Oi!
Short Messaging in Opportunistic Wireless Named-Data Networks

(Version 1.0)

Technical Report COPE-SITI-TR-18-03
January 31st, 2018

Editor
Omar Aponte (COPELABS/ULHT)

Authors
Omar Aponte (COPELABS/ULHT)
Paulo Mendes (COPELABS/ULHT)

Executive Summary

This technical report describes the architecture of Oi! [6], a novel short messaging Android application for the Named-Data Networking (NDN) framework. Oi! can be used on top of NDN Android, and also on top of NDN-OPP [1,4,5], an extension of the Named-Data Networking (NDN) framework for opportunistic networking scenarios.

Oi! code is open and is available as source code on https://github.com/COPELABS-SITI/ and its APK can be downloaded on Google play[1]. The available Oi! APK works on top of NDN-OPP, allowing Oi! to be used even in the presence of intermittent wireless connectivity.

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.

[1] https://play.google.com/store/apps/details?id=com.copelabs.android.oi

Table of Contents

Executive Summary 2
Table of Contents 3
List of Figures 4
1 Introduction 5
2 General Description 5
3 System Components 6
3.1 User Interface Module 6
3.2 End User Interface 6
3.3 Authority User Interface 8
3.4 Name module 8
3.5 Data Manager Module 9
3.6 JNDN Library 10
3.7 Data Base Manager 11
4 Communication System 11
4.1 Pull Communication Model 11
4.2 Push Communication based on Special Data Packets 13
4.3 Push Communication based on Special Interest Packets 13
5 Ongoing Work 14
6 References 14

List of Figures

Figure 1 – General Component and their interfaces 6
Figure 2 – Main activity whit chats available. 7
Figure 3 – User configuration activity 7
Figure 4 – Emergency Message Activity 7
Figure 5 – Conversation Activity. 7
Figure 6 – Authority Selection Activity 8
Figure 7 – Authority Activity 8
Figure 8 – Flow chart Name Manager 9
Figure 9 – Flow chart Content Manager 10
Figure 10 – Data Base Structure 11
Figure 11 – Communication System using typical NDN interest and data packet. 12
Figure 12 – Communication System using Push Data 13
Figure 13 – Communication System using LLI 14

1 Introduction

Since the communication paradigm Named-Data Networking (NDN) [2] was proposed, a lot of research work has been carried out with the intention of providing new functions and ideas to exploit the capabilities of such infrastructure. At the same time, we have been watching to an increase of the popularity of applications such as file sharing and group text messaging.

Although, continuous research has helped to create a more stable NDN network, it is important to create applications that bring value and usability from the perspective of the end user. On another hand, people want to share content more often, for that reason we consider that NDN needs similar applications to illustrate its potential.

In this technical report Oi! [6] is described, its components and operational flow chart are shown. With this application we attend to show the push model and Long Live Interest(LLI) capabilities exiting in NDN-OPP [1,4,5] allowing data to be exchanged even in the presence of intermittent connectivity.

In the context of Oi! two different types of user are considered, End Used and Authority User. The idea of this is to show the capabilities of Oi! in different scenarios. In that sense, an End User is going to be able to stablish chat sessions in other to exchange messages with others End Users, as well as, to send message to Authority Users, for instance in an emergency scenario after occur an accident and End User could send emergency messages to the Authorities.

In another hand, in the actual version of Oi! an Authority User is going to be able to receive messages only from End Users. The intention of this is to demonstrate the benefits of use LLI and Push Data. These mechanisms are explained in NDN-Opp technical report [1].

2 General Description

Oi! is an open-source application developed for Android that allows users to exchange short text message over an NDN infrastructure. Usually applications that work on top on NDN infrastructure use the pull communication paradigm in order to exchange data. However as shown by Van Jacobson et al [7], it is possible to develop push communication applications on top of NDN, namely to establish a voice conversation over content-centric network. In the same way we aim to implement a mechanism to exchange data in Oi!, which makes use of the push communication model implemented in NDN-Opp.

With Oi!, users can maintain conversation sessions with different users at the same time. Since Oi! was developed on top of NDN-OPP, Oi! is able of exchanging messages even in condition with intermittent wireless connectivity.

Oi! is specified based on a modular approach, so it is possible to add new components and allow integration with the NDN Forwarding Daemon (NFD) [2] which is a network forwarder that implements NDN: however, the operation of Oi! on top of NFD is only possible if NDN Android implements a push communication model, as done by NDN-OPP. The modular architecture of Oi! is illustrated in Figure 1.

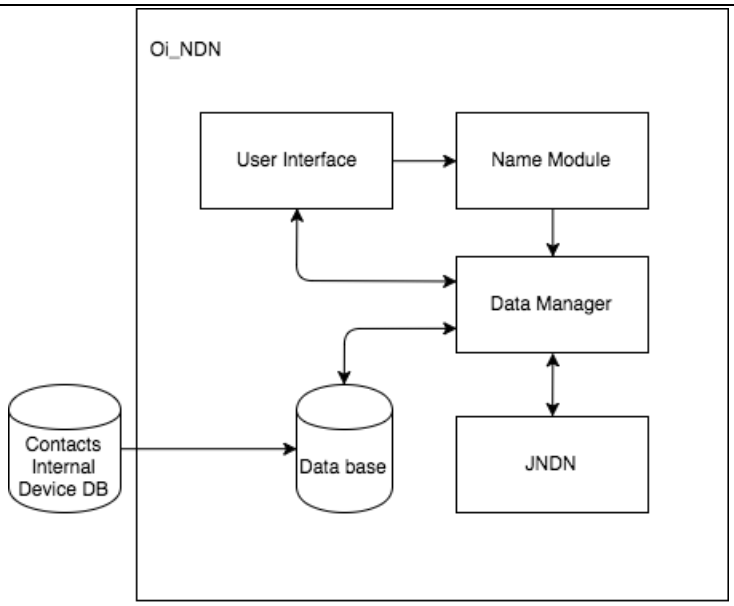

Figure 1 – General Component and their interfaces

As it is illustrated in the figure, Oi! has five mains components: User Interface; Name Module; Data manager Manger; Data base and the JNDN Library.

3 System Components

This sections shown the description of each component presents in Oi!. Besides, it illustrates diagrams and flowcharts that help to understand how the application is structured.

3.1 User Interface Module

Since there are two types of users that could use the application, two different interfaces were created. In this section End user Interface and Authority interface are explained.

3.2 End User Interface

This module is integrated by Android activity components that allows users navigate through the application. The main activity is shown in the figure 2, which provides the user with information about the conversations that are available. From here is possible to add more chats, change the user configuration and send emergency message.

When a user wants to change its profile information, the activity shown in the figure 3 is displayed, allowing the modification of the user name and the phone number information. On other hand, from the main activity is possible to open the Emergency Message activity, illustrated in figure 4, where users can send message that will be received by the users that are running the application in authority mode.

In order to display the content of a conversation, figure 5 shows the activity where the messages are displayed and where users can create new content and share it. Each message sent has a status signal that is displayed with the intention of show if it was sent or are still in the device.

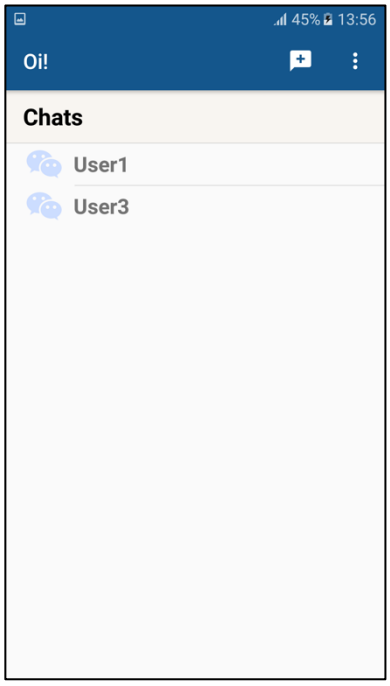

Figure 2 – Main activity whit chats available.

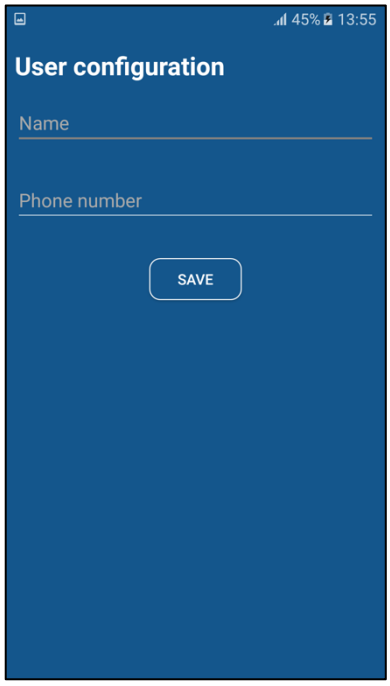

Figure 3 – User configuration activity

Figure 4 – Emergency Message Activity

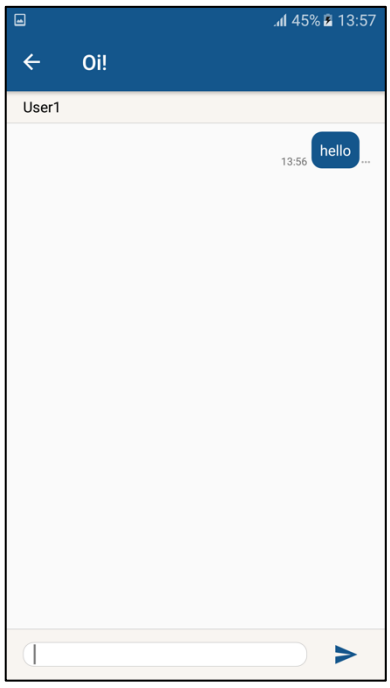

Figure 5 – Conversation Activity.

3.3 Authority User Interface

When users play the role of an Authority, first of all, they need to specify the type of authority that they represent, which will be reflected in the name prefix used by Oi!. The activity shown in the figure 6 allows to select the authority institution. After this, an activity is started and the device is able to receive information related to the selection that was done, figure 7.

As extra functionality, in this last activity is possible to select from the Message Setting button, the option that allows users to receive information from people nearby, that means, the device will be able to receive push data. More information about push data is presented in section 4.1.

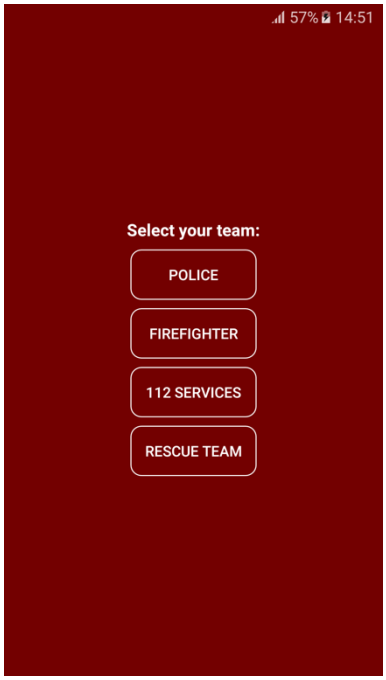

Figure 6 – Authority Selection Activity

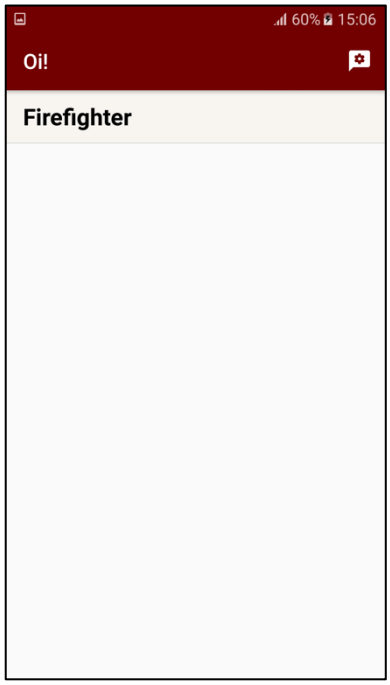

Figure 7 – Authority Activity

3.4 Name module

The Name module is used to define the name prefixes that will be used by Oi!. The main functionality of this module is to take the name prefixes associated with the conversation and append them with additional information (e.g. application name), resulting in the name prefix for the specific conversation.

In order to share data using Oi! application an agreement in terms of interest name have been done. For that reason, is used a type of name for regular chats and other for emergency messages. An example of it used in a conversation is the following:

/ndn/oi/47f0813d44bc14203be7c854fa1d22cb/0

In this example, we can consider a conversation between two devices, one of them with the identifier (phone number) 1234567, and the second one with the identifier 9876543. In this name prefix the two first parameters indicate the network and application. The third parameter represents a MD5 hash resulting of a combination of the two identifiers (1234567 + 9876543). And the last parameter indicates the number of the message. An important characteristic is that we use this name as identifier of the messages.

In another hand, the Name module has a specific name prefix that is register to NDN when the application starts. This action allows Oi! to send emergency message and also receive it from other users. The following example illustrated how is integrated an Emergency prefix:

/ndn/oi/emergency/police

When Oi! starts, the Name module provides the information of the prefixes that are going to be used by the application. After this, the module remains idle until the user selects a contact from the User Interface, which results in the creation of the name prefix of a conversation by the Name module. Figure 8 illustrates this process.

For emergency scenarios, the Name module has a specific name prefix that is register to NDN when the application starts. This action allows Oi! to send emergency message and also receive it from other users.

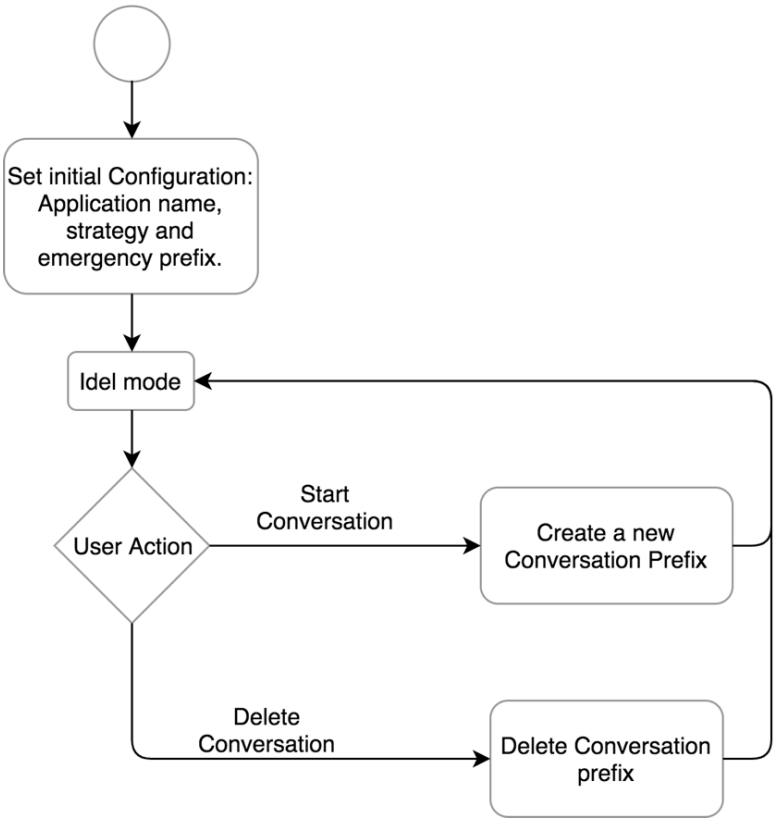

Figure 8 – Flow chart Name Manager

Note: In the actual version of Oi! application, delete conversation functionality is ongoing work. To do that, it is being implemented the option of remove the prefixes, related to the conversation to be delete, that were expressed and are still in the PIT.

3.5 Data Manager Module

The Data manager module has the function of handle the messages that are being shared. Such messages are saved in an SQLite database in the device, and sent to the user interface module with the intention to be displayed. To do this, the data manager module implements interfaces that allow it to communicate with the JNDN library.

On the other hand, when a user sends a message, Data manager receives the information from the User Interface Module, stores it in the local database and sends it to the JNDN library. Figure 9 illustrates the process that is executed in the module.

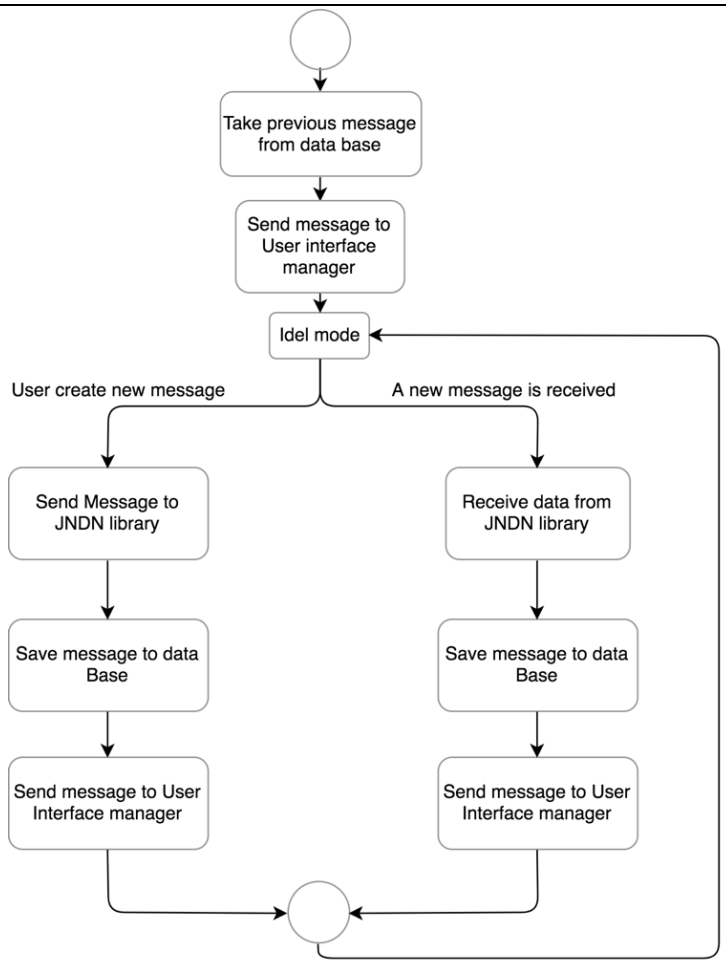

Figure 9 – Flow chart Content Manager

Messages sent to the network have a JSON structure, encompassing information about the conversation identifier, content and status. The following example illustrates a typical JSON object:

```
message= {
        “content: ” hi,
        “time:” 12:00,
        “id:” /ndn/oi/47f0813d44bc14203be7c854fa1d22cb/0,
        “c_i:” 47f0813d44bc14203be7c854fa1d22cb
        }
```

When the application is initialized, regardless of the selected user mode, the data manager registers emergency prefixes, as shown in the previous section, in order to hold messages that match stored prefixes, even when those message are not going to be displayed to the user. This action has the intention of allow the opportunistic communication implemented in NDN-OPP [1].

3.6 JNDN Library

JNDN library is a Java client library that allows Oi! to use NDN. It is used to create the faces and register the prefixes. In order to allow Oi! to use the push communication methods implemented by NDN-OPP, the JNDN library was extended to include new primitives to create Push Data packets (pData) and Long Live Interest (LLI). More information about the new push communication methods can be found in the NDN-Opp technical report [1].

It is important to highlight that the implementation of push data mechanism and Long Live Interest only works on top of NDN-OPP, since the regular NDN implementation for Android does not encompass such functionality.

3.7 Data Base Manager

Data base manager has the function of save all the information that is manage by the application: this module stores users' information, conversations and messages. This manager uses an SQLite class encompassing two tables: Messages and Conversation. Each of those tables have internal fields that allow the application works properly in order to maintain all data that is exchanged between the users.

All the information that is saved in the database is going to be used to control the messages that are involved in the conversation. The following figure 10 shows all the filed that are part of the tables.

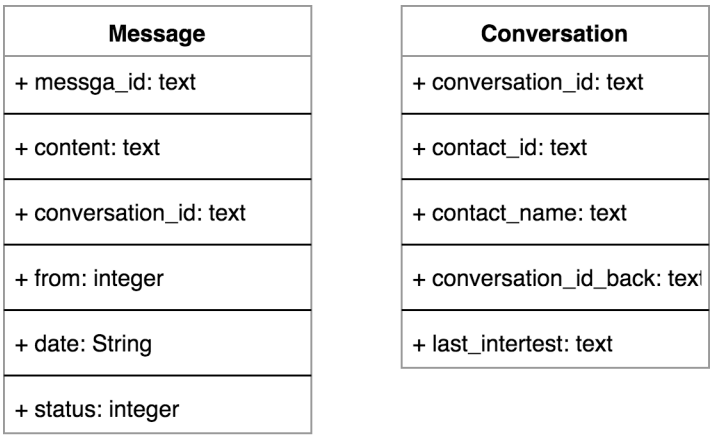

Figure 10 – Data Base Structure

4 Communication System

Oi! uses NDN-OPP as a transport system, meaning that all the information that is shared within a chat goes to NDN-OPP. For this extend Oi! makes use of the pull communication model made available by the basic NDN specification, as well as two extra mechanisms implemented in NDN-OPP to support push communications: Push Data, and Long Live Interest.

These three communication models are used in order to satisfy different scenario, which are explained in the next sub-sections. In general the pull communication model is used to allow users to chat, independently of their location, and independently of the used NDN Android implementation: users do not need to be co-located in the same wireless network; Oi! can be used on top of NDN and NDN-OPP in the same conversation.

The new push communication models are used to allow the exchange of emergency messages within the same wireless network (e.g. message will not cross a wired core NDN network). In this case Oi! makes use of two different push communication models:

- Push communication based on special data packets (pData), used to send emergency messages to any other Oi! user within a close topological distance (e.g. a citizen requesting help from any nearby person).
- Push communication based on a special Interest packet (LLI - Long Lived Interest), used to send emergence messages to specific authorities within the same wireless network, independent of the topology distance.

4.1 Pull Communication Model

As mentioned Oi! makes use of the normal pull communication model of NDN to allow users to chat, independently of their location, and independently of the used NDN Android implementation. The basic reasoning is: to start a conversation a user A must show intention to receiver data from another user B. The chat is established when user B also shows intention to receiver data from user A. This procedure is illustrated in Figure 11.

This procedure is done based on the regular exchange of interest and data packets in NDN, with a specific configuration: in order to reduce the number of transmitted Interest packets when no data is generated, Oi! sends Interest packets with a time-out longer than the default value of 4000 ms. This means that in a wireless network, PIT entries will be available for longer times.

As shown in Figure 11, a user A starts a conversation by selecting the recipient (user B) of the communication (from its local contact list – users are identified by their phone number). At this point, Oi! sends an Interest indicating that user A is interested in receiving data from user B, by using the name prefix /ndn/oi/userAUserB/0, where UserAUserB represents a MD5 hash resulting of a combination of the two identifiers, as mentioned before. User B follows a similar procedure sending an Interest packet with name prefix /ndn/oi/UserBUserA/0, as soon as it receives an Interest from user A of the form /ndn/oi/userAUserB/0 (the first data request, with parameter /0, is used to setup the chat).

After this setup phase, all messages created by user A are saved in the internal database of Oi!. Such messages are sent to user B every time Oi! receives and Interest packet from user B with the name prefix /ndn/oi/UserBUserA/x, where x is the parameter that identifies the message that user B is interested in receiving.

When data manager receives the first interest packet of a conversation, a notification is displayed in order to allow users start a new session chat. In other hand, every time that a data packet arrives to Oi!, the data manager generates an Interest packet with the number of the next data packet.

In order to make the user aware about the status of the chat, Oi! notifies the user when a data packet was sent and when a data packet was received by the recipient. The latter is done based on an **acknowledgment process** based on received Interest packets: the reception of an Interest of the form /ndn/oi/UserBUserA/x provides User A with an acknowledgement that User B got data packet x-1.

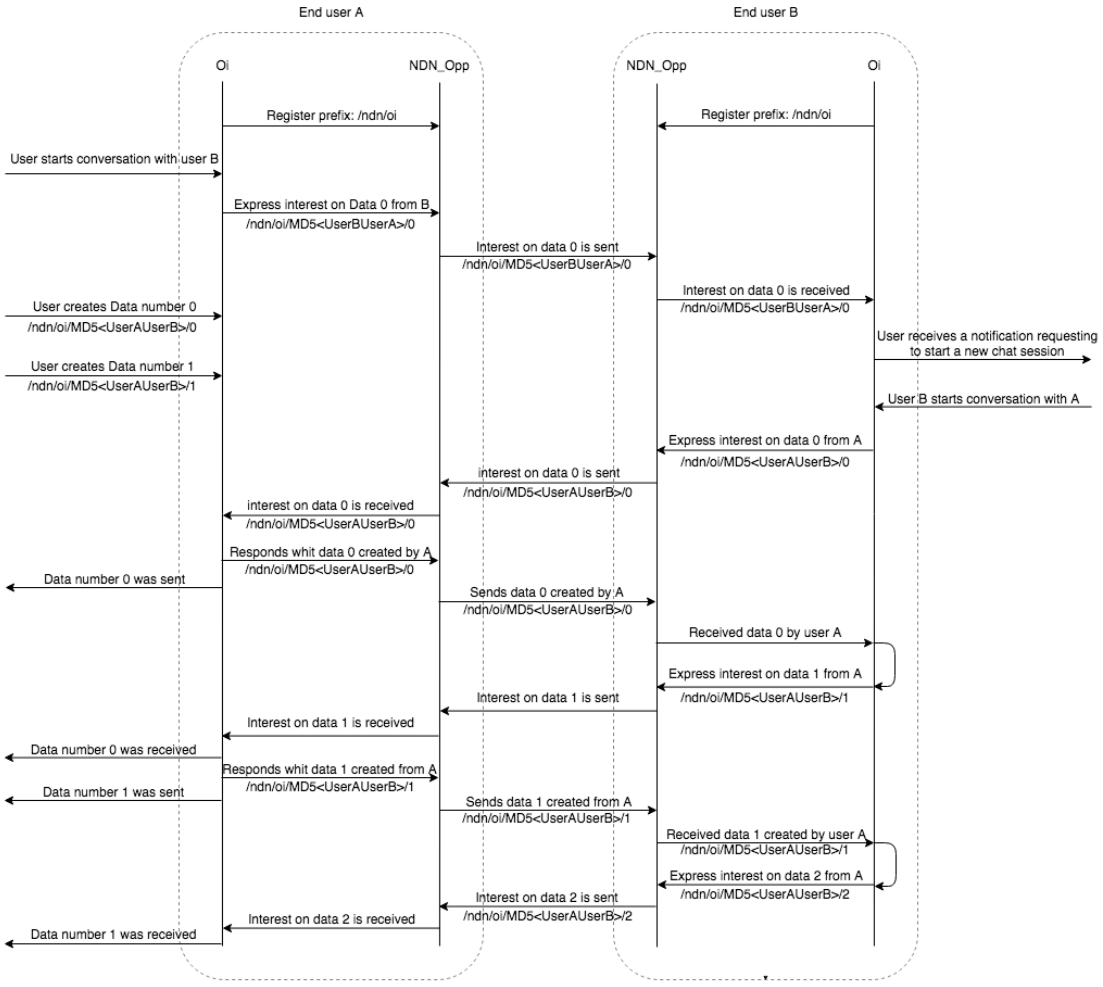

Figure 11 – Communication System using typical NDN interest and data packet.

4.2 Push Communication based on Special Data Packets

Push communication based on special data packets, called pData, is used to send emergency messages to any other Oi! user within a close topological distance (e.g. a citizen requesting help from any nearby person). pData are special data packets implemented by NDN-OPP to allow emergency data to be pushed to any nearby user without the need to have these users sending Interest packets. In general terms, when a pData is received, NDN-OPP does not check the local content store neither the PIT, which is the regular flow of a packet data. Instead NDN-OPP sends the pData directly to the next-hop based on the information stored in the FIB, if the pData counter is higher than zero **(this counter is decreased by NDN-OPP before dispatching the packet).** Further explanation of pData implementation is provided in NDN-OPP technical report [1].

In the context Oi!, this communication mechanism is used in emergency scenarios where devices cannot wait to receive an interest packet in order to be able to send a data packet. For this propose, with Oi! a User creates an emergency message and selects the option of "People Nearby" from the Emergency Activity. This message is sent to NDN-OPP, via JNDN, as a **pData, with a counter of XX.**

The following diagram illustrates in a high level abstraction how this process is executed.

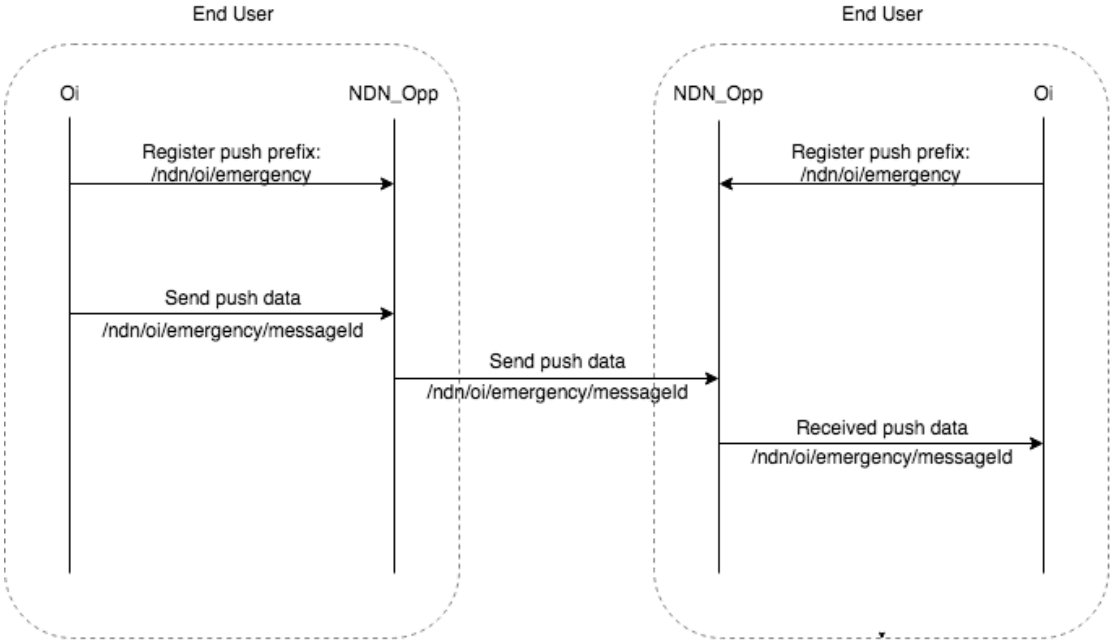

Figure 12 – Communication System using Push Data

4.3 Push Communication based on Special Interest Packets

Oi! uses a push communication based on a special Interest packet (LLI - Long Lived Interest) to allow users to send emergence messages to specific authorities within the same wireless network, independent of the topology distance. LLI are special interest packets implemented by NDN-OPP to allow emergency data to be pushed to a well-identified authority that is willing to receive emergency messages.

Authorities use Oi! to express their interest in receiving emergency messages for a certain amount of time (e.g. 3 days). Oi! passes this interest to NDN-OPP by creating an LLI via JNDN. After the reception of an LLI, NDN-OPP sends it as a regular Interest packet, the difference being that data packets do not consume a LLI entry on PIT. Such entry will only be deleted after the LLI times out. Further explanation of pData implementation is provided in NDN-OPP technical report [1].

With LLI, Oi! immediately dispatches emergency message to special authorities without the need to wait for an Interest packet. On the other hand, this mechanism reduces the number of Interest packets

that an authority would produce for a certain amount of time (e.g. 3 days) if regular Interest packets would be used.

The following figure is shows an example of this process, and comparing with the typical NDN communication mode shown in the previous sub section

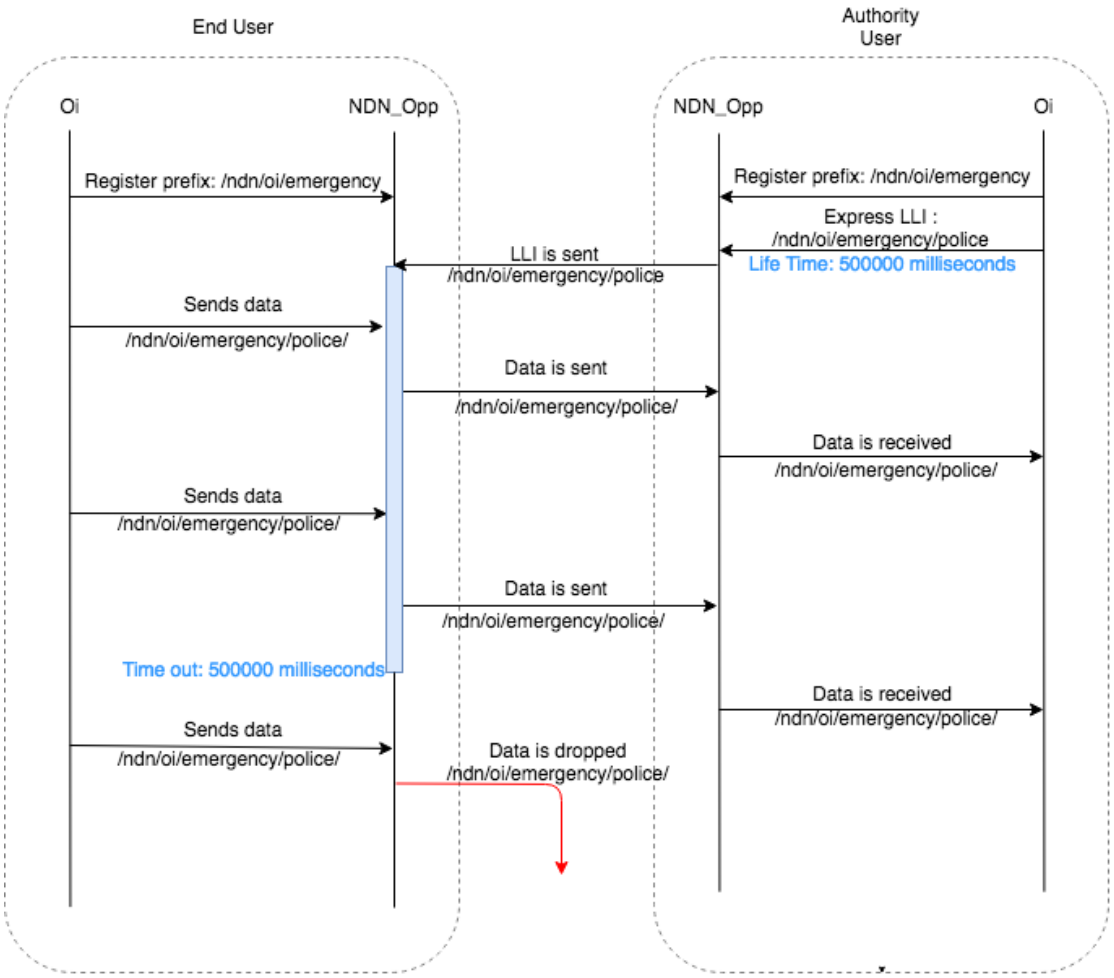

Figure 13 – Communication System using LLI

5 Ongoing Work

With the intention of improving the functionalities present in the application, we are doing validation tests of LLI's time out in different scenarios including fixed Wi-Fi networks.

6 References

[1] Miguel Tavares, Paulo Mendes, "NDN-OPP: Named-Data Networking in Opportunistic Networks", COPELABS Technical Report, COPE-SITI-TR-18-01, Jan 2018.

[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] Alexander Afanasyev et al. "NFD Developer's Guide". Technical Report NDN-0021. NDN Consortium, 2015

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

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

[6] 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

[7] Van Jacobson et al. “VoCCN: Voice Over Content-Centric networks”. In proc of Reach workshop, Rome, Italy, Dec 2009.