the opportunistic networks side-by-side with traditional communication schemes possible.

The implementation of the OPPFaces depends upon the specific link layer protocols based on two basic policies: In the presence of a peer-to-peer link layer protocol, such as Wi-Fi Direct or D2D LTE, one OPPFace should be created for each wireless neighbour; In the present of broadcast link layer protocols, such as Ad-Hoc Wi-Fi, a unique OPPFace should be created.

The state of an Opportunistic Face reflects the fact that the corresponding neighbour device is currently reachable in the Wi-Fi Direct range. Based on this information, the Opportunistic Face decides whether to simply queue the packet or attempt a transmission over the associated Opportunistic Channel (section 4.3.1.).

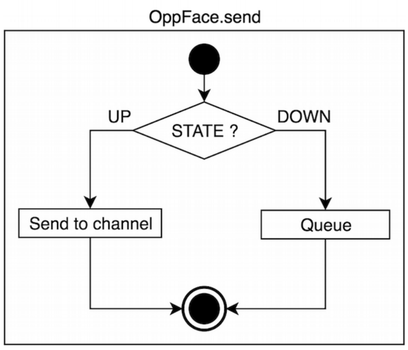

Figure 6 – Flowchart of Opportunistic Face

Based on the feedback provided by the Channel, the Face can decide whether to remove the packet from the queue once it has been passed on to its intended peer. Furthermore, the Opportunistic Face integrates a mechanism to automatically flush the queue whenever the Face is brought up upon detection of the corresponding peer being available in the same Wi-Fi Direct Group.

4.2 Multi-homing

Multi-homing is a component which allows NDN-OPP to operate with more than one face at time. In order to do that, our component keep listening to specific networking events provided by Android. For instance, when NDN-OPP detects that a Wi-Fi connection is established, the multi-homing module automatically creates a face in order to use Wi-Fi network. The same could be implemented for the remaining networking interfaces such as bluetooth.

4.3 Wi-Fi direct communications

4.3.1 Packet Manager

When a packet arrives to NDN-OPP from a OPPFace, that packet goes directly to the Packet Manager. This component performs the decision of which hypothesis shown on section 5.1 is going to be used. The Packet Manager also receives all packets from Wi-Fi Direct on NDN-OPP. Hence, these packets are dispatched to NFD.

4.3.2 Communication Manager

The decision of using a connection-oriented or connection-less communication is performed by the Communication Manager, as illustrated in Figures 7 and 8. The reasoning is as follows:

- Hypothesis 1: Packets are exchanged between two wireless peers over a reliable TCP socket is such socket already exists;
- Hypothesis 2: If a TCP connection does not exist, decision is take based on packet size. The connection-less mechanism is used for fast dispatching of small packets (limited to the size if the TXT record, which depends upon the Android version; Bigger packets will require the establishment of a reliable TCP connection over the Wi-Fi direct group owner.

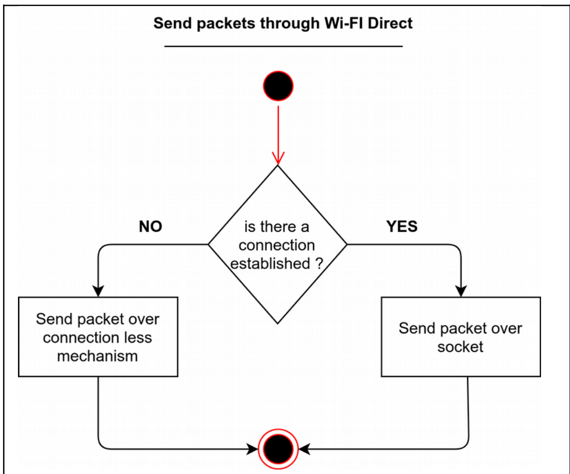

Figure 7 – Connection already exists

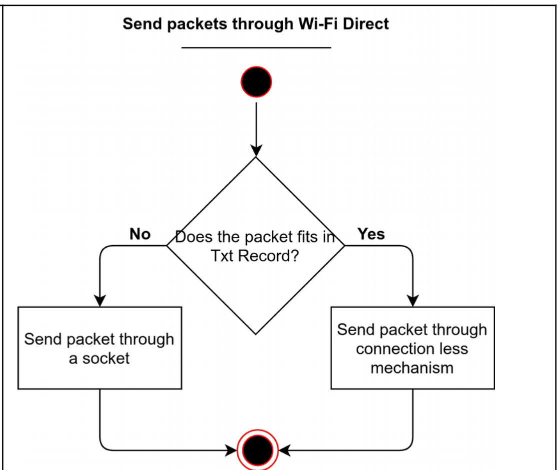

Figure 8 – Decision based on packet size

Both communication methods are configurable on NDN-OPP. For instance, if we want to transfer packets over a socket every time that is possible, we must select the first hypothesis. Otherwise, could be more convenient do a selection based on packet size, in this case, we must select the second hypothesis.

After deciding which communication method to use (connection-oriented or connection-less), communication manager takes care of all the issues related with the transmission and reception of packets: handling channels and receivers in the connection-oriented method and handling the operation of TXT record for the connection-less communications.

Opportunistic Channel

The Opportunistic Channel implements the communication medium through which packets are effectively passed between nodes when using the connection-based modules. In reaction to a new peer being detected as available, the Opportunistic Face Manager creates a new Opportunistic Channel with the connection details that are required to transfer packets to the remote peer.

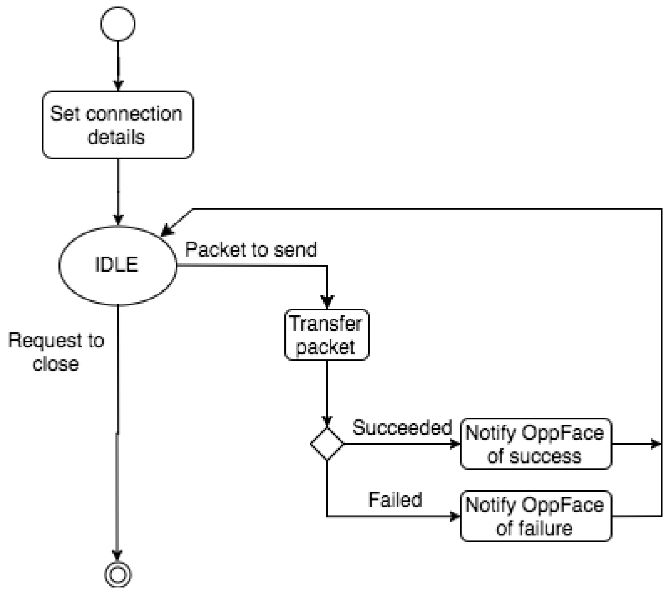

Figure 9 – Flowchart of Opportunistic Channel

Whenever the NDN Opportunistic Daemon, as part of its forwarding process, decides to send a packet through the Face to which this Channel is attached, the connection details are used to effectively transfer the packet and the result of the transfer is then sent as feedback to the Face.

TXT record

A TXT record is a type of resource record in DNS. In our case we are using this feature to transfer data among devices without establish a TCP connection. As you can read on the section 5.3, TXT records have some limitations related with the amount of data that they are transporting.

Opportunistic Receiver

The Opportunistic Receiver is the component that handles the reception of packets from other peers. This component handles the reception of packets coming from other peers connected to the same Wi-Fi Direct Group and subsequently dispatches it to through the appropriate Opportunistic Face.

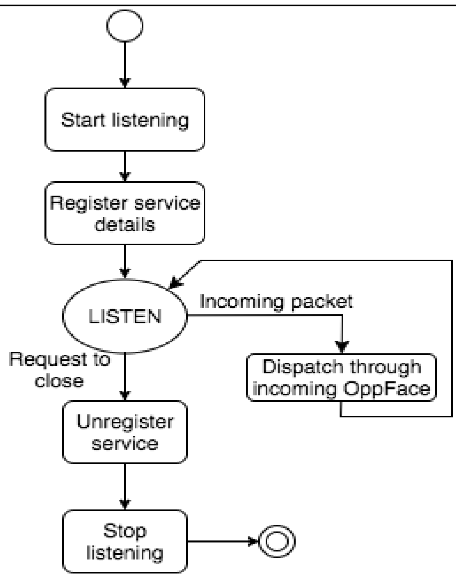

Figure 10 – Flowchart of Opportunistic Receiver

4.3.3 Connectivity Manager

The Connectivity Manager takes care of the entire logic of forming, joining or maintaining Wi-Fi Direct Groups. In order to build a Wi-Fi Direct network, the Connectivity Manager starts by electing the group owner. The group owner will be responsible of handle connection requests and reacts as DHCP server. If the group owner disappears, a new one will be elected. The Opportunistic Connectivity Manager also manages the connection to groups in response to whatever changes are detected in the vicinity.

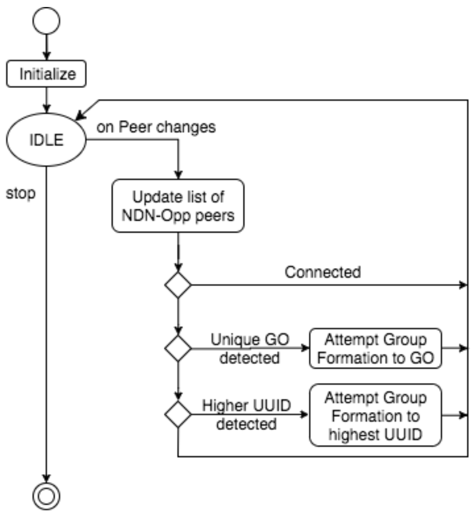

Figure 11 – Flowchart of Opportunistic Connectivity Manager

As illustrated in Figure 11, the Connectivity Manager reacts to changes in the list of peers available within scanning range of the current device. It oversees managing the connection to a Wi-Fi Direct Group and automatically joins a group if it is not part of one. All the intelligence required to manage the connectivity and transition across groups, as the current device moves around, is implemented in this component.

Whenever notifications regarding changes in the peers list are received, the Connectivity Manager updates its internal peer list. If the device is not currently connected to a Wi-Fi Direct Group, it performs a selection heuristic to determine to which group 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.

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

The Connectivity Manager reacts to notifications regarding changes related to the availability of peers in the vicinity. In order to do that, we used Wi-Fi Direct discover peers, and discover services to filter the devices that are using NDN-OPP. The component that manages this list is the Opportunistic Peer Tracker.

This task requires additional experimentation to answer a wide array of questions regarding the most appropriate strategy of forming and maintaining groups; how many devices per group; how long to remain connected before moving on to the next group; handling loss of connectivity from the current group. All these questions require more thorough investigation.

4.3.4 Face Manager

The Opportunistic Face Manager is in charge of translating changes about the availability of peers into the status of their corresponding Opportunistic Faces. In order to do that, this component uses a Wi-Fi Direct functionality called discover peers. This functionality performs Wi-Fi scans in order to discover Wi-Fi Direct devices around.

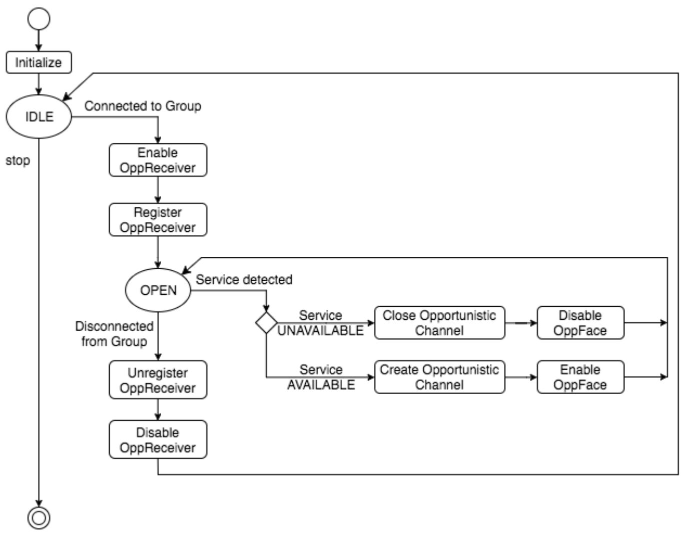

Figure 12 – Flowchart of Opportunistic Face Manager

The Face Manager also maintains the Opportunistic Faces in order to allow a device to receive packets from neighbour devices. In order to achieve this, whenever the current device successfully joins a Wi-Fi Direct Group, it enables the Opportunistic Receiver and registers it within the Group so that other connected peers can send packets to this device. After this setup, all service changes detected within the Group (peer available or unavailable) are reflected into the Face Table of the NDN Opportunistic Daemon.

Upon disconnection from the Group, the Opportunistic Receiver is disabled and unregistered and the device returns to a state of waiting for a Group to be joined.

5 Direct Wireless Communication System

This section describes the two communication methods implemented in NDN-OPP to allow the direct communication between wireless neighbours over Wi-Fi Direct: connection-oriented and connection-less.

The connection-oriented solution allows peers to exchange packets by means of a reliable TCP connection established over the Wi-Fi direct group owner. This type of communication system is used mostly for large packets that may require reliable communication. The connection-less communication method allows peers to exchange packets directly without the establishment of any Wi-Fi direct group, by exploiting the TXT records available during the Wi-Fi Direct service discovery phase. This type of communication is used to exchange small packets that require fast transmission (e.g. emergency apps, Chronosync status messages).

The connection-oriented solution requires the formation of a Wi-Fi direct group by the connectivity manager and the control of the OPPFaces towards all neighbours by the Face manager. In the connection-less solution, packets can be transferred immediately, as part of the service discovery exchange. However, these transfers are limited in the amount of bytes that can be passed successfully.

5.1 Connection-oriented solution

The connection-oriented solution establishes a connection over a TCP socket to the next hop over the Wi-Fi Direct group owner. To do this, the sender must know the IP address of the destination peer in order to create peer-to-peer connection to transfer the required packets.

Once a device joins to the group, it will receive a list containing information related to each connected device. Since we are working with opportunistic networks, it is common that some devices could join or leave the group frequently. In order to deal with it, the group owner is also responsible to keep the list of connected devices updated.

When NDN-OPP needs to transfer NDN packets through connection-based solution, first of all, we perform a search into the list that contains the information related to connected devices. Once we found the destination's IP address, a TCP socket is instantiated in order to serialize and send packets. With the intention of receive these packets, the recipient instantiates a server socket once it joins to a Wi-Fi Direct network. Therefore, this is a multi-thread mechanism in order to be able to receive multiple packets. Without this implementation, some packets could suffer some delay or even be lost.

The following figure describes how the connection-oriented solution works.

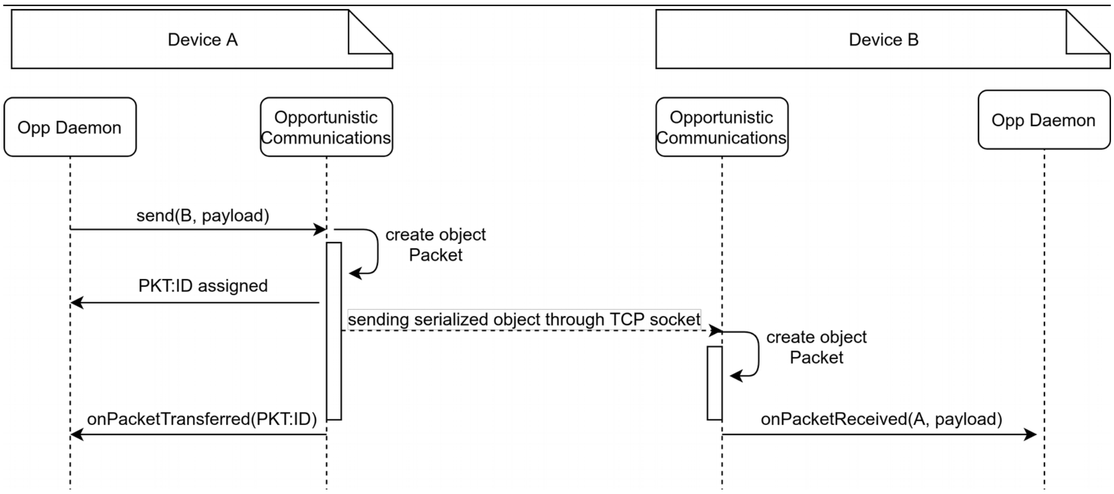

Figure 13 – Connection-oriented solution flowchart

When the NDN-OPP daemon needs to transfer a NDN packet, the payload is encapsulated in an object Packet. That object contains the payload to be sent, as well as the sender and recipient UUIDs of that packet. Per each packet object is generated, by the sender, with a PKT ID. This identifier is used to notify the sender when packets were transferred. On the recipient side, upon reception of the packet object, the Opportunistic Receiver will notify the NDN-OPP daemon that a packet was received with a certain payload from device A.

5.2 Connection-less solution

The previous solution requires the establishment of a connection, a process that might be too lengthy in the cases where a small number of packets need to be exchanged. This is made even worse in the case of higher mobility, where wireless contracts between devices are short, as so devices would be spending more time establishing and tearing down wireless connections in contrast with the time spent exchanging packets.

In order to cope with such situations, NDN-OPP implements a connection-less solution that enables devices to perform exchanges as soon as the probing phase successfully results in the discovery of a device.

In contrast with the connection-oriented mechanism, in the connection-less solution NDN-OPP allows peers to exchange a short number of bytes without the establishment of a TCP socket. To use this methodology, each device that is running NDN-OPP is always listening to all announced UUID related with it. Every time that a device needs to send a packet, it serializes the packet and starts announcing it over TXT Record with the UUID of the destination. When the recipient sees that there is a packet with its UUID as destination, it reads the packet and send it to NFD. Android versions below 5.1 allows us to transfer 88 bytes using this connection-less mechanism, in versions above 5.1, we are able to transfer up to 900 bytes.

The Connection-less Transfer Manager (CLT Manager) is the central component that encapsulates the entire logic required to perform the connection-less solution. This manager notifies the Opportunistic Daemon whenever a packet is (a) successfully transferred to its intended recipient or (b) received for the Daemon to process.

In order to implement a reliable connection-less solution, the CLT Manager maintains a TXT Record for each intended recipient of packets, which contains data packets and an acknowledgement list. Since the sequence and order in which devices probe and respond to one another is not controlled, a device might perform the acknowledgement of a given packet received from a remote peer, but might receive the packet again in the next TXT Record in the event the remote peer does not successfully probes the current device to get the pending acknowledgements.

When a remote peer becomes available (comes within transmission range), the state of the Opportunistic Face is set to UP. As a consequence, it begins by passing the Interests and Data that are stored in the respective queues. When the Opportunistic Daemon has a packet to send (Interest or Data), it notifies the CLT Manager, which assigns an identifier to the packet. Such identifier is returned to the Opportunistic Daemon to identify future events regarding this packet transfer. At this stage, one of two things can happen with respect to this transfer: (a) the packet can be acknowledged by the recipient; or (b) the transmission can be cancelled by the sender. The first situation corresponds to the fact that the CLT Manager at the intended recipient has acknowledged the reception of the packet by placing its Identifier in its own acknowledgement list. The second situation corresponds to the case where the packet is an Interest whose lifetime expired.

When the CLT Manager receives a TXT Record from a peer, which is intended for it, it notifies the Opportunistic Daemon about all the newly received packets. Based on the contents of the received TXT Record, it also updates the contents of its own TXT Record intended for that particular peer. In particular, the following situations need to be handled to reliably perform the transfer of a packet:

1) All the packets that have their identifiers in the acknowledgement list of the received TXT Record are removed from the TXT Record sent to the remote peer. Also, the Opportunistic Daemon is notified that the packet has been successfully transferred so that the Opportunistic Face can remove it from its queue.

2) All the identifiers of the packets received in the previous TXT Record, which are not included in the new record, are removed from the acknowledgements list of the TXT Record that this device sends to the remote peer.

3) All the identifiers for the newly received packets in the current TXT Record are added to the acknowledgements list of the TXT Record that this device sends to the remote peer.

This process is illustrated the following figure.

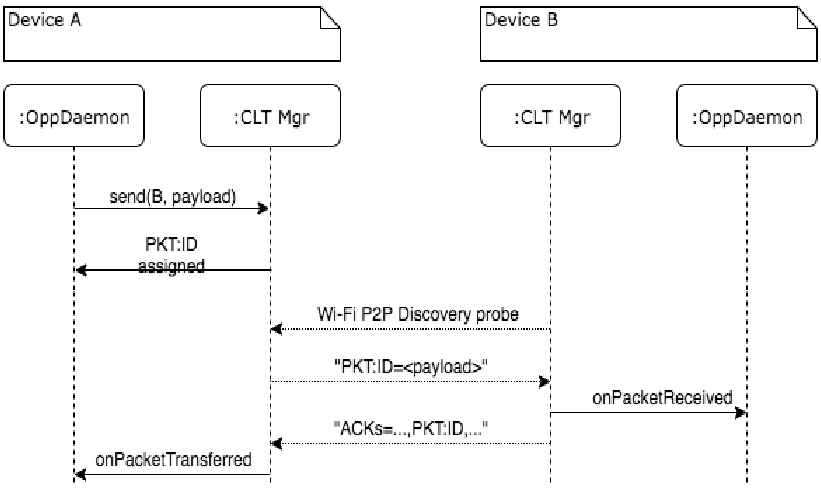

Figure 14 – Sequence diagram of the acknowledgement mechanism of the CLT Manager

6 Multi-homing Wireless Communication System

Although NDN-OPP was developed to allow the operation of NDN over an opportunistic wireless network, NDN-OPP also allows devices to exploit the present of Wi-Fi infrastructure connections made available by a nearby Wi-Fi Access Point (AP).

For this propose NDN-OPP makes usage of the new OPPFaces (describe in this document) to allow direct wireless communication between mobile devices, and also of the Wi-Fi Face previously implemented in NFD. The Wi-Fi face may provide higher communication resilience and lower delays, as well as the possibility for NDN-OPP devices to exchange data with other NDN-OPP or standard NDN devices deployed in a far away location.

Currently NDN-OPP makes usage of a broadcast forwarding strategy. Meaning that Interest packets are forwarded to all OPPFaces (independently if they are UP or DOWN), as well as to available Wi-Fi Faces. Other multi-homing strategies can be used. For instance Interest packets may be forwarded to a subset of available faces (OPPFaces of Wi-Fi Faces) depending on decisions made by the DABBER protocol. A more static alternative can be to avoid replicating Interest packets among wireless peers devices when an Wi-Fi network is available, as shown in following figure.

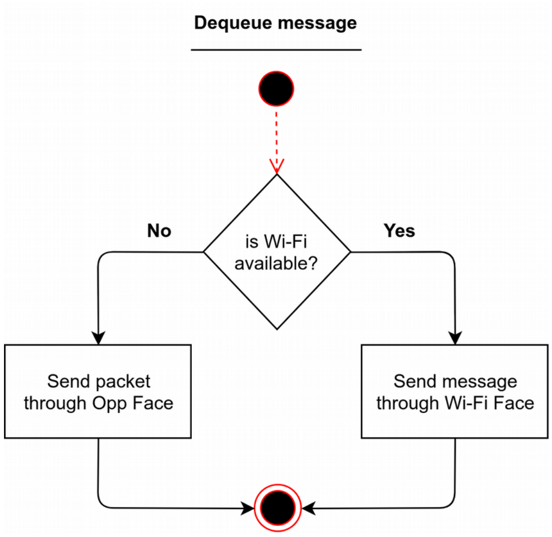

Figure 15 – Static Multi-homing configuration

As illustrated in figure 15, to implement this feature NDN-OPP needs to keep listening changes on Android Wi-Fi interface. As soon as there is a good Wi-Fi connection established to an AP, NDN-OPP creates a Wi-Fi face and also a route to the NDN router that is announced via the local AP.

It is expected that NDN-aware APs (as the ones developed within the UMOBILE project) will be able to provide mobile devices with the IP address of the local NDN router within wireless beacons. In the current version of NDN-OPP this information is made available during the configuration phase. By default, NDN-OPP is configured to connect to the NDN router deployed in COPELABS. Nevertheless, we are able to select another NDN router. In order to do it, we should tap on the menu the option "Connect to NDN Testbed", and then select a node from the list.

NDN-OPP does not need to care about how the Wi-Fi Face is managed: NFD Android already does that by itself. This algorithm is also described on the flowchart right below.

7 Support for Push-communication Methods in NDN

As mentioned, NDN-OPP implements two methods of push communication: i) a mechanism based on Long-Lived Interests (LLI), which persist in the network to a predefined time, allowing several data packets to flow; ii) a mechanism based on pData, which allows a device to send data to any device in a short range (topology distance defined by the pData), without the latter one sending Interests.

This section provides a brief introduction to these two methods.

7.1 Push communication based on Long-Lived Interest

NDN-OPP uses the same packet format as the original NDN specification with a minor modification by the addition of a specific TLV for marking Long-Lived Interest. Figure 16 illustrates a typical exchange of packets between two devices running and encountering each other. As a device A receives an Interest and determines that device B is the next peer to pass it on through its Opportunistic Face, the packet is stored given that B is currently unreachable. Upon the establishment of the Wi-Fi Direct connection, the Opportunistic Face flushes its queue and transmits the pending Interest packet to device B. The Interest continues its path from peer to peer. At a later point, the Data packet comes in response to the forwarded Interest. In the same way, the packet is queued until device A connects with B over a Wi-Fi Direct connection again. At this point, the packet is transmitted to A, which passes it on to whoever sent it the original Interest.

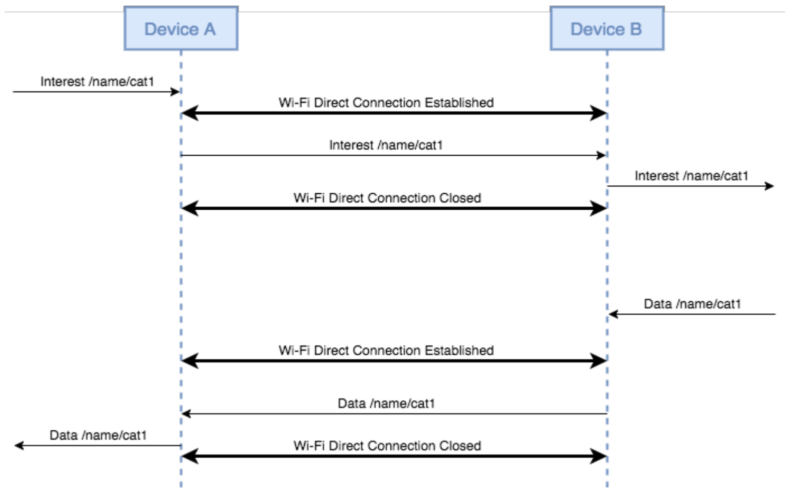

Figure 16 – Standard exchange of NDN packets with intermittent connectivity

The modification we consider for the packet specification enables the implementation of a notion of Long-Lived Interest, where pending Interest can be kept in the PIT even after matching Data has satisfied them. This modification means that an NDN node can setup a method for other nodes to push messages to it. The Long-Lived Interest modification effectively restricts the conditions under which a pending Interest is removed from the PIT by only considering its normal expiration at the end of its inscribed lifetime. All Data packets matching the name of a Long-Lived Interest will follow the breadcrumb back-forwarding of Data [2].

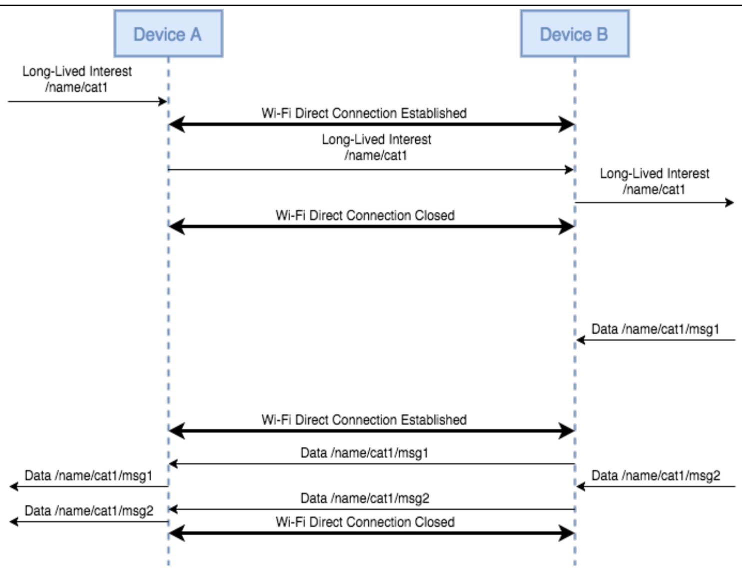

Figure 17 – Exchange of Long-Lived Interests with intermittent connectivity.

Figure 17 presents one such example. Some device A receives a Long-Lived Interest for /name/cat1 and passes it on according to whatever decision the forwarding strategy decides and device B forwards it further as well. At a later point in time, a Data packet is received by B, which forwards it back through the face on which the Long-Lived Interest was previously received. However, as a second Data packet matching the same name is received, it is also forwarded down the same face as the first Data packet. This second transmission can be fully performed in the context of the current Wi-Fi Direct Group.

7.2 Push communication based on pData

As we know, NDN data packets are processed if they match any PIT entry. Otherwise, the data packet is dropped. That procedure is shown on the figure right below.

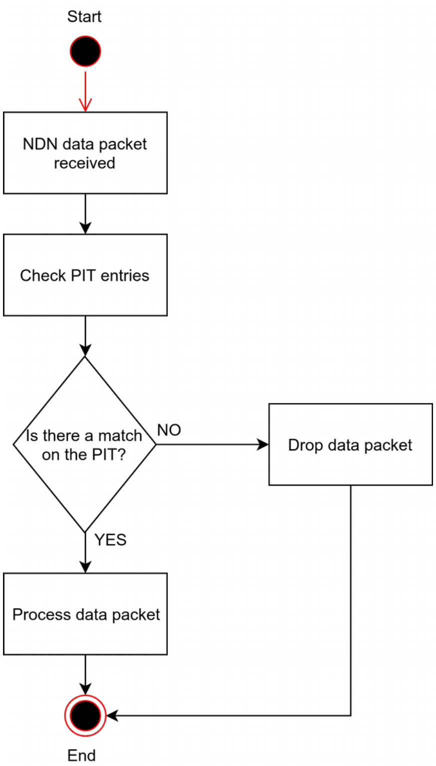

Figure 18 – NDN data packet processing flowchart

The pData is a special case of a NDN data packet where the packet is processed even if there is none PIT entry that could be satisfied. In order to do that, we performed a minor change on the class that abstracts NDN data packets. This change includes a new boolean attribute named isPushed and if is set to true, in case that there is no match on the PIT the data packet will be processed. Hence, if this attribute is set to false, this data packet is processed as a regular data packet. Our solution is illustrated on the Figure 19.

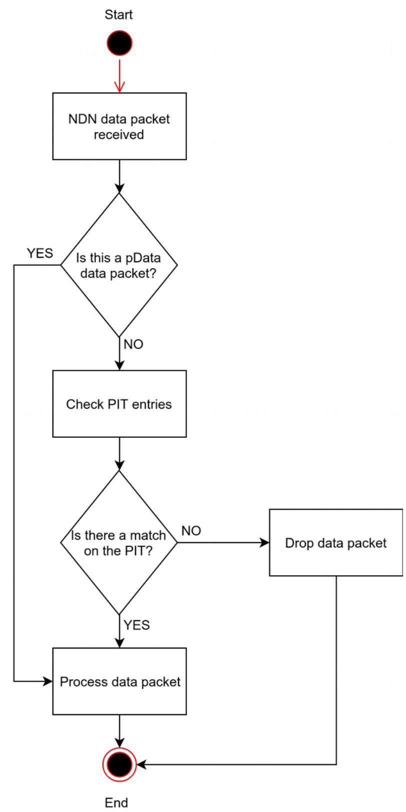

Figure 19 – pData data packet processing flowchart

8 Improvement of Network Service Discovery

This section provides information about a set of improvements that were included in the operation of Wi-Fi Direct during the development and evaluation of NDN-OPP. Such improvements are related to the operation of the Network Service Discovery (NSD) mechanism.

Network Service Discovery (NSD) gives access to services that other devices provide on a local network. Devices that support NSD include printers, webcams, HTTPS servers, and other mobile devices. NSD implements the DNS-based Service Discovery (DNS-SD) mechanism, which allows services to be requested by specifying their type and the name of a device instance that provides the desired service. DNS-SD is supported both on Android and on other mobile platforms.

NSD was also implemented on NDN-OPP, were it is responsible for detecting other devices that are using NDN-OPP via a Wi-Fi Direct network.

After a set of tests, the DNS-SD library revealed some flaws: it was noticed that in some old versions of Android, sometimes devices could not get registered. This means that such devices could not be discovered. Moreover, registration and discovering processes revealed to be too slow. For that reason, NSD service should be running all the time, not only to detect new devices but also device disconnections. Once NDN-OPP deals with opportunistic communications, it should be capable of performing such processes quickly.

Hence, in order to solve these issues, we developed a NSD similar implementation, based on the following guidelines: since a device joins a Wi-Fi Direct network, that device already knows the group owner's IP address. Then, we use this information to build a solution based on sockets, which has higher performance. Our solution implementation has four main components. All of them are explained in the next sub-sections.

8.1 Peer Registration Service

The registration service, illustrated in Figure 20, allows devices in a Wi-Fi group to discover NDN-OPP peers by sharing information via the group owner. When a Wi-Fi Direct connection is performed successfully, the device that performed this connection only knows the group owner's IP address. In order to be discovered, this device must advertise that it already joined the network. In order to do that, the device should make its IP address and UUID available in the group. These data is encapsulated in an NsdInfo object that is serialized and then sent over a socket to the group owner: the register service remains sending this object in configured time intervals.

If the Wi-Fi Direct connection goes down, the mechanism that sends these objects stops. Then, eventually the Disconnect Detector Service classifies this device as a disconnected device.

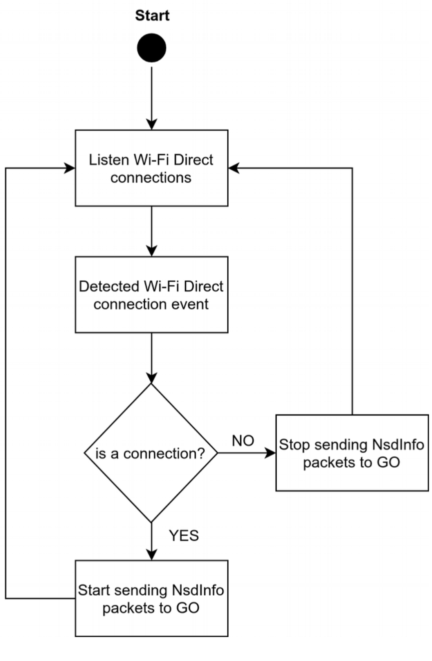

Figure 20 – Register Service Flowchart

8.2 Peer Announcement Service

This component is responsible to guarantee communications among all connected peers. In order to do that, the Discovery Service uses a socket system. As illustrated in Figure 21, when a device tries to register itself, it starts by sending, to the group owner, a NSD packet containing its personal information. The Announcement Service, which runs on the group owner, receives this packet through a socket and will notify all registered listeners.

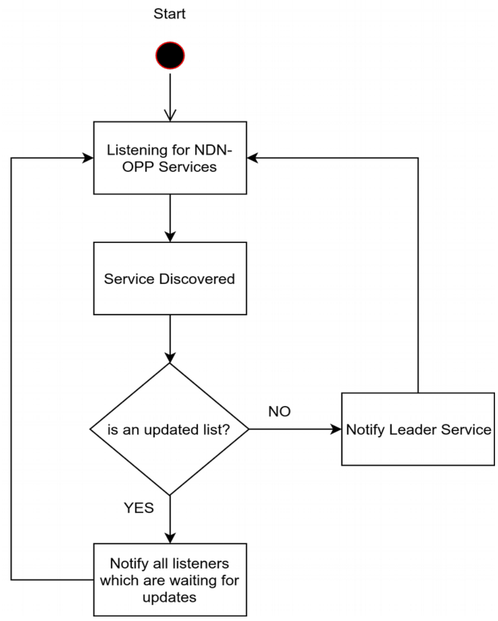

Figure 21 – Announcement Service Flowchart

8.3 Leader Service

This service is instantiated only by the group owner. The group owner is responsible to keep the list of connected devices consistent and updated. In order to do that, when a device joins a network and registers itself, the group owner will be notified. The Leader Service will receive the UUID and IP address of the registered device and then, if that device is not already in the list, the Leader Service will add it, and also notify all connected devices in order to keep them updated.

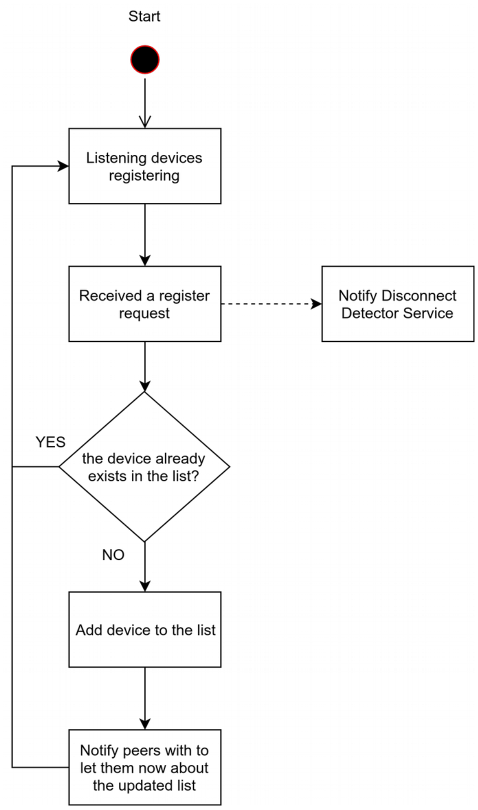

Figure 22 – Leader Service Flowchart

The periodical registration requests are sent by the Register Service in order to inform the Leader Service that this device is still alive. Also a copy of these requests is sent to Disconnect Detector Service that decides when a device is considered disconnected.

When a device is considered disconnected, the Disconnect Detector Service notifies the Leader Service saying that this device is considered disconnected. At that moment, the Leader Service removes that device from the list of connected devices, and notifies all connected devices.

8.4 Disconnect Detector Service

Since the group owner does not know when a device leaves the network, we developed an additional component to deal with it. The Disconnect Detector Service is responsible to define when a device is considered disconnected from the network.

The Disconnect Detector Service runs periodically, incrementing a counter per each device. This mechanism is represented in a flowchart form on figure 23.

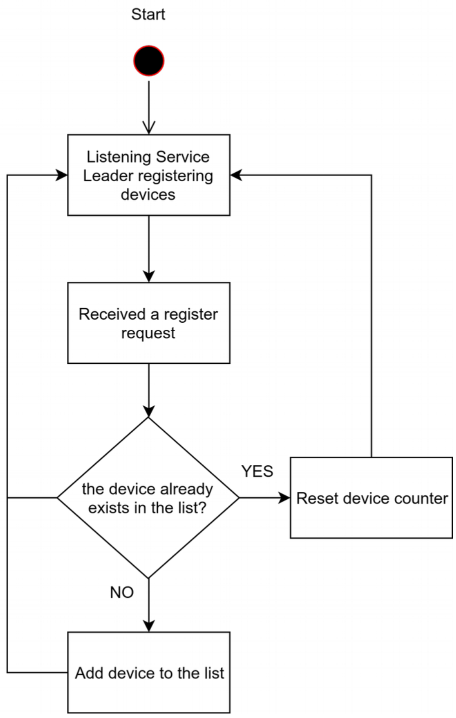

Figure 23 – Disconnect Detector Service

When this counter achieves a preconfigured number, that device is considered disconnected. Then, the Disconnect Detector Service notifies the Leader Service that such device is disconnected. This notification is performed through onPeerLost method.

The reset of this counter is performed every time that the Leader Service receives a register request from that device. This mechanism is explained on figure 24.

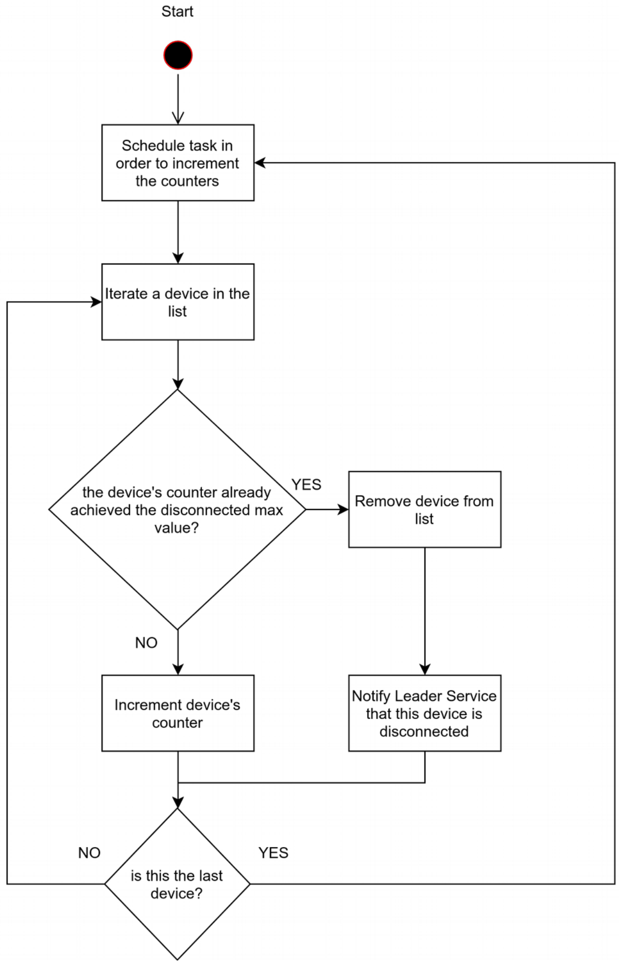

Figure 24 – Disconnect Detector Service Task

9 Ongoing Work

The current implementation of NDN-OPP provides several functionalities that are required for the proper operation of NDN over dynamic wireless networks.

The performance of the current implementation is being evaluated in three phases:

- Operation among a large set of opportunistically deployed devices making usage only of direct wireless communications. Tests are being conducted with two applications Now@ (data sharing) and Oi! (short messages) in order to evaluate:
    - The operation of the two communication methods: connection-oriented (Now@) [8, 10] and connection-less (Oi!) [9, 11].
    - The operation of the push communication models, in which case Oi! uses pData to send local alerts to all available NDN-OPP devices using Oi!, and LLI is used to allow the communication with specific authorities.
- Operation on a multi-homing environment with NDN-OPP devices deployed in two different wireless networks, both connected to the NDN global test-bed.
- Operation with DABBER aiming to avoid flooding the wireless network with Interest packets.

10 Bibliography

[1] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, M. Waehlisch, "Information-Centric Networking (ICN) Research Challenges", IETF RFC7927, Jul 2017.

[2] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, K.C. Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang, "Named Data Networking" NDN, Technical Report NDN-0019, April 2014.

[3] Paulo Mendes et al, "Push Communications in Named-Data Networking", IETF draft-mendes-pushmodel-icnrg-00, Jan 2018. – to be released

[4] Paulo Mendes, Rute Sofia, Vassilis Tsaoussidis, Sotiris Diamantopoulos, Alexandros Sarros, "Information-centric Routing for Opportunistic Networking Environments", IETF Draft, ICNRG, draft-mendes-icnrg-dabber-00 (work in progress), Feb 2018.

[5] UMOBILE Project, "D3.4 "UMOBILE ICN layer abstraction final specification", July 2017

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

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

[8] Omar Aponte, Paulo Mendes, "Now@: Data sharing in Opportunistic Wireless Named-Data Networks", COPELABS Technical Report, COPE-SITI-TR-18-02, Jan 2018.

[9] Omar Aponte, Paulo Mendes, "Oi!: Short Messaging in Opportunistic Wireless Named-Data Networks", COPELABS Technical Report, COPE-SITI-TR-18-03, Jan 2018.

[10] Omar Aponte, Paulo Mendes, "Now@ - Content Sharing Application over NDN", in ACM ICN, Berlin, Germany, September 2017

[11] Luis Amaral Lopes, Rute C. Sofia, Paulo Mendes, Waldir Moreira, "Oi! - Opportunistic Data Transmission Based on Wi-Fi Direct" in Proc. of IEEE INFOCOM, San Francisco, USA, April 2016

[12] Waldir Moreira, Ronedo Ferreira, Douglas Cirqueira, Paulo Mendes and Eduardo Cerqueira "SocialDTN: A DTN implementation for Digital and Social Inclusion", in Proc. of ACM Mobicom workshop on Lowest cost denominator networking for universal access, Miami, USA, September 2013