Action full title:

Universal, mobile-centric and opportunistic communications architecture

Action acronym:

UMOBILE

Deliverable:

D5.2 "Validation methodology and evaluation report (2)"

Project Information:

Project Full Title	Universal, mobile-centric and opportunistic communications architecture
Project Acronym	UMOBILE
Grant agreement number	645124
Call identifier	H2020-ICT-2014-1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Topic	ICT-05-2014 Smart Networks and novel Internet Architectures
Programme	EU Framework Programme for Research and Innovation HORIZON 2020
Project Coordinator	Prof. Vassilis Tsaoussidis, Athena R.C.

Deliverable Information:

This deliverable provides an update on the validation setup for the system and its individual components of the UMOBILE architecture. In addition, this report includesnew results of the evaluation of the platform through simulations.

In this deliverable we also show the updated diagram block for each of the proof-of-concepts, as part of the final demonstration that will be carried in Italy and also add a section regarding the system integration and its validation.

Deliverable Number-Title	D5.2 Validation Methodology and Evaluation Report
WP Number	WP5
WP Leader	FON
Task Leader (s)	Alberto Pineda (FON)
Authors	**COPELABS**: Paulo Mendes **ATHENA**: Sotiris Diamantopoulos, Christos-Alexandros Sarros, Vassilis Tsaoussidis **UCL**: Ioannis Psaras, Sergi Rene **UCAM**: Adisorn Lertsinsrubtavee, Carlos Molina-Jimenez **SENSECPTION:** Rute Sofia **TECNALIA**: Susana Pérez **TEKEVER**: Luis Deprez, Ricardo Faria **AFA**: Francesco Amorosa, Giammichele Russi **FON:** Alberto Pineda, Pablo Salvador
Reviewer	Alberto Pineda, Pablo Salvador
Contact	alberto.pineda@fon.com
Due date	M24: 31/01/2018

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Actual date of submission	06/02/2018

Dissemination Level:

PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

Document History:

Version	**Date**	**Description**
Version 0.1	27/12/18	Initial template with partners' input
Version 0.2	02/01/18	Circulate among partners first draft proposal and ask for extra input
Version 0.3	15/01/18	Revised first version based on feedback
Version 0.4	22/01/18	Added some partners' contributions
Version 0.5	28/01/18	Finalized preliminary version of the deliverable and circulated for internal review
Version 1.0	08/02/17	Final version after reviewers' comments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

List of Figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Executive Summary

Background

Work Package 5 "**Overall platform integration and validation"** of UMOBILE project evaluates the solutions developed in the project.This Report is written in the framework of Tasks5.1 "**Definition of the validation of the setup**" and 5.2: "**Evaluation through Simulation and Emulation**" of UMOBILE project.These two tasks aim to validate the architecture and services, derived from results developed in WP3 and WP4.

The ultimate objective of UMOBILE is to advance networking technologies and architectures towards the conception and realization of Future Internet. In particular, UMOBILE extends Internet (i) functionally – by combining ICN and DTN technologies within a new architecture -, (ii) geographically – by allowing for internetworking on demand over remote and isolated areas – and (iii) socially – by allowing low-cost access to users but also free user-to-user networking.

Objectives

This document briefly describes UMOBILE system components from a functional point of view (for further detailed please check D5.1 [1]). Furthermore, this deliverable updates the validation setup of the overall system, detailing on the integration of each of the components and the APIs exposed as well as on the use cases evaluated. This entails defining the setup and assets involved in the validation scenarios, use cases, operational and environmental conditions, measures of performance, and measures of effectiveness. There will be both component and system-level validations. The platform is also evaluated through a series of planned simulations and emulations.In this deliverable we also show the updated diagram block for each of the proof-of-concepts, as part of the final demonstration that will be carried out both in Italy and in Brussels, and also add a section regarding the system integration and its validation.

1. Introduction

The UMOBILE project aims to introduce new paradigms in the provision and consumption of the Internet connectivity by developing a mobile-centric service-oriented architecture that efficiently delivers content to the end-users. To this end, UMOBILE provides an architecture that merges information-centric networking (ICN) with delay tolerant networking (DTN) in order to efficiently operate in different networks situations, reaching disconnected areas and users and providing new types of services. This architecture allows decoupling services from their origin locations and shifting the host-centric paradigm to a new paradigm that incorporates aspects from both information-centric and opportunistic networking.

The solutions defined and prototyped in WP3 and WP4 work packages give a glimpse of the specific performance and/or functionality improvements developed within the project. In order to demonstrate the performance and efficiency of UMOBILE architecture within a testbed, we have selected some specific use-cases identified in the WP2 for the purpose of feature demonstration.

More specifically, we focus on two demonstrations: the first one covers both the emergency and the civil protection scenarios, and the second demonstration targets the service announcement and social routine scenarios, which both are presented in detail in the deliverables D5.3 [2] and D5.4 [3]. The aim of this document is to update the evaluation of the individual components of the UMOBILE platform, as well as the system integration and its validation.

This document is organized as follows: in Section 2 we describe the UMOBILE platform and their components according to functional blocks. Following, we present the simulation platform used to evaluate the different services and functionalities in Section 3. Each of the UMOBILE services and their corresponding validation by means of simulations or experimental tests is presented in Section 4. Section 5 presents the validation of the UMOBILE system platform in terms of integration among different UMOBILE components following the proof-of-concept description carried out in D5.3 [2]. Finally, the conclusions are drawn in Section 6.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

2. UMOBILE Platform

Here we briefly describe the already well-known UMOBILE platform as well as each of their elements. In addition, we outline all the elements of the platform and explain them.

Figure 1shows a high-level description of the UMOBILE platform in a specific scenario, highlighting both domains: the UMOBILE domain; this is the NDN-DTN area where the UMOBILE platform provides services to end users, and the Internet domain.

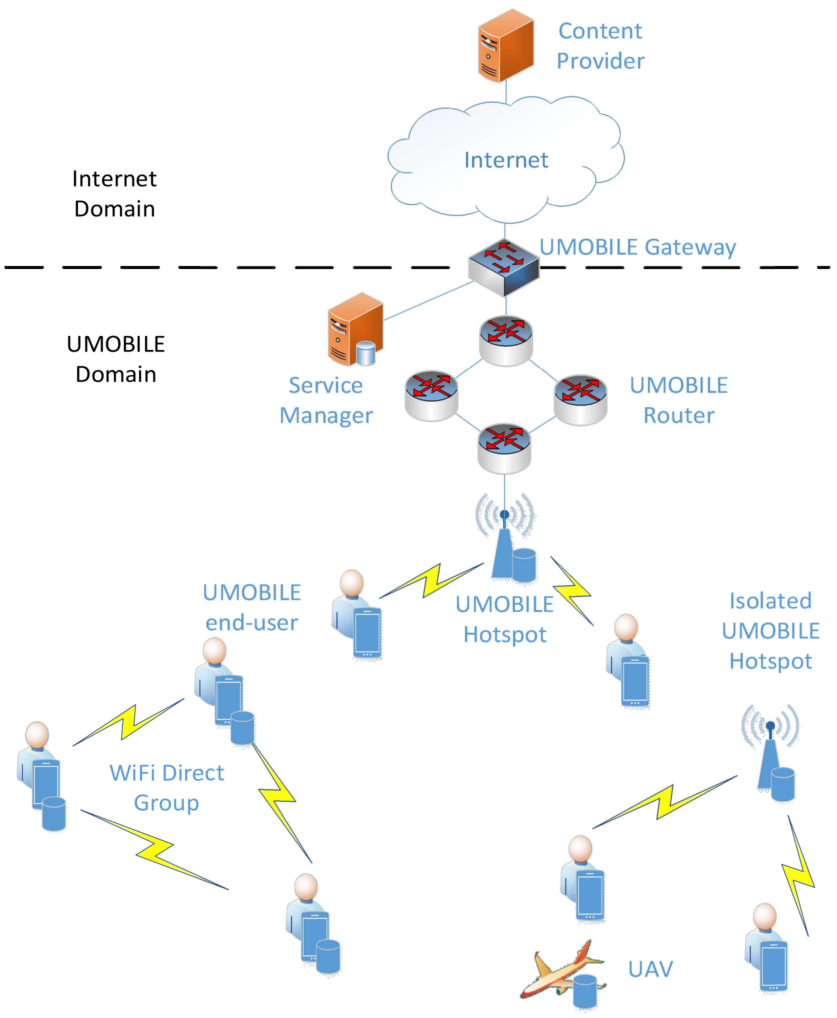


Figure 1: Overview of the UMOBILE platform

Figure 2extends the concept ofFigure 1containing different scenarios where the UMOBILE platform operates.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

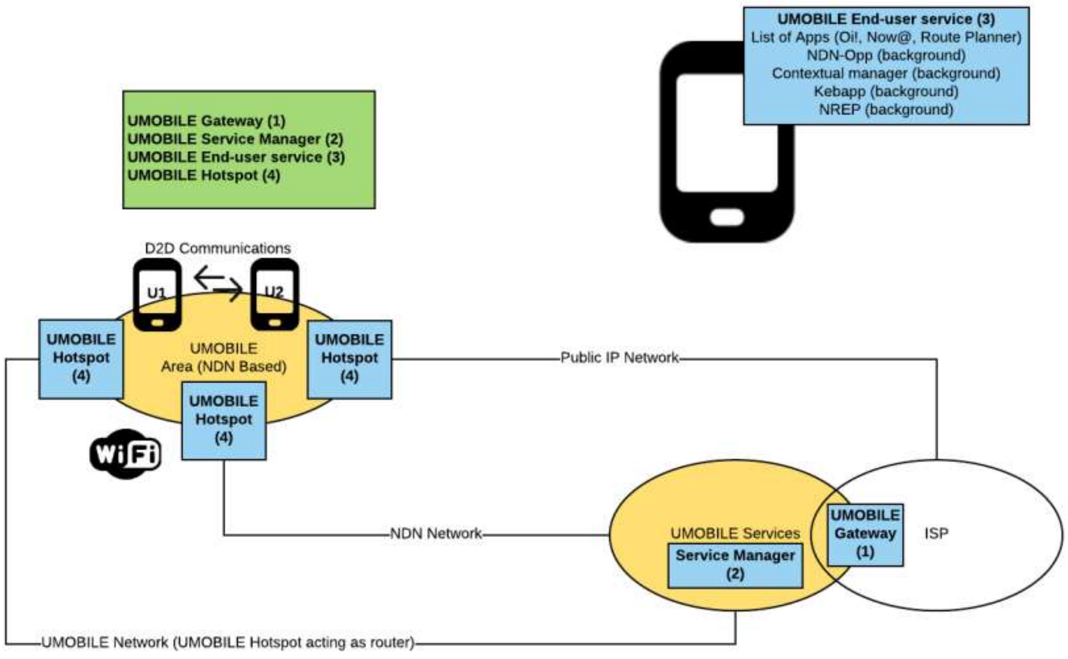


Figure 2: Functional blocks of the UMOBILE platform

The actors involved in the UMOBILE architecture include:

- **UMOBILE Hotspots**: deployed in the fixed part of the network or on-board mobile nodes (e.g., UAVs). They collect and relay relevant information, host some instantiated services or store collected data, check its validity and perform computational functions to increase the value of the information to the civil authorities.
- **UMOBILE gateway**: provide interconnectivity between UMOBILE domain and the Internet domain.
- **UMOBILE service manager**: implement the functionality that the Service Provider needs to deploy his services.
- **UMOBILE end-user service:** send and receive, as well as carry and forward data, based on an opportunistic networking approach.

Note that UMOBILE is being developed as a modular software architecture, where some modules may or may not reside in a specific hardware element. For further information regarding the actors part of the UMOBILE architecture; please check D5.1 [1].

2.2.1. UMOBILE UAVs

UMOBILE-enabled UAV devices collect and relay relevant information and connect two isolated areas, as they are a specific case of a UMOBILE hotspot. The UAVs are equipped with WiFi to create a local communication infrastructure: the UAVs are instructed to perform manoeuvres that maximize the covered area and enable communication with

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

other equipment in the area. UAVs can be used as well to collect the emergency message and forward it until it reaches the authorized emergency entity.

After a long set of experiences and tests, the final selected UAVsused to achievethe UMOBILE objectives are the AR3 from TEKEVER (over the initially supposed AR4), and alsothe VR1 (TEKEVER). Indeed, the AR4 previously selected turned out to not having the necessary payload requirements for this mission, which led to its exchange to AR3 that proved to be the appropriate vehicle for this mission regarding wide areas, with VR1 taking advantage of its quadcopter VTOL characteristics to operate in small and strictly local areas – which, for instance, can be a valuable tool for emergency cases.

The AR3 Net Ray is a fully autonomous tactical, UAV for ISTAR (Intelligence, Surveillance,Target Acquisitionand Reconnaissance) to be used in the defence market. With an endurance of more than 10hours and a range of 80km it is designed essentially for diverse activities such as search and rescue, surveillance or maritime patrol, for instance, having also been validated and battle proven by multiple security and military forces.. Being launched with the use of a catapult, it weighs about 22kg and is capable of carrying up to 5kg payload – including the Gimble camera. This represents an advantage over the also tested UAV AR4, since this last one had a payload of 1 kg also including the Gimble camera and therefore not enough to support the necessary devices for a wide area mission.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

3. UMOBILE Simulation Platform and Methodology

As the UMOBILE platform integrates a diverse set of modules and mechanisms, no single simulator is used; instead we opt for the simulators that are most relevant to the individual modules with respect to the functionalities being assessed. Relevant information concerning validation tests can be found on the appropriate sub-sections of Section 4.

As far as evaluation by simulation is considered, we leverage three specific platforms: ns3, ndnSIM and ONE. Each of these platforms focuses on different network settings and scenarios, so the modules assessed played a crucial role for the simulator being used.Network Simulator 3 (ns3) [14] is a discrete event network simulator, open-source, used to test network protocols.NdnSIM is an extension to the ns3 simulator, implementing the basic components of the Named Data Networking architecture in a modular way. [15]. The Opportunistic Networking Environment (ONE) simulator [16] is a Java-based tool for opportunistic networks, offering a broad set of DTN protocol simulation capabilities in a single framework. It was designed targeting simulation and analysis of DTN routing and application protocols.

A detailed description of the simulation tools employed by the UMOBILE project can be found in Sections 3.1 and 3.2 of Deliverable D5.1 [1].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4. UMOBILE Validation

In this section wepresent the update on the validation of the different components of the UMOBILE network devices and end-user servicescorresponding to software modules by means of experiments or simulations.For some of the components we refer to the specific section in D5.1 [1].

4.1. In-Network Resource Pooling Protocol (INRPP)

4.1.1. Description

Within task 4.1 a new congestion control protocol named In-Network Resource Pooling Protocol (INRPP) has been developed, aimed at improving congestion control in the fixed network. It uses caches as temporary custodians to deal with congestion locally, and using backpressure techniques to slow-down senders when necessary. There are two versions of the INRPP flowlet congestion control. D4.1 [5] discusses the first version of a congestion control protocol, called In-Network Resource Pooling Protocol (INRPP) that addresses congestion control in TCP/IP networks. It extensively covers the specifications for a Flowlet Congestion Control to be used in the fixed part of the network. The final version of this protocol is detailed in D4.4 [22], including the development of the Flowlet Congestion Control to be used in the NDN domain.

For more information on the IP version of INRPP please check Section 4.1 in D5.1 [1].Regarding the NDN version of the INRPP [21], we developed the INRPP congestion control for NDN routers, where we use caches not only to temporary cache content, but also to temporary deal with congestion locally and hop-by-hop, avoiding shaping interests, reducing immediately interests rates and being able to make use of unused residual bandwidth of in-network detour paths (enabling multipath forwarding). INRPP outperforms the Interest-based Congestion Protocol for NDN [22] and the recently proposed Practical Congestion protocol[23], as stated in the evaluation of Section 4.1.4. In the second version of INRPP (D4.2) we assume an NDN-enabled network, where the whole network can be NDN, or it also can be NDN nodes working as an overlay network for TCP/IP networks to allow incremental deployability.

4.1.2. Demo Key Features

This service is not planned to be implemented in real devices and deployed in the final demo. INRPP can work together with the service migration to improve QoS, as detailed in deliverable D4.4, which is going to be implemented and deployed in the final demo. In order to evaluate and demonstrate the good performance of INRPP we require a large scale network, therefore we will evaluate and validate INRPP only by simulations.

4.1.3. Hardware and Software Requirements & Experimental Settings

This software module does not impose any hardware requirement as it is being validated by means of simulations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.1.4. Validation & KPIs

We next engage in a detailed investigation of the performance of the proposed INRPP TCP/IP version [4] scheme in an infrastructure scenario by means of simulation. We compare the performance of INRPP with TCP, RCP, and MPTCP for various topologies and scenarios. We implemented INRPP in ns3 [14], ported the existing ns2 implementation of MPTCP to ns3 and used RCP's existing implementation for ns3. We use the New Reno version of TCP. We evaluated INRPP in a three different scenarios. The first one is a very simple topology (dumbbell topology), in order to show in detail, the operation of the different INRPP mechanisms detailed in Section 2.2. The second scenario is a simple multi-homed scenario where we can compare INRPP with not only single-homed transport protocols, such as TCP and RCP, but also multi-homed transport protocols, such as MPTCP transport protocol. The third scenario, is a more complex hierarchical scenario, where we can evaluate INRPP performance in a similar scenario that can be used for the service placement UMOBILE module (introduced in D3.3, Section 4.4[20]), where communications take place in the Internet domain between central servers and service instances placed in the edge of the network.

For the NDN version of INRPP, we evaluate the solution using the ndnSIM simulator [11] in the aforementioned dumbbell topology with one detour path, comparing it with ICP and PCON, also implemented in the ndnSIM (code available here[1])

INRPP for TCP/IP networks

Dumbbell topology with a detour path

We first evaluate a simple scenario using a dumbbell topology with a fully-connected core component (nodes 0, 1, and 2), depicted in Figure 3. The purpose is to show in detail the operation of the different INRPP mechanisms in a simple setup. This topology has a bottleneck link (link 0-2) with 10 Mbps capacity, but it also has another 10Mbps capacity one-hop detour path (link 0-1-2) that can be used to extend the bandwidth available at the bottleneck. Hosts have access links with 40Mbps bandwidth capacity, and links 4-0 and 2-3 have more capacity than the rest of the links. We pair the senders (three hosts on the left) and the receivers (three hosts on the right), and each sender initiates a single flow to its receiver pair. The three flows from the senders are initiated with one second gap in between. Each flow has the same size of 10 MB, and the size of each router cache is set to only 1.25 MB in order to demonstrate the activation of the backpressure mechanism, which eventually propagates to the sender. We set the packet size to 1500 bytes. Router interfaces buffer size is set to 50 ms worth of traffic using Drop Tail and link latencies are 5 ms. We simulated the scenario using INRPP, RCP and TCP. In this scenario, we do not evaluate MPTCP since no hosts are multi-homed, and therefore there is no possibility of establishing multiple sub-flows between peers.

[1]https://github.com/schneiderklaus/ndnSIM

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

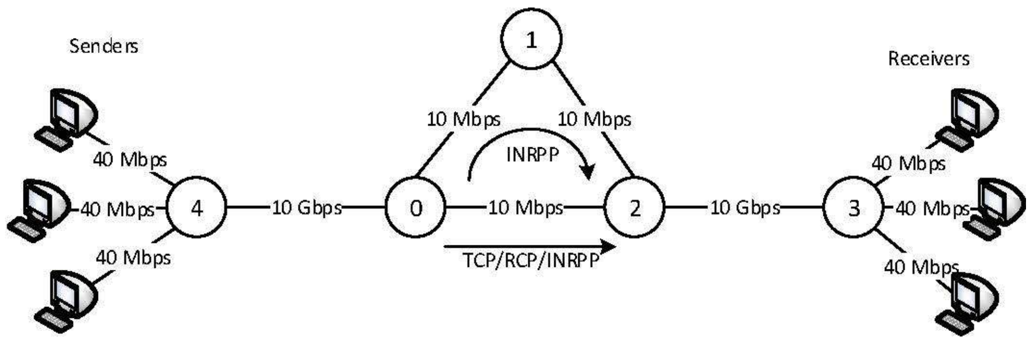


Figure 3: Detour simple scenario

In Table 1we observe the average flow completion time (AFCT). We see that INRPP complete the flows much faster than TCP and RCP, because INRPP is using all the bandwidth available in the bottleneck, plus the bandwidth available in the detour path, when TCP and RCP is only using the bottleneck link capacity. In Figure 4, we demonstrate the goodput at the receiver in bps. With INRPP (top figure), we observe that the bandwidth is shared equally between the existing flows and there is no fluctuation when a new flow arrives. Particularly, the first flow starts at second 1 and is transmitted at 20Mbps using both the bottleneck link and the detour path shown with an arrow in Figure 3. When the second flow starts at second 2, the capacity is immediately shared between the two active flows (10Mbps each), while when the third flow starts at 3 seconds the available bandwidth is immediately split between the three active flows (≈ 6.66 Mbps per flow). When flows start completing, the existing flows adapt their rate and share the bandwidth no longer used by the completed flow. With TCP (bottom figure), on the other hand, we can observe that the goodput at the receiver is erratic and fluctuates excessively. TCP shares the bandwidth equally; however, it needs time to adapt to the new flow arrivals. Even after all the flows begin, the goodput oscillates continuously throughout the simulation due to the saw-tooth behaviour of the congestion window of TCP. In contrast, RCP does not have such oscillation in goodput, but we see that RCP also requires time to adapt and share the bandwidth between flows efficiently. In this simple scenario, RCP's slow adaptation to arriving flows is more pronounced due to the small number of flows: RCP shares the bandwidth among flows by estimating the number of active flows, and when the number of active flows is small, this estimation is less accurate; that is, the error rate is higher when the estimation is, for instance, off by one. The slow adaptation of RCP to arriving flows leads to worse AFCT than TCP in this scenario.

Table 1: Dumbbell topology simulation results

Protocol	AFCT	Fairness	Fairness different RTTs
INRPP	10.50s	0.9945	0.9972
RCP	25.96s	0.9934	0.9994
TCP	19.08s	0.8902	0.8321

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

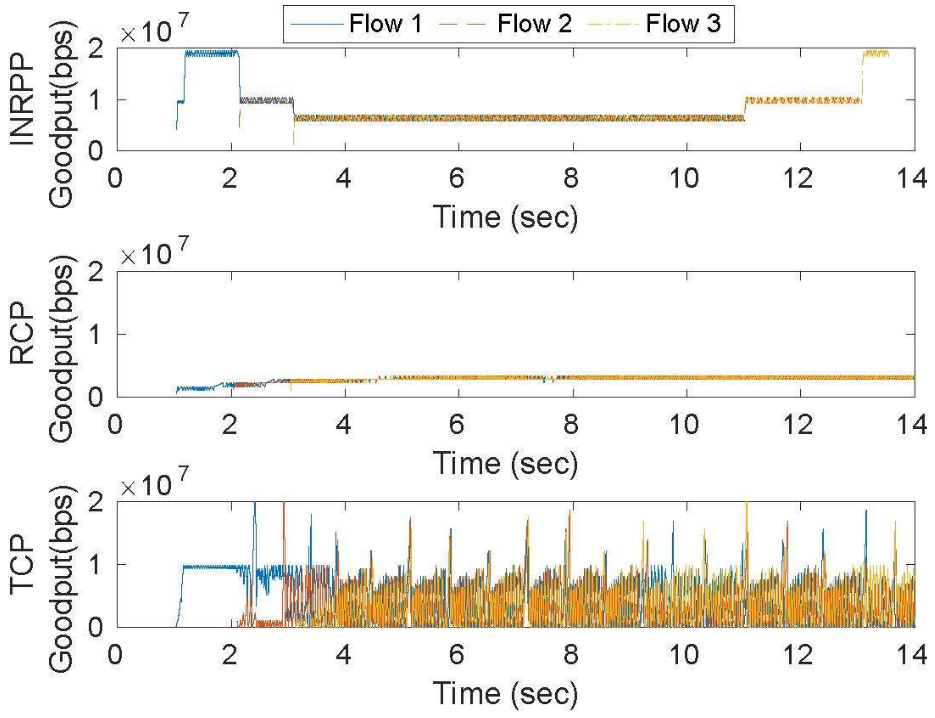


Figure 4: Receiver goodput

In Table 1 we observe the fairness during the simulation. To measure it, we first compute the Jain's index on the instantaneous throughput of flows, sampled every 10 ms throughout the simulation, and then take the average of the samples to obtain a single average fairness value, shown in the third column. In order to demonstrate that INRPP is not affected by diverse RTT path, we also evaluated fairness for heterogeneous access link latencies: 50, 100 and 200 ms. With homogeneous RTTs, we observe that INRPP and RCP fairness is close to the optimal with TCP having the worst fairness because of throughput oscillations. With heterogeneous RTTs, on the other hand, we observe that TCP fairness becomes even worse, while INRPP and RCP fairness is maintained close to 1, given their rate-based transmission pattern.

Rocketfuel topology

In this last scenario, we evaluate INRPP in a real, large-scale topology. For that purpose, we use the 3257.pop.cch network topology2 as provided through the Rocketfuel dataset. Spanning over the European continent, the 3257.pop.ch topology has V = 440 routers and 681 bidirectional links. Out of the 440 routers, 267 are edge routers (with degree less than 3), 126 are gateway routers (i.e., connected to an edge router and has degree larger than 2), and 47 backbone routers. A total of 13350 senders and receivers are connected to the edge routers and flows are established randomly in the same way than previous topologies. We randomly pair each sender with a receiver and start a flow from each sender to its pair, which makes a total of 6675 flows. In this scenario, we start these 6675 flows with an offered load of the access link $\rho = 0.8$ and an$E[L] = 125$. All the links have the same capacity of 1 Gbps except the links connecting edge

[2]https://github.com/cawka/ndnSIM-sample-topologies/blob/master/topologies/rocketfuel_maps_cch/3257.pop.cch

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

routers with gateway routers and users with edge routers, which have 100 Mbps capacity. In this scenario, we can have multiple bottlenecks and multiple detour paths in the network. Here, we also compare INRPP against RCP, TCP and MPTCP, where users are multi-homed with two of the edge routers uniformly randomly distributed.

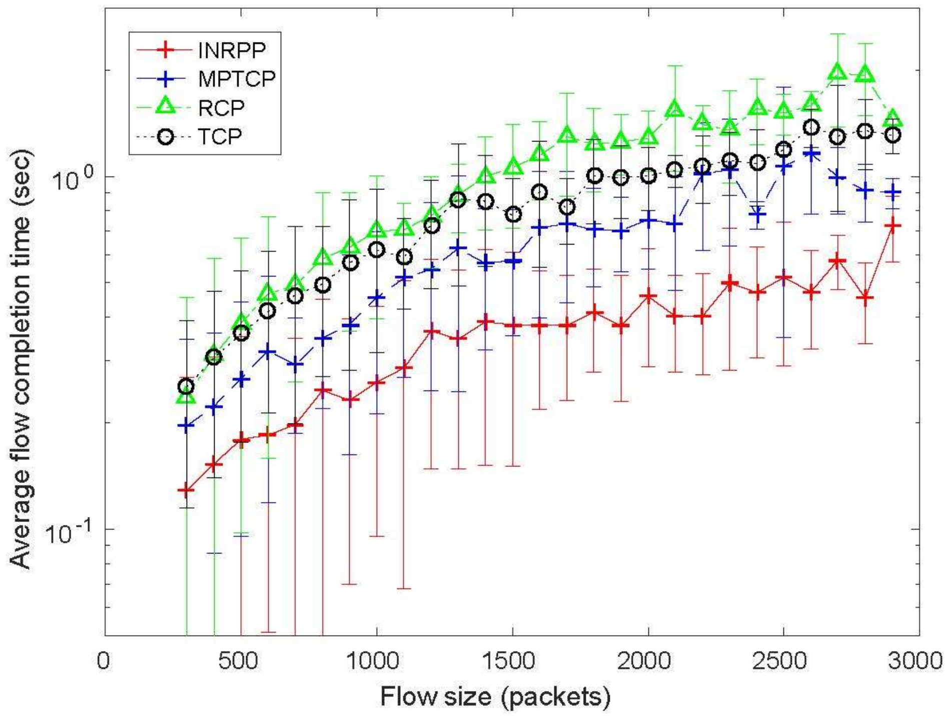


Figure 5: AFCT for the rocketfuel topology

In Figure 5, we show the AFCT for the different flow sizes, using the same PBPP process, previously described. We observe that even in this challenging scenario, INRPP clearly outperforms RCP and TCP (by at least 50%) in terms of flow completion time, but also MPTCP. It does so because INRPP can take advantage of available residual bandwidth, even when it is available for very short time-intervals, e.g., milliseconds, to detour excess traffic. INRPP can also take advantage of the in-network multiple paths because of the hop-by-hop congestion control. MPTCP is not able to take advantage of all the bandwidth available in all paths because of the e2e design limits the knowledge of the in-network paths to deal with congestion. We can conclude that dealing with congestion locally using INRPP is better than the e2e congestion control used by RCP, TCP or even MPTCP given that the topology has detour paths and nodes possess caches.

INRPP for NDN networks

To evaluate the INRPP NDN version [24], we evaluate the dumbbell topology with a fully-connected core component (nodes 0, 1, and 2), depicted in Figure 6 (same scenario evaluated in Figure 3), comparing INRPP with the Interest Control protocol [26], and the practical congestion control (PCON)[27]. ICP is a window-based control protocol similar to the original TCP control protocol. ICP is an interest-shaped control protocol that uses the shortest-path only, signalling congestion events when FIFO queues drop packets. PCON detects congestion based on the CoDel AQM (by measuring packet queuing time), then signals it towards consumers by explicitly

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

marking certain packets, so that downstream routers can divert traffic to alternative paths and consumers can reduce their Interest sending rates. However PCON is unable to use detour paths when routing costs are different. The details of the scenario (cache capacity, flows size, link bandwidths) are the same than scenario depicted in Figure 3. The purpose of this evaluation is to show the benefits of using INRPP when using all available in-network paths, improving flow throughput, and avoiding any drops and retransmissions, minimizing link queues, and therefore delay caused by congestion.

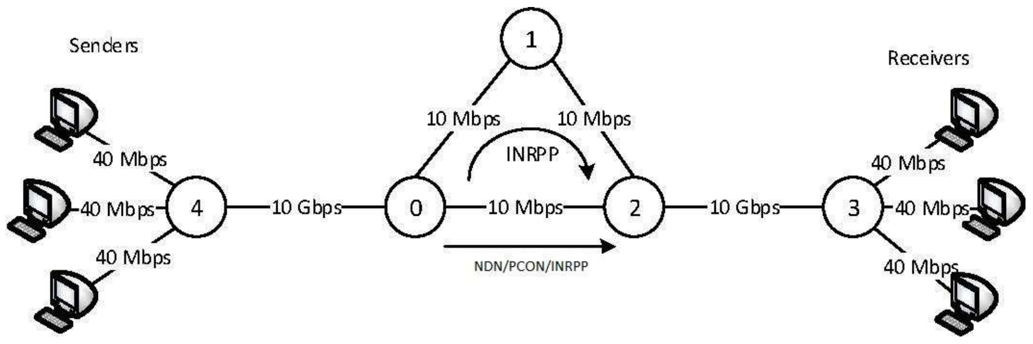


Figure 6: Dumbbell topology with detour path for INRPP NDN

In Figure 7-Figure 9, we can observe the instant throughput of the three flows existing in the simulation for the three evaluations (ICP, PCON and INRPP). In the first graph Figure 7, we can observe the instant throughput for the ICP protocol. Here we can see the ICP is the least stable protocol, caused by the retransmissions required to resend the dropped packets in the network FIFO queues. We can also observe that the average throughput is limited to the bottleneck of the link (10 Mbps) and is unable to use any unused in-network path. We can also see that, since the 3 flows are not started synchronously, the rate between controls is not stable and the fairness between flows is not perfect.

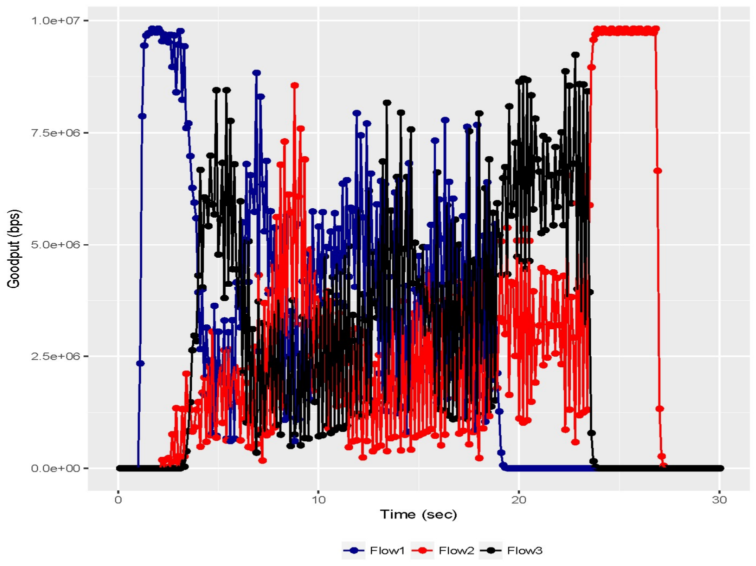


Figure 7: Goodput evaluation. Goodput using ICP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

In Figure 8, we can observe the instant throughput for the PCON protocol. Here we can see that the PCON protocol is more stable than ICP, since it does not cause network drops and the bandwidth is more equally shared between flows (better fairness). However, PCON is not able to use the unused bandwidth in the detour link (0-1-2), and the flow completion time is similar to ICP, despite being more equal between the three flows.

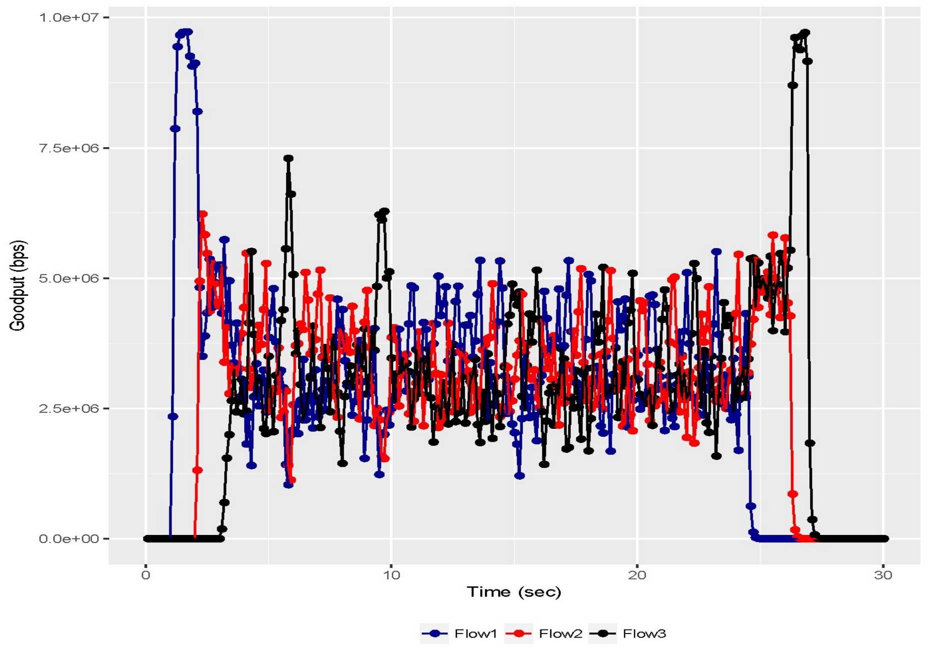


Figure 8: Goodput evaluation. Goodput time PCON

In Figure 9, we can observe the instant throughput for the INRPP protocol. Here we can see that the bandwidth is equally shared between the 3 flows (the instant throughput is exactly the same) and the throughput is stable and constant. Also, INRPP is able to use the detour path (0-1-2), increasing the instant throughput (20Mbps shared between the 3 flows) and reducing the flow completion time close to the half.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

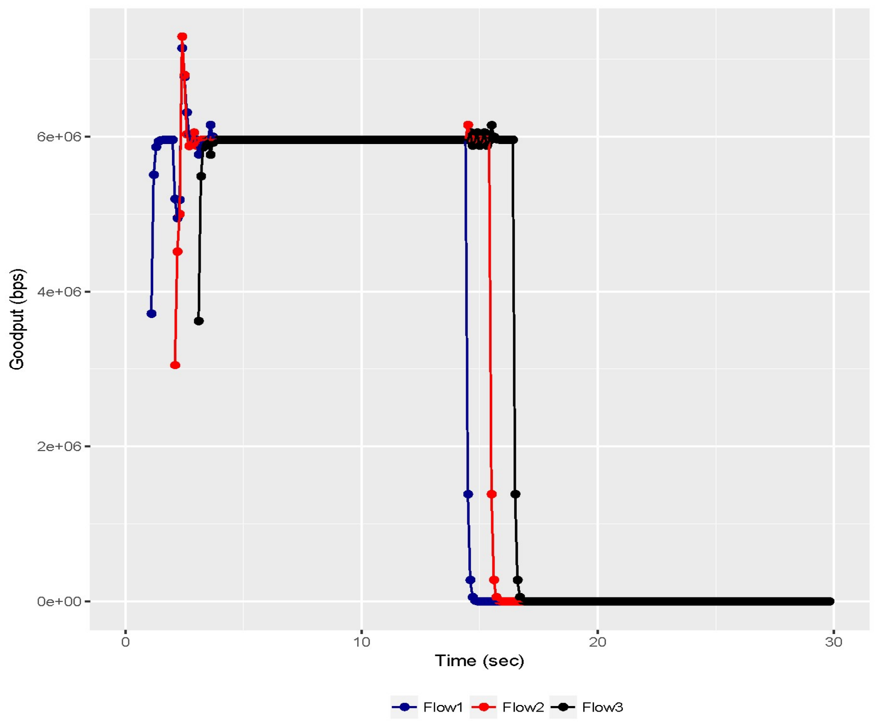


Figure 9: Goodput evaluation. Goodput time INRPP

In Figure 10, we can observe the number of drops in the network (the aggregate of all routers) for the three congestion protocols (ICP, PCON and INRPP) evaluated. Here we observe that the only protocol dropping packets is ICP, causing retransmissions. PCON avoids dropping packets, keeping the queues under control because of the CoDEL congestion detection and marking packets to reduce throughput before drops happen. INRPP also avoids dropping packets, since it is using the in-network caches to accumulate unsent traffic in the network routers, and not sending more packets than the links can absorb by sending exactly at the link rate.

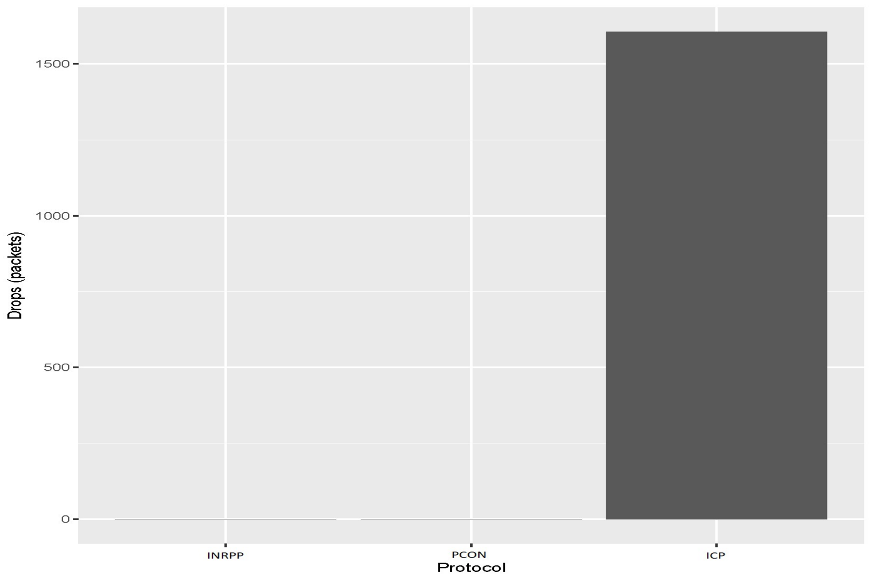


Figure 10: Dropped packets evaluation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

In Figure 11-Figure 13, we can observe the round-trip time (RTT)between the interests sent by consumers and the data packet received. In Figure 11, we can observe that for ICP protocol, there are some peaks for the RTT. This is exactly caused by the dropped packets and the retransmissions required by the protocol. Despite the average RTT is kept low, waiting for certain packets can cause buffer blocking at the application level, and therefore the delay perceived by the application is the peak delay in certain times, which is undesired.

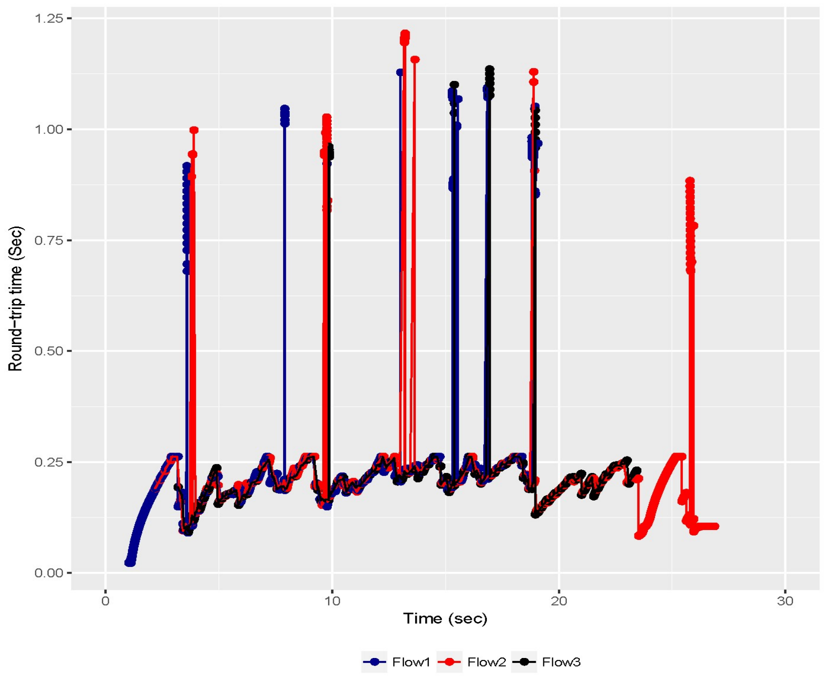


Figure 11: Round-trip time evaluation. Round-trip time ICP

In Figure 12, we can observe the RTT for the PCON protocol. In this case we can observe the RTT has less peaks than ICP because there are no retransmissions. However, PCON still cause some congestion in network queues that affects that the RTT is not completely constant and the RTT oscillates between bounded values (30-50ms). This behaviour is the expected performance for a congestion control since the RTT is kept low and bounded to acceptable values for constant bit-rate transmissions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

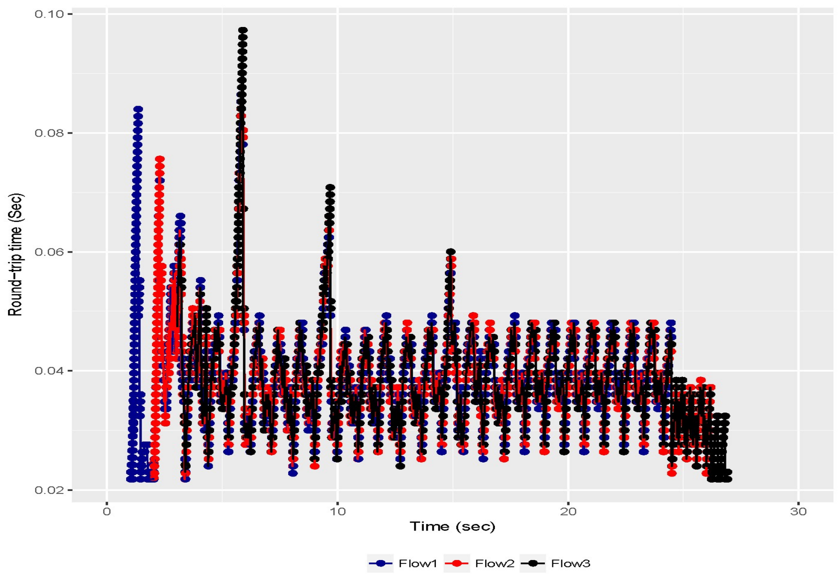


Figure 12: Round-trip time evaluation. Round-trip time PCON

In Figure 13, we can observe the RTT for the INRPP protocol. In this case, despite seeing an initial peak caused by sending interests in bulk initially, once the first data packets are received, the RTT is constant and are the lower values between the three options (lower than 30ms). This is caused mainly because INRPP is not dropping packets, is keeping network queues at minimum and the throughput is constant. That said, INRPP can cause RTT values can increase when network state changes (e.g. cross traffic) -not observed in the evaluation-, since there is not direct relation between the interest rate and the data rate in the INRPP protocol. However, in bulk transmissions the important is the data rate reception in the consumer and not the RTT between the interests and the corresponding data.

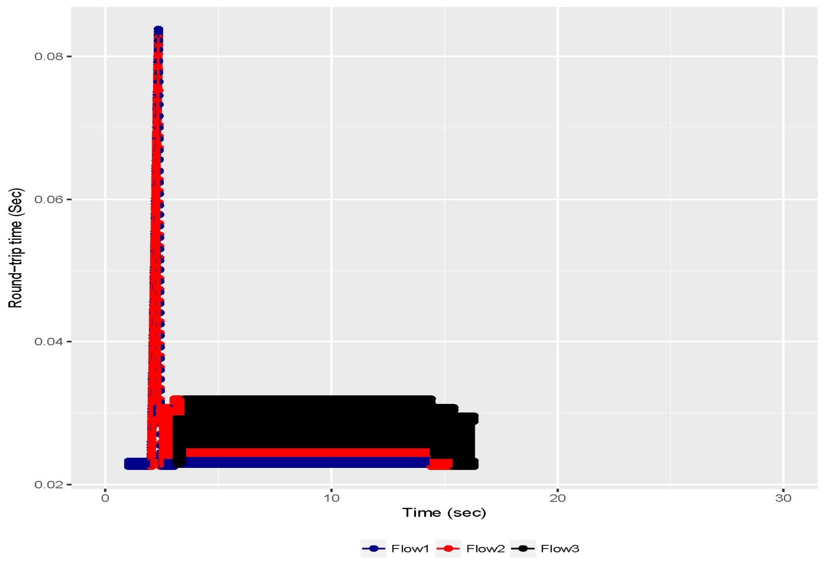


Figure 13: Round-trip time evaluation. Round-trip time INRPP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.1.5. Use case mapping

This service is not demonstrated in the proof-of-concept.

4.2. Opportunistic Off-Path Content Discovery (OOCD)

4.2.1. Description

In ICN, routing and forwarding is based on uniquely identified and authenticated content identifiers/names, rather than end-host addresses [6]. An important part of the in-network caching research in ICN is the actual content discovery mechanism, namely the mechanism to direct content requests to the right cache mainly to increase cache hit ratio and minimize content delivery latency. It can follow two approaches: i) opportunistic on-path routing, where content is searched on-path as the request is travelling towards the content source; or ii) coordinated off-path resolution-based, where requests are forwarded off the shortest path to some designated cache that islikelyto hold this content. The off-path resolution-based routing is a deterministic solution which maps requests to content items cached in nearby (or not) nodes, usually at predetermined rendezvous points e.g., [7]-[10].

Both on-path and off-path caching and content discovery present trade-offs. On-path mechanisms require less coordination and management, but may provide limited gains. Conversely, off-path techniques can attain higher hit rates at the cost of extra co-ordination and communication overhead. We try to combine the merits of both worlds by using traditional on-path mechanisms enhanced with a lightweight off-path content discovery approach that requires from each router to keep track of only a minimal amount of information in order to locate content.Particularly, we enhance the NDN router architecture with a new component called "Downstream Forwarding Information Table (D-FIB)", which keeps track of the direction (i.e., next hop) in which the Data packets were sent in the past. More details of the OOCD proposal will be provided in the final version of the architecture design in D3.2 and, its updated version, D3.4[13].For a more detailed evaluation of this component please check Section 4.2 in D5.1[1].

4.3. Named-based replication

4.3.1. Description

UMOBILE has developed in task 4.3 of WP4 a mobile name-based replication system, where message replication will be limited by time and space within a certain geographic area and with specific life expectancy, optimised by prioritisation rules considering emergency messages more important than other messages, e.g., messages between friends. There is no update on the evaluation of this component, for further information please refer to D4.3 [1].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.4. Keyword-based Mobile Application Sharing (KEBAPP)

4.4.1. Description

The Keyword-based Mobile Application Sharing (KEBAPP) framework enables users to share the applications, e.g. route planner,which they have on their mobile devices with nearby users that want access to processed information, which their own applications cannot provide. In a sense, the client application instance can also act as a server instance in order to serve requests from nearby users.There is no update on the evaluation of this component, for further information please refer to Section 4 in D5.1 [1].

4.4.2. Use case mapping

The KEBAPP service is showcased in the civil-protection use-case proof-of-concept.

4.5. IBR – DTN

4.5.1. Description

To facilitate intermittent communications in NDN, we have implemented a DTN face. This new face communicates with an underlying DTN implementation that handles intermittence by encapsulating Interest and Data packets in DTN bundles. Store-and-forward techniques are used by the underlying DTN implementation to reliably deliver the bundles utilizing standard DTN routing protocols. Upon reaching their destination, the bundles are decapsulated and the original Interest/Data messages are forwarded, resuming typical NDN communications. A possible deployment related to this demo includes a NDN-capable edge node (e.g., hotspot) that maintains a FIB entry towards the next edge node through the DTN face and one or more intermediate DTN mobile nodes that forward packets between the NDN nodes, essentially forming a tunnel.

By tunnelling traffic through DTN, we create an alternative, reliable communication channel for NDN in situations where typical TCP and UDP faces would fail. We also enable data forwarding between two remote NDN nodes by using different upstream and downstream intermediate (DTN-capable) routes/nodes, without any modification of NDN semantics. This can alleviate the fact that the NDN breadcrumb routing approach requires the same node that forwards an Interest to return the Data as well, inhibiting data delivery capabilities in such environments. While the DTN face can be readily used by the existing NDN forwarding strategies, it is in our future plans to design strategies that fully leverage its potential.

With the objective of having a full UMOBILE over DTN stack implementation on Android, it has been also developed an evolved version of the NDN Forwarding Daemon (NFD) for Android phones. That is, the UMOBILE version of NFD can natively recognize (create and route traffic via) DTN faces as an alternative when no other traditional face is available. This feature allows UMOBILE architecture to consider additional scenarios where a UMOBILE edge node cannot discover a distant equivalent peer if a DTN island separates them both.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.5.2. Demo Key Features

This service provides the following features:

- NDN/IBR – DTN integration
- DTN tunnelling functionality on an ICN network
- Data exchange between UMOBILE nodes with intermittent connectivity
- Data exchange between UMOBILE Android nodes via DTN face

4.5.3. Hardware and Software Requirements & Experimental Settings

IBR – DTN is experimentally validated. To that aim, we required the following:

- 1 Linux desktop
- 2 Android smartphones
- 1 hotspot

4.5.4. Validation & KPIs

DTN tunnelling functionality

We have performed a successful initial assessment of the IBR-DTN integration into the NDN forwarding daemon (NFD)[19],[3] which constitutes the basis of the UMOBILE platform. The tests proved that the modules interact correctly and messages can be exchanged between them. Tunnelling through IBR-DTN has been also achieved, as packets have been proved to be forwarded correctly via intermediate nodes which only run the IBR-DTN daemon. This is a critical milestone, as it constitutes the basic functionality to be offered by the module. As long as the validation equipment is concerned, it consists of:

- Two Ubuntu 14.04 desktops running the integrated NDN/IBR-DTN implementation
- Two Android 6 smartphones running the IBR-DTN implementation

The basic information about the test settings is depicted inFigure 14:

[3]https://github.com/umobileproject/ndn-dtn-neighbour-routes

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

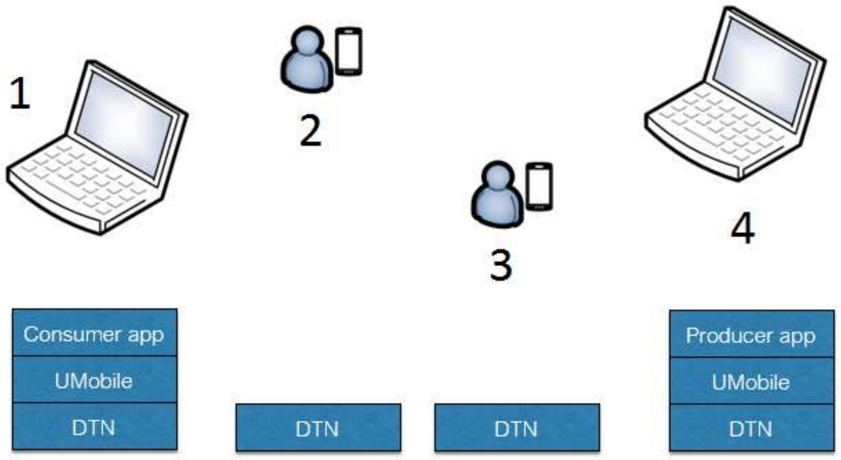


Figure 14: Evaluation setup for IBR-DTN

Two different scenarios were orchestrated. The first one included a basic setting to validate the correct integration of IBR-DTN with NDN and tested the basic functionality of the module. The second one simulated challenging network conditions in order to assess functionality of the component under network stress, such as the one to be present in the final demonstrations.

Scenario 1: Continuous end-to-end connectivity

The configuration for the scenario is depicted inFigure 15:

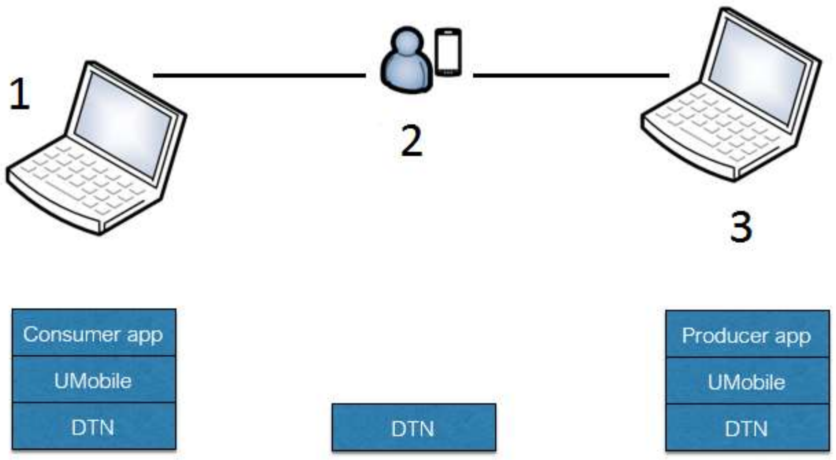


Figure 15: Scenario 1 for IBR-DTN

Two UMOBILE Ubuntu hosts (nodes 1 and 3) were set up to exchange messages, tunnelled via the Android IBR-DTN smartphone daemon (node 2). To this end, two sample NDN applications were used: *ndnping* and *ndnpeek/ndnpoke*. The Android phone served as WiFi hotspot, to which the laptops connected.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Both applications worked as expected and the hosts successfully exchanged data packets, validating that DTN tunnelling functionality has been integrated into the UMOBILE platform under the presence of regular network conditions.

Scenario 2: Intermittent connectivity

The configuration for the scenario is depicted inFigure 16:

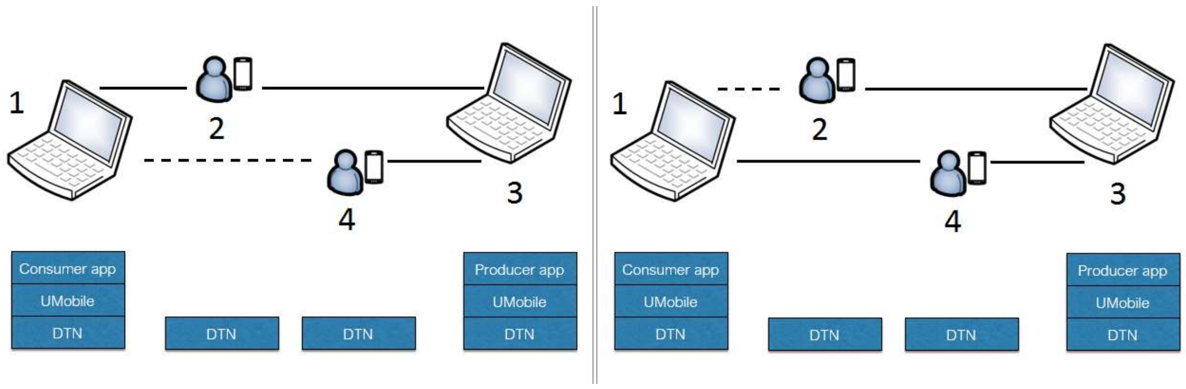


Figure 16: Scenario 2 for IBR-DTN

This time, 2 mobile hosts are present. Again, the *ndnping* and *ndnpeek/ndnpoke* applications were used and the Android smartphones were used as WiFi hotspots. As disruption-tolerant functionality was being assessed, Interest lifetime was set to a large value (60s) so that intermittent connectivity and large delays would not cause the Interests to be dropped (default interest lifetime is 4s). Intermittent connectivity and large delays were simulated by switching off WiFi links.

We note that we have set up static links so that Node 1 can only reach Node 3 via Node 2, and Node 3 can only reach Node 1 via Node 4. This was done in order to assess the enhanced functionality of a “DTN island”. When tunnelling over a “DTN Island”, the UMOBILE platform can use a different node to return the Data responding to an Interest, rather than the one that forwarded it. This way, an alternative is offered to the classic NDN breadcrumb routing approach and data availability is improved when the original node experiences disconnections.

The second test consisted of 3 phases presented inTable 2:

Table 2: Configuration of the different links

	Link 1-2	Link 2-3	Link 3-4	Link 4-1
Phase 1	Up	Up	Up	Down
Phase 2	Down	Up	Up	Down
Phase 3	Down	Up	Up	Up

The communication pattern is as follows:

- **Phase 1:** At first, Node 1 (consumer) sends several Interests to Node 3 (producer), which are tunnelled via Node 2 running the IBR-DTN daemon. The producer then responds with the corresponding data packets are tunnelled via Node 4 while trying to reach Node 1. At

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

this point, there is no connection between Node 4 and Node 1 so the data packets (encapsulated into Bundles) are stored by the IBR-DTN daemon in Node 4.

- **Phase 2:** Link 1-2 becomes inactive, so Interests stop to be forwarded to the Producer. At this point, Node 1 has no connection to the rest of the network. The corresponding data packets continue to be stored by the IBR-DTN daemon in Node 4, waiting for a connection to Node 1 to be delivered.
- **Phase 3:** Link 4-1 becomes active. IBR-DTN in node 4 detects the change and forwards the corresponding data packets back to Node 1.

Several iterations of the test were performed, with varying levels of network disconnections (from 5 to ~40 seconds). The flow of information above was validated at both the NDN producer/consumer terminals, as well as on the IBR-DTN mobile daemons. Network responses to altering the connectivity status were present, just as expected (e.g. continuous increase in data stored in the IBR daemon in Phase 1, no further increase in Phase 2, burst of returning Data packets in Phase 3).

Based on the experiments above, we can safely deduce that the IBR-DTN daemon has been successfully integrated into the UMOBILE platform. In both experiments, the module behaved as expected, while the second test solidified that IBR-DTN is now able to provide connectivity in the case of network disruptions to a pair of remote UMOBILE hosts.

Native NDN over IBR-DTN on Android

To validate a full functional Android node implementing a NFD version with IBR-DTN integration, we have performed some additional tests over two separate scenarios.

In the first place we wanted to validate the data exchange process over two Android phones running the NFD/IBR-DTN implementation. This is the scenario depicted in Figure 17.

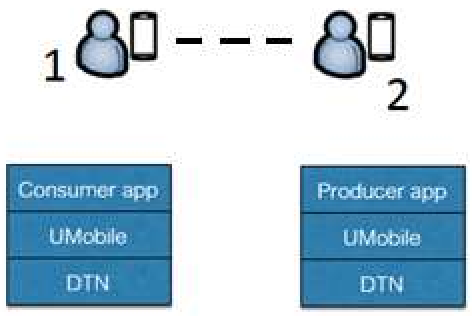


Figure 17: Scenario A for native functionality on Android

Both Smartphones are connected to the same WiFi SSID (so that IBR-DTN is able to discover its neighbours), and running the same set of applications: NFD modified daemon, IBR-DTN daemon and an NDN app on top acting as consumer/producer role.

We performed several tests with the NDN Whiteboard and Chronochat apps running on top.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

NFD daemon is started in the first place

In this case, there is no DTN neighbour discovery in the beginning. So the NFD daemon starts as usual, waiting either for the user to create a new pair face/route manually, or for an NDN app to do so automatically.

Whenever the IBR-DTN daemon is started on both nodes, the neighbour discovery process is triggered. Then it is established that phone 2 is a DTN neighbour for phone 1, and vice versa, so that their NFD daemons get a specific notification and can create an associated DTN face in a seamless way. From that point on, once there is an available DTN face for NDN traffic, when Whiteboard or Chronochat apps are launched, traffic can be immediately exchanged between the two Smartphones, although they don't expose a traditional TCP or UDP face for that purpose.

If the consumer/producer app is launched before IBR-DTN daemon, a communication error occurs (that can be observed from the NFD log), since there is no available route/face for forwarding those packets. After this error, if IBR-DTN is started within a short time period, the NFD daemon is able to send pending Interest packets and the messages get to be sent, experimenting a short delay though.

IBR-DTN daemon is started in the first place

In this case, the neighbour discovery occurs in the beginning. When NFD is started, a DTN face towards each DTN neighbour is already created during launch, and a general NDN route is registered.

Using either Chronochat or Whiteboard apps produces the same immediate effect of automatic data exchange, via the native DTN face on each phone.

NDN consumer/producer app is launched in the first place

In this case, if you try to exchange NDN traffic from one of the apps, neither Whiteboard nor Chronochat is successful, just the same way when the app on top cannot detect an NFD daemon available underneath. If NFD is started, this common error is eliminated and the test performs like one of the cases described above.

We also reproduced these three situations but introducing intermittent connectivity between the two DTN neighbours. You can easily control connectivity in this scenario by altering the on/off switch estate of the IBR-DTN daemon. That is, if you change estate to 'off', the DTN face on the NFD daemon will not reach is destination peer, so that traffic gets interrupted. If then you select 'on' again, neighbour phone is visible through DTN and traffic exchange is restored.

The second scenario we used to validate a full stack implementation of UMOBILE over IBR-DTN on Android was the one represented inFigure 18.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

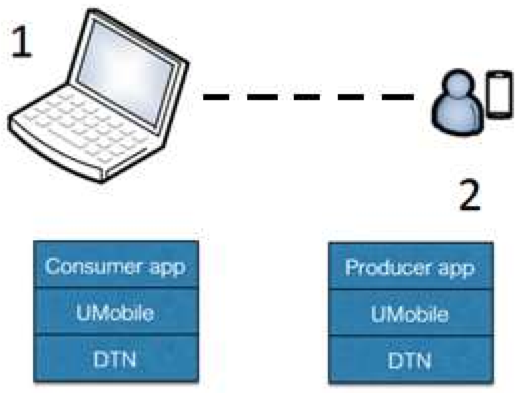


Figure 18: Scenario B for native functionality on Android

For testing this scenario we used the *ndnping* (Smartphone) and *ndnpingserver* (Linux desktop) applications on top of UMOBILE/IBR-DTN. In this case, the objective was simple: validate that the native functionality integrated on Android was also functional when exchanging data with a node executed from a Linux desktop.

Tests were performed with both nodes under the same WiFi SSID, so that they could see each other as a DTN neighbour. If both started their IBR-DTN daemons, there was no problem with the neighbour discovery, so that the phone automatically created a DTN face for node 1, and NDN traffic was forwarded via that active face. Some adjusting with the IBR synchronization was required so as to be able to exchange traffic though.

Then the *ndnpingserver* was launched on node 1, and the *ndnping* from node 2 was successful and got response without problems. If connection was disrupted (on/off switch on IBR-DTN daemon), then the effect was immediate in the traffic disruption experimented, but was also recovered according to the DTN face availability state.

4.6. Service Migration

4.6.1. Description

A central objective of the UMOBILE project is to provide network connectivity in locations with limited connectivity or no connectivity at all. To address the challenge, we are developing a service migration platform that on the basis of information collected by monitors, can strategically deploy services to ameliorate the impact of network problems or ideally, to prevent the materialization of threats. As explained in deliverable D5.3 [1], to demonstrate that the service migration platform has a potential to solve the problem, we will use an emergency scenario as use case. The central idea is to use the service migration platform to deploy services to be used by emergency teams and victims in areas struck by undesirable events (fires, floods, earthquakes, etc.) where the network infrastructure is absent or disturbed by the event.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

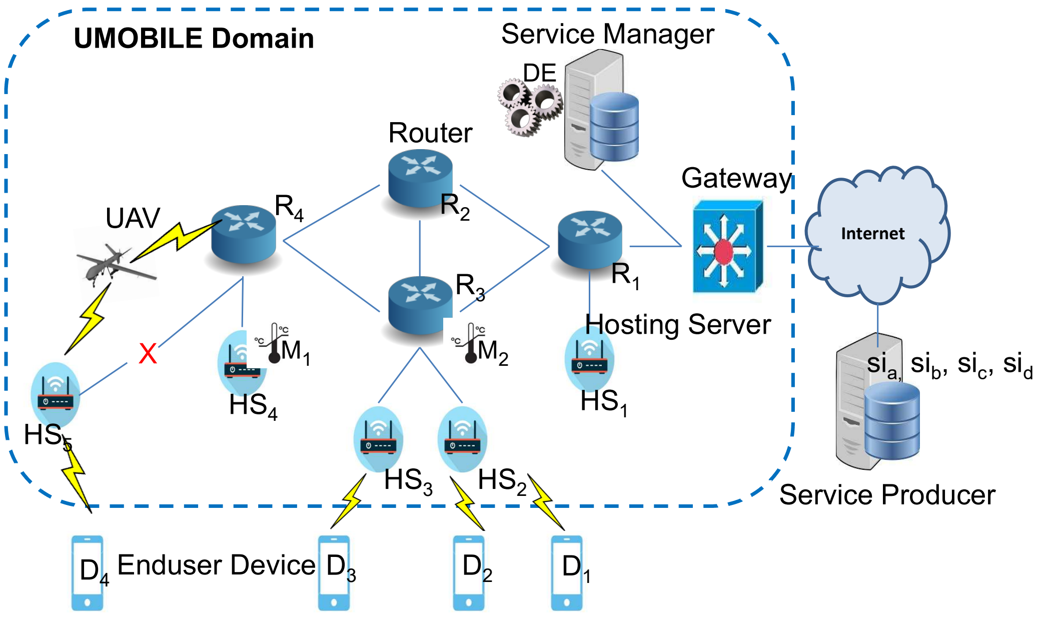


Figure 19: View of UMOBILE deployment with focus on service migration

In the following discussion we assume one of the emerging business models where the network operator is responsible for providing both network connectivity and access to content and services to their end-users. The network operator is not necessarily the owner of the services but it acts (on behalf of the actual owner of the services) as the party responsible for the availability of the services and their quality.

Imagine for example, the components shown in Figure 19(Service Producer, Service Manager, Routers, Gateway and Hosting Server) are instrumented with the service migration platform. Assume that the Service Producer is in possession of compressed images of several services (Si_a, Si_b, Si_c and Si_d) and that he has delegated to the network operator the responsibility of delivering them to the end-user upon requests. For example, S_a is the service that can be enacted from its compressed image Si_a. Let us assume that S_a is a service needed by a rescue team involved in the emergency scenario. The compressed image Si_a is accessible to the Service Manager from the Service Producer and might be stored locally (within the Service manager) as well.

A central objective of the UMOBILE project is to provide network connectivity with different levels of QoS, ranging from best effort to more stringent levels. Of particular interest to UMOBILE is the provision of services in locations with limited network connectivity or no connectivity at all as illustrated by the network link between R_4 and HS_5.

To address the challenge, we are developing a **service migration platform** that uses a decision engine (shown as DE in the figure) that on the basis of information collected by monitors can strategically deploy services to ameliorate the impact of network problems or ideally, to prevent the materialization of threats. Only two monitors (M_1, M_2) are shown in the figure yet there are no technical restrictions to deploy as many as necessary in strategic locations as they are software components (for example, Python scripts).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.6.2. Validation Key Features

In D5.1, the validations of service migration were focussed on the quantification and capabilities. For instance, we evaluated the scalability of service migration on UMOBILE hotspot devices. In those validations, we were able to quantify the maximum number of services that could be run on the single UMOBILE hotspot (i.e., raspberry Pi) as well as the maximum number of simultaneous users who can access to a service instance (i.e.,Docker container) at the same time. Furthermore, we were also able to identify the critical parameters that impact the service quality. Those parameters include CPU usage, memory usage and CPU load. As a result, we applied this knowledge and understanding to develop the decision engine in order to provide the QoS in UMOBILE network, the results of this study was also reported in D4.4[20]. In this deliverable (D5.2), we aim to further validate the functionality of service migration platform while focusing on the challenge network environment. The goal of this evaluation is to prepare and validate the service migration platform to support the scenario in PoC1.

This evaluation has three objectives:

- Objective 1 (O1): To validate the service deployment and execution functions of service migration platform with the scenario explained in PoC1
- Objective 2 (O2): To validate the emergency application used in PoC1

In pursuit of O1 and O2, we have setup the emergency scenario in the lab as illustrated inFigure 20. We assume that HS1 is an UMOBILE hotspot located in the disaster area which is isolated from the main UMOBILE network. In this scenario, the service manager and service repository are configured in the main UMOBILE network. We assume that the civil protection authorities upload their emergency services to the Service Repository before the service deployment operation. The aim of this experiment is to push a service (S) from service repository and deploy it in HS1. Notice that in this configuration, we assume that the service repository is co-allocated within the service manager machine. Service S is a stateless service (e.g., a self-contained web server) which provides the emergency information along with GPS location. For instance, the service will provide the location of the food shelter or safe place to the end users getting stuck in the disaster area. The image of the service is available in [19]. In this validation, we use an Android phone as a DTN node that establishes a DTN link between HS1 and the service manager. In fact, any smart device attached to a UAV or any other means of transport can replace the Android phone, as long as it supports a DTN framework (e.g., IBR-DTN).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

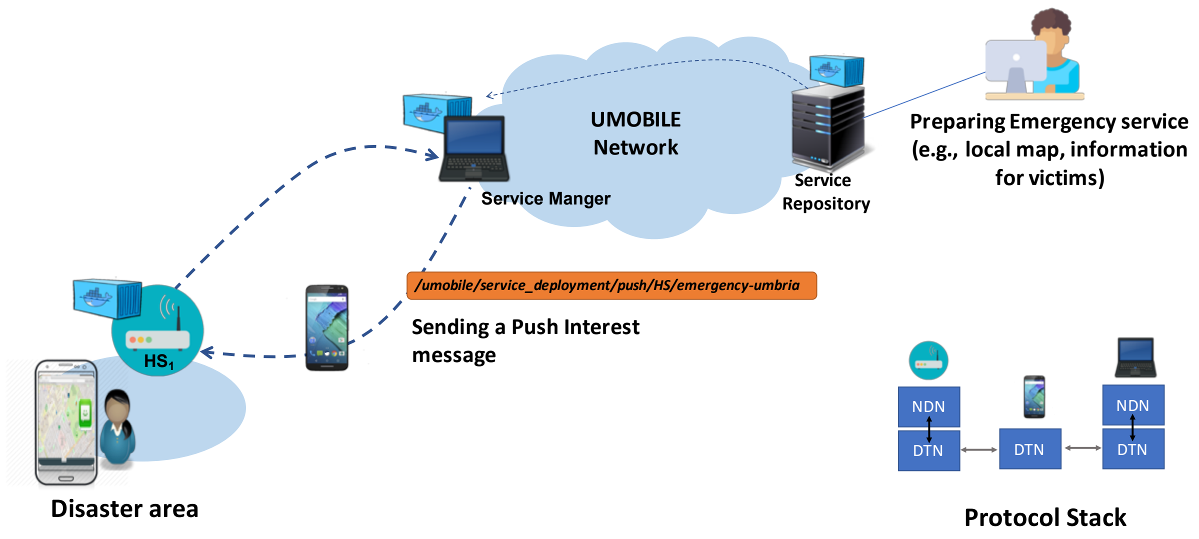
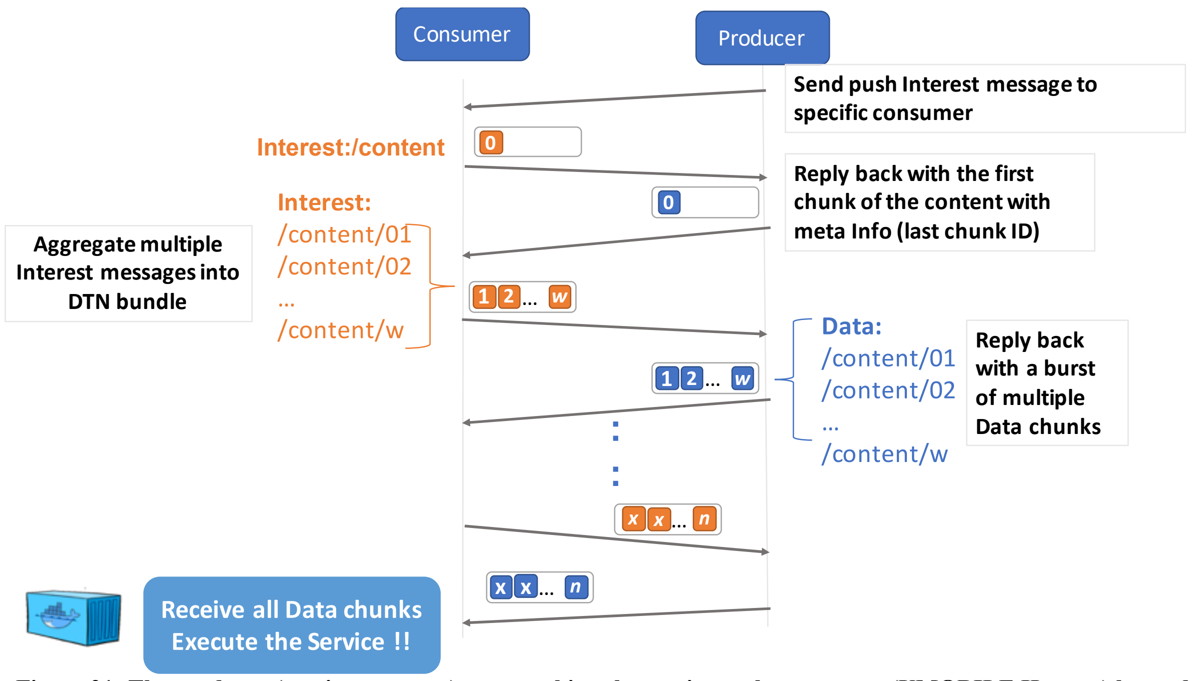

UMOBILE Network
Service Manger
Service Repository
Preparing Emergency service (e.g., local map, information for victims)
/umobile/service_deployment/push/HS/emergency-umbria
Sending a Push Interest message
HS1
Disaster area
NDN
DTN
DTN
NDN
DTN
Protocol Stack

Figure 20: Test scenario for service deployment in challenge network environment

Validating the service deployment and execution functions of service migration for PoC1

As NDN natively follows a synchronous communication model where a consumer sends one Interest message to retrieve one Data message in turn, then this model is inadequate in scenarios where there is a single communication link between the consumer and producer, like in the disaster scenario depicted in Figure 20. In this example, the Android phone would need to travel several times between the main network and the disaster area to deliver all the data chunks. As a solution, we implemented a multi-Interest forwarding model to support the NDN forwarding daemon: the idea is to allow the consumer to issue an aggregation (burst) of w Interests and retrieve multiple Data chunks at once.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124



Consumer
Producer
Send push Interest message to specific consumer
Interest:/content
0
Reply back with the first chunk of the content with meta Info (last chunk ID)
0
Interest:
/content/01
/content/02
...
/content/w
Aggregate multiple Interest messages into DTN bundle
1 2 ... w
Data:
/content/01
/content/02
...
/content/w
Reply back with a burst of multiple Data chunks
1 2 ... w
x x ... n
x x ... n
Receive all Data chunks
Execute the Service !!

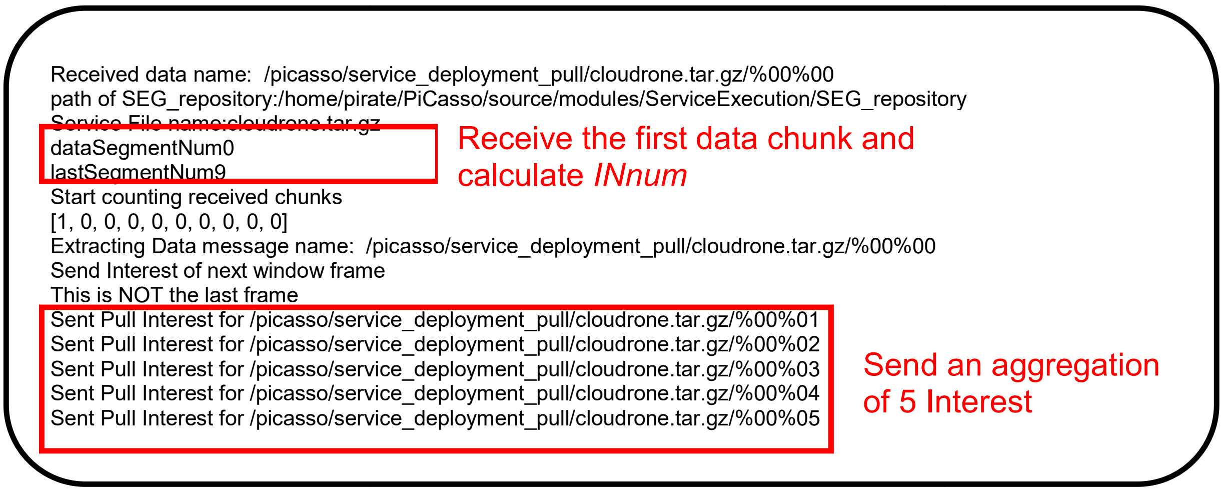
Figure 21: The producer (service manager) start pushing the service to the consumer (UMOBILE Hotspot) located in the remote area via DTN tunneling

Figure 21illustrates the service deployment process between the service manager (producer) to the HS1 (consumer) in the disaster area. The producer initially sends a push Interest message with the name of emergency service. As a consequence, a consumer sends back a pull Interest message requesting a service, which is forwarded towards the same path as the push message was sent. Along with the first Data message, the consumer receives information about the total content size (Scon) and NDN data chunk size (Schunk). On this basis, it can calculate the total number of Interest messages (INnum) to be sent in order to retrieve the whole content size.

$$INnum = \left\lceil \frac{Scon}{Schunk} \right\rceil$$

The consumer can request all remaining chunks within the second request by setting w = 1 − INnum at the risk of creating network congestion if the content size is large. To avoid the problem, the value of w should be dynamic regarding the network condition. In our experiment, we set the data chunk size (Schunk) as 2MB and burst size (w) as 5 messages. Therefore, the DTN bundle can contain the size of data chunks up to 10MB per request. The size of our emergency service is about 18 MB therefore the Android phone (DTN bundle) needs to travel forward and backward around 3 times including the first push Interest message.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124



```
Received data name:  /picasso/service_deployment_pull/cloudrone.tar.gz/%00%00
path of SEG_repository:/home/pirate/PiCasso/source/modules/ServiceExecution/SEG_repository
Service File name:cloudrone.tar.gz
dataSegmentNum0
lastSegmentNum9
Start counting received chunks
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Extracting Data message name:  /picasso/service_deployment_pull/cloudrone.tar.gz/%00%00
Send Interest of next window frame
This is NOT the last frame
Sent Pull Interest for /picasso/service_deployment_pull/cloudrone.tar.gz/%00%01
Sent Pull Interest for /picasso/service_deployment_pull/cloudrone.tar.gz/%00%02
Sent Pull Interest for /picasso/service_deployment_pull/cloudrone.tar.gz/%00%03
Sent Pull Interest for /picasso/service_deployment_pull/cloudrone.tar.gz/%00%04
Sent Pull Interest for /picasso/service_deployment_pull/cloudrone.tar.gz/%00%05
```

Receive the first data chunk and calculate *INnum*
Send an aggregation of 5 Interest

Figure 22: Screenshot of HS1 when it receives the first data chunk and sends back with an aggregation of multiple Interest messages.

Figure 22 shows the screenshot of HS1 when it receives the first data message. It uses the information inside this Data message to calculate the total number of Interest messages (INnum). In this example, the INnum is 10 data chunks. As the burst size (w) is set as 5, the HS1 sends back an aggregation of 5 Interest messages. In return, the Data mule (Android phone) carries 5 Data messages (i.e., chunk 1 to 5) in order to delivery to HS1.

Once the HS1 receives all the data chunks, it calls the service execution function to execute the service image and provide the emergency service to local users via WiFi. Figure 23shows the user interface of our UMOBILE emergency service. The end users in remote area can look for emergency information like safe place, food shelter, and clean water station on the local map. In addition, the end user can interact with other local users by sending the chat message or uploading the photos with the GPS location.

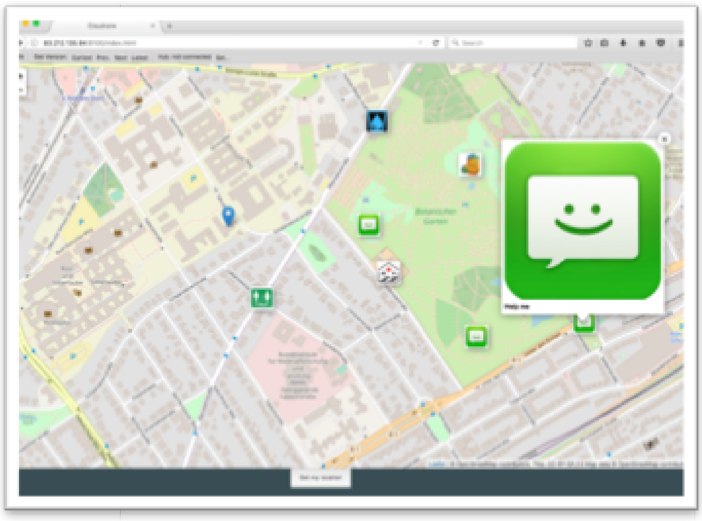


Figure 23: User interface of UMOBILE emergency service

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.6.3. Hardware and software requirements and experimental settings

- Hardware
    - a Linux machine to run as service manager and service repository
    - 1 raspberry Pi to operate as UMOBILE hotspot (HS1)
    - An Android phone with IBR-DTN capability to operate as DTN data mule
- Software
    - Linux debian operating system for service manager
    - Hypriot OS Version 1.2.03, a customized Rasbian integrated with Docker daemon
    - NFD
    - Service migration code (can be found in https://github.com/umobileproject/PiCasso)
- Experimental settings
    - To facilitate the experiment setup, we have prepared the image for raspberry Pi to run as the UMOBILE hotspot[4][19]https://github.com/umobileproject/raspi-image). This image includes all necessary software for service migration platform such as service migration source code, NFD and DTN integration code

4.6.4. Use case mapping

Emergency scenario and civil protection (PoC1)

4.7. Context Based Routing in Information-centric Opportunistic Environments

4.7.1. Description

Combining opportunistic networking principles with ICN principles is relevant to efficiently evolve wireless architectures to i) improve service delivery (e.g., take advantage of traffic locality); ii) improve the social routine of the end-user via technology (e.g., assist in the development of services that take into consideration crowd analysis parameters, or interests shared among familiar strangers).

In such context, data transfer is performed via mobile, end-user services, and users carrying mobile devices will not always be aware of the origin of the content/service. Likewise, the sources of data and services may not even know the destination. Hence, the principles of *carry-and-forward* are not applicable in this case, being replaced by the data-centric approach of *store-and-forward,* while having to deal with the high topological variability inherent to opportunistic networking. Routing in these environments needs to consider forwarding strategies that take into consideration, at a network level, *context* that surrounds users and that can assist in better defining opportunities for data transmission over time, and space. Such opportunities can be better addressed if routing metrics take into consideration levels of *contextual similarity* between devices in what concerns data transmission. Contextual similarity has, in this work, two different dimensions: i) affinity network characterization data; ii) usage and similarity characterization

[4]https://github.com/umobileproject/raspi-image

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

data. Affinity network information concerns e.g., on peer status over time and space as well as affinities (matches) between source nodes and peers. Usage and similarity characterization are built upon data collected internally (in the device), e.g., battery status, type and/or category of preferred applications; time spent per application category.

Metrics derived from such contextualization can then be applied to opportunistic routing in ICN/NDN scenarios, to assist in interest packets and/or data forwarding.

This demo concerns the use of the UMOBILE Contextual Manager to assist multihop data transmission (routing computation) in the context of information-centric environments. For this purpose, we have developed a routing interface that interconnects the Contextual Manager (a service that resides in the UMOBILE End-User system) and the opportunistic routing approach (DABBER, Information-centric Routing for Opportunistic Networking Environments) being developed[25]. The aim is to show that contextual awareness both in what concerns the usual approach of node popularity (betweenness) and new indicators concerning node availability (status of resources) as well as nodes' similarity (affinity in terms of resource and app usage) can be beneficial in the context of information-centric opportunistic routing.

This demo is developed having in mind the UMOBILE social routine scenario as described in [2]-[3]. The purpose of this demo is therefore to show how to extend the support of UMOBILE to opportunistic scenarios. We consider two main approaches for data dissemination in the demo. The first is a push-oriented mode, which we shall demo based on the UMOBILE Oi! Software [24]. The second is the usual pull model which we have integrated into the UMOBILE Now@ broadcaster solution.

4.7.2. Demo Key Features

The demo is based on 3 main technical contributions: I) NDN-Opp, which provides support for opportunistic communications in information-centric environments; ii) DABBER (integrated in NDN-Opp), which is an NLSR extension proposal to allow current NDN-based routing to operate in opportunistic environments; iii) the routing interface towards the Contextual Manager (internal interface), which is used to get information from the Contextual Manager and to send it to NDN-Opp upon request or periodically.

This demo is being developed in a local pole in Lisbon, compose of 6 Android nodes from COPELABS, as illustrated in Figure 27. This pole is interconnected to the COPELABS NDN testbed, which is a part of the worldwide testbed.

Each node (Android smartphone) has the UMOBILE UES installed, with the Contextual Manager running. The key features of the demo are:

- Showing that contextual awareness improves routing without incurring a significant overhead.
- Understand which indicators are the most relevant, for the computation of successor availability and betweenness

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

- Understand whether or not measures of similarity (in addition to betweenness and availability) are relevant in the context of routing improvement (and as a consequence higher QoE end-to-end).

4.7.3. Hardware and Software Requirements & Experimental Settings

- Hardware and software requirements:
    - Hardware
        - 5 Samsung S5 NEO Android Smartphones
        - 1 NDN router installed at COPELABS, and that provides access to the NDN global testbed.
        - 2 Access points (to allow end-user devices to connect to the NDN test-bed in two different physical locations)
    - Software
        - Android 5.1 (min 4.2, max: 6?)
        - NFD for Android
        - NDN-Opp (based on NFD for Android) allowing devices to exploit direct wireless communications to exchange data.
        - Contextual Manager installed on the devices (integrated into the UMOBILE EUS)
        - Oi! and Now@ applications (which run on top of NDN-Opp)
- Experimental Settings
    - Creation of different settings (minimum involvement of 5 devices; maximum of 10 and 3 different clusters)

4.7.4. Validation & KPIs

The validation shall be performed on the testbed illustrated in Figure 26. The benchmark shall be based on flooding (NDN-Opp version 1.0).

Relevant KPIs in this demo are: time-to-completion and copy overhead. Time-to-completion refers to, in a push model, the time it takes to trigger a message, until the instant of time the destination receives such message. While for the case of a pull model, time-to-completion concerns the instant of time between the sending of an Interest packet (by a source) until the first reception of a requested information category.

For KPI validation our benchmark shall be based on a flooding model which has been implemented in NDN-Opp version 1.0. Hence, the purpose of this demo is to show that, in addition to robustness, routing based on contextual awareness can be beneficial for information-centric approaches in opportunistic environments.

The key performance indicators used in these two validations set are related to:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

- **Compatibility with the NDN framework**. In this case we aim to analyse how can devices implementing NDN-Opp communicate with devices implementing the regular NDN framework. In this case the indictors are related to the *rate of successful transmissions.*
- **Performance with NDN-Opp in a fully opportunistic scenario**. In this case we aim to analyse the performance of the supported routing solutions while exploiting existing WiFi direct transmissions in terms of the *percentage of delivered messages*, the cost of communications in terms of the ration between delivered and replicated messages, and the delay of communications.
- **Performance of NDN-Opp in a hybrid networking scenario**. In this case we aim to analyse the performance of NDN-Opp in scenarios where devices can communicate directly, via existing WiFi direct connections, as well as indirectly, via an existing WiFi infrastructure. We will use the two KPIs described above.
- **Analysis of the best solution to support push communication models in an opportunistic scenario**. In this case we aim to analyse different solution in terms of its *scalability* (e.g. size of the PIT table) and *delay* (e.g. number of used messages, for instance in the case of three hand-shake approaches). The scalability will be measured in terms of memory needed for smart routing when the networks grows in number of nodes as well as in time needed for processing of all the information in order to decide whether to forward the packets or not.
- **Performance of Now@ application in terms of its capability to keep data synchronized among devices sharing the same interests.** In this case the key performance indicators will be the *number of messages exchanged* to keep data synchronized and the *delay in synchronizing* data among the set of devices that showed interested in that data.

While the Now@ application has a clear set of performance indicators, the same those not happen with the Oi! application. In the latter case, since Oi! is an instant message application, its operation rely fully on the performance of NDN-Opp in terms of the *message delivery success rate* and *message delay*.

4.7.5. Use case mapping

The validation of the UMOBILE end-user device developed based on NDN-Opp and the Contextual Manager will be done based on the Social Routine Improvement Scenario, corresponding to the second proof-of-concept presented in deliverable 5.3 [1].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.8. UAV Service

An important aspect to have into account during the evaluation phase of the UAVs is the confirmation that it is established a secure and flawless transmission of information between the user, located at the ground, and the UAV – which can be either the AR3 or the multicopter VR1.

Therefore, the communication requires the consideration of the existing conditions during the transmission of the information – either for personal emergencies (if a civilian needs medical assistance, for instance), detection of accidents or danger situations, or simply to access the Wi-Fi network in areas without connectivity. The transmission is then ensured through a Wi-Fi connection between the user and the UAV. For this end, certain requirements must be considered. The most obvious is that the user that needs the connection has a smartphone, in order to initiate the transmission of the intended message. From the other side of the connection, an UAV has included in its payload a specific tool – the metal boxed antenna Mikrotik 2SHPn – that, with its appropriated characteristics for these missions (such as high-power transmitting) and for the conditions where it will actuate (such as impermeability, resistance, sealing), allow the messaging reception and delivery to the user or to the GCS on the ground. In fact, after the establishment of this Wi-Fi transmission, it is then possible to validate the air-ground messaging exchange through NDN with DTN data, for the diverse possible situations (including the previously mentioned emergency cases).

However, the main situations related to the use of UAVs for the project concern two fundamental methodologies: the first one is its application as an end-user to migrate data, and the second is its operation as an access point to migrate data (that can be more or less heavy) for an operation center.

Finally, on the other side, it is relevant to distinct whether it should be used the AR3 or the VR1. This decision mainly comprehends on the dimensions of the actuating area – where the civilian user is in need of connectivity. In fact, the appliance of the AR3 is more preponderant in environments with a larger area and without network coverage. Due to its features as an UAV, AR3 allows the monitoring of wide areas. However, it is crucial to have into account that the UAV speed has to comply with certain limits, in order to prevent communication gaps with the smartphone of the ground-user. With regard for the system to proceed correctly, it was previously studied the appliance of loiter maneuver by the AR3, concretely flying around the local where the eventual message is going to be transmitted from the ground, as it is illustrated in the next figure. It is necessary, then, to have into consideration that the distance to the user's smartphone does not constitute a major problem for the UAV-user connection.

This project has received funding from the European Union's Horizon 2020 research and innovation programme
under grant agreement No 645124

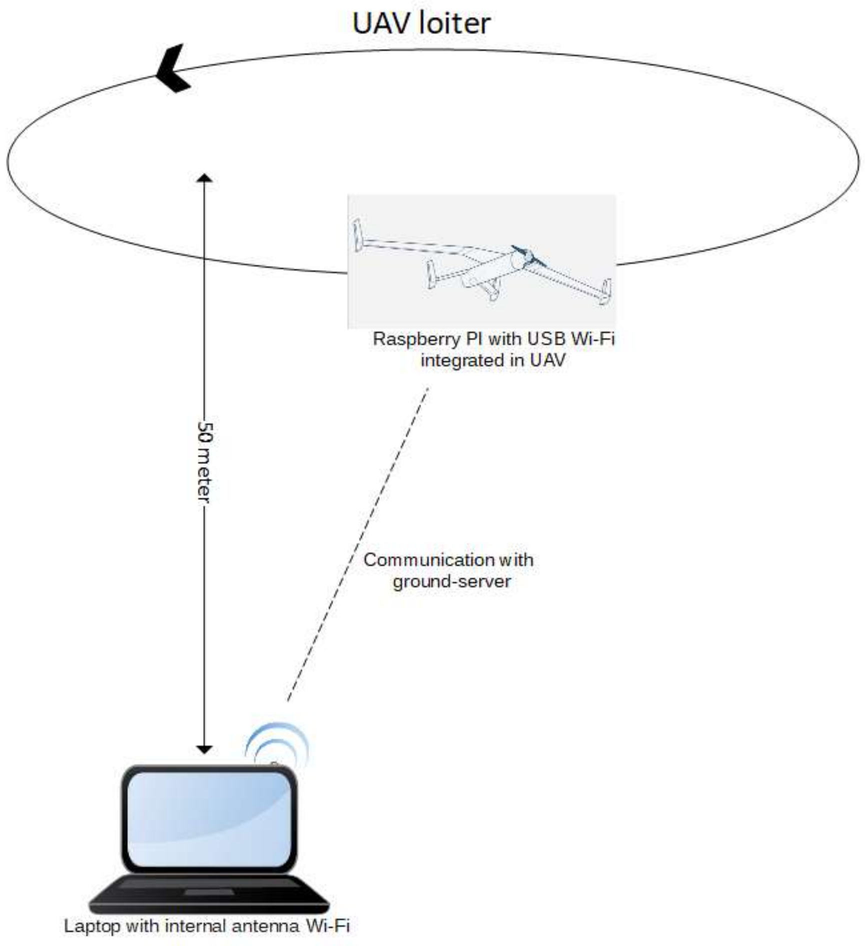

UAV loiter
Raspberry PI with USB Wi-Fi integrated in UAV
50 meter
Communication with ground-server
Laptop with internal antenna Wi-Fi

Figure 24: Performance of the loiter maneuver by the AR3

By its turn, the VR1 will have a propensity to be more incorporated in messaging/data transmissions in small local areas, where the connectivity may present network gaps. VR1 function is essentially the same as in AR3, however, due to its hover capacities the multicopter operates statically nearly above the user, allowing a full secured WiFi ground-air connectivity. This procedure, illustrated in the next figure, guarantees a static data transmission, as long as the multicopter is not too distant in relation to the user's smartphone on the ground.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

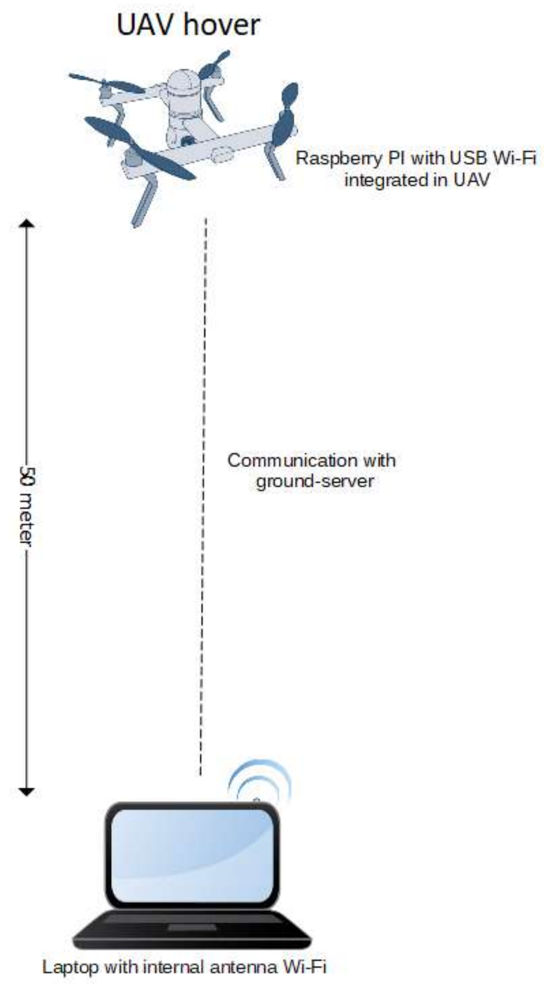

UAV hover
Raspberry PI with USB Wi-Fi integrated in UAV
50 meter
Communication with ground-server
Laptop with internal antenna Wi-Fi

Figure 25: Performance of the hover by the VR1

In the tables below, it is listed the equipment that is included in the UAVs for POC1.1 and POC1.2, with the correspondent additional info for each device. The mentioned equipment allows the validation of the UAV-ground transmissions during the emergency scenarios that were previously described.

In POC1.1, that consists in the first stage where the initial user send a request message that is aimed to the local authority, the UAV is used as a hotspot and data mule and operates with a payload that includes the following set of devices listed in Table 3:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Table 3: Payload of the UAV in POC1.1

Equipment	Additional Info
BananaPI	Banana PI BPI-M3
UAV	AR3 or VR1
GCS	Tekever ground control station
WLAN Adapter	Mikrotik Metal 2SHPN
Android Device*2	Smartphone with Android 5.1 or more

On the other hand, in POC1.2 – that consists in the reply path of the initial request (this means the path of the information sent that by the local authority intended to be delivered to the rescue team and corresponding citizens that may be in an emergency situation), is also accomplished with the support of the UAV as an hotspot and data mule. Therefore, this UAV is incorporated with the following devices described inTable 4:

Table 4: Payload of the UAV in POC1.2

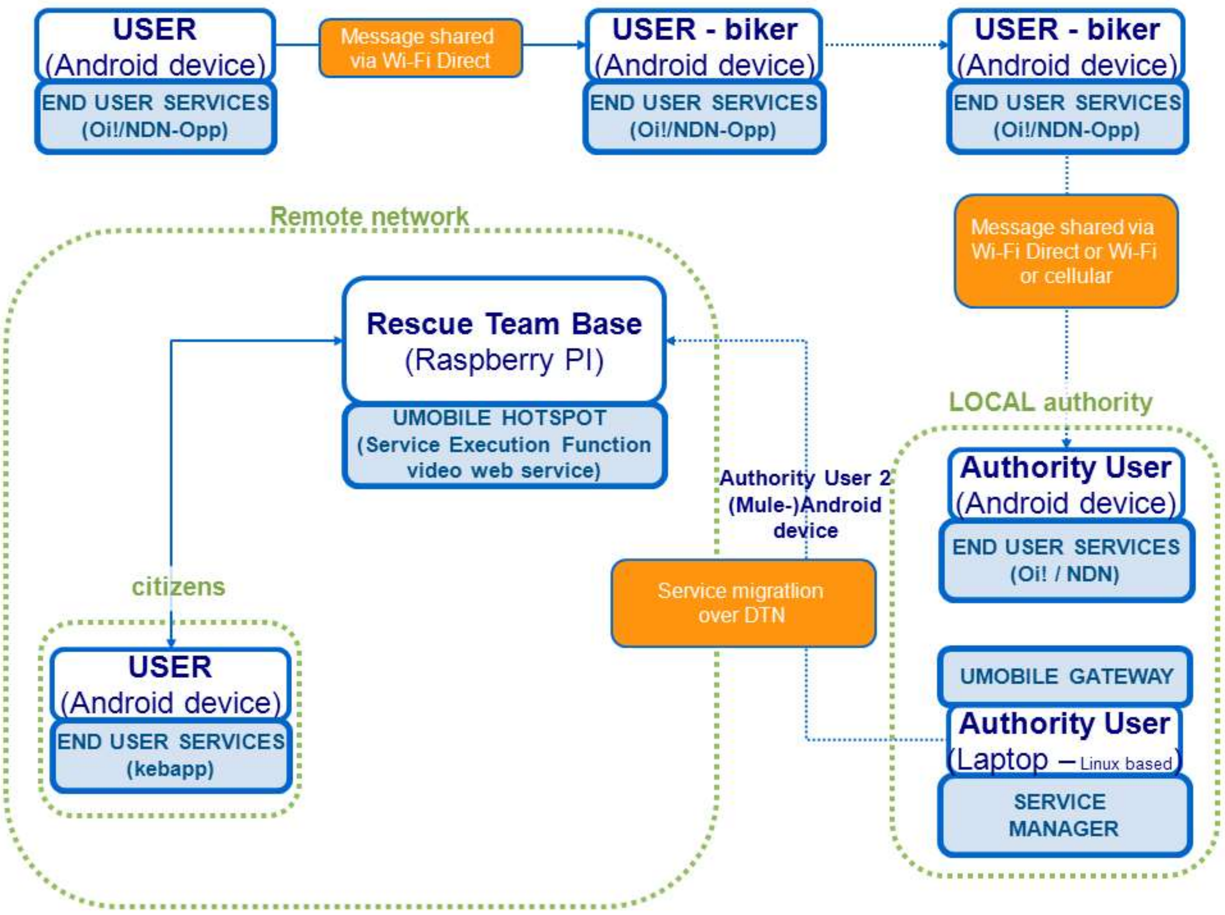
Equipment	Additional Info
UMOBILE Hotspot	Raspberry PI 3
UAV	AR3 or VR1
GCS	Tekever ground control station
WLAN Adapter	Mikrotik Metal 2SHPN
UMOBILE Gateway	Raspberry PI 3
Authority User	Laptop with Ubuntu Software

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

5. UMOBILE System Integration &Validation

In this section we present the validation of the UMOBILE system platform in terms of integration among different UMOBILE components following the proof-of-concept description carried out in D5.3[2] (and the updated version D5.4 [3]). Note that D5.5 will update results on validation and final integration.

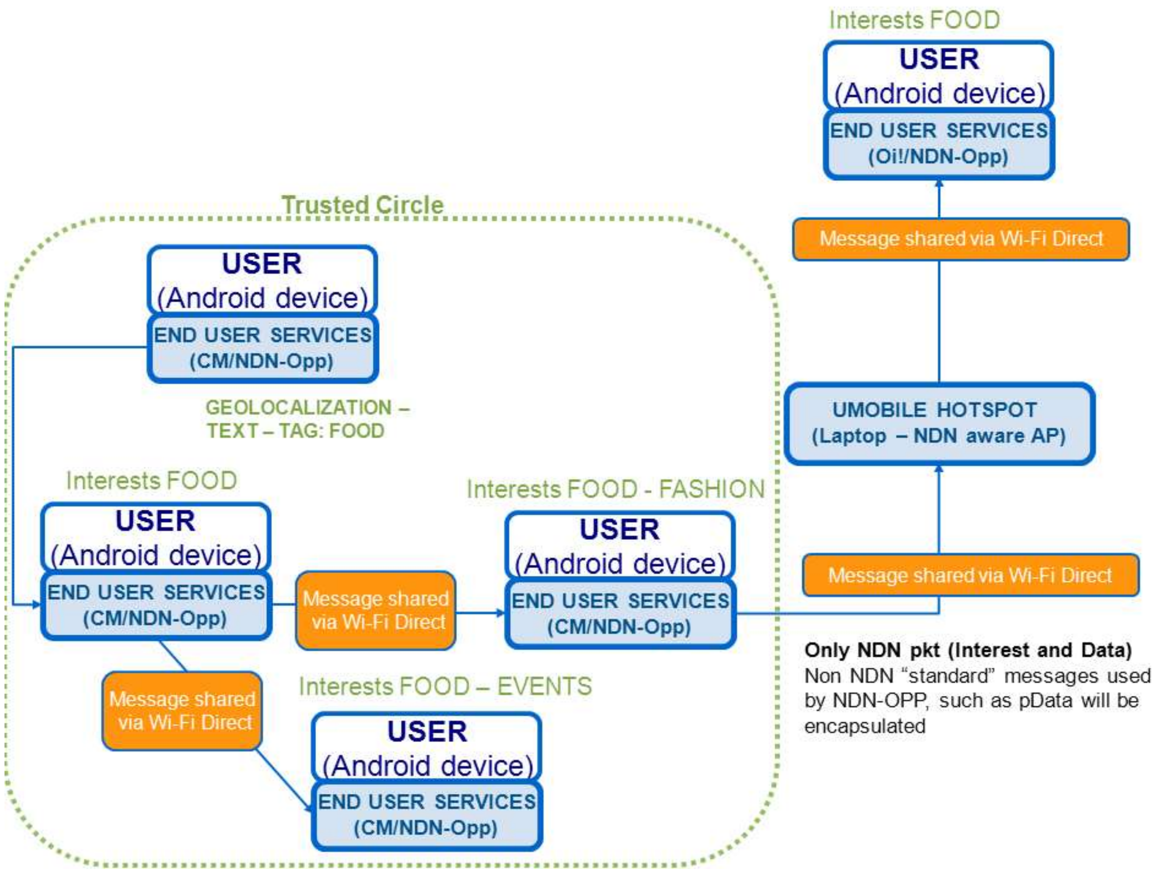
Following we present the integration and interaction between different UMOBILE components for each of the described POCs.Figure 26 identifies each of the components, services, hardware and OS for POC1, the one regarding the "Civil and Emergency Scenario".



USER (Android device)
END USER SERVICES (Oi!/NDN-Opp)
Message shared via Wi-Fi Direct
USER - biker (Android device)
END USER SERVICES (Oi!/NDN-Opp)
USER - biker (Android device)
END USER SERVICES (Oi!/NDN-Opp)
Message shared via Wi-Fi Direct or Wi-Fi or cellular
Remote network
Rescue Team Base (Raspberry PI)
UMOBILE HOTSPOT (Service Execution Function video web service)
Authority User 2 (Mule-)Android device
LOCAL authority
Authority User (Android device)
END USER SERVICES (Oi! / NDN)
citizens
USER (Android device)
END USER SERVICES (kebapp)
Service migration over DTN
UMOBILE GATEWAY
Authority User (Laptop – Linux based)
SERVICE MANAGER

Figure 26: POC1 block diagram scheme

Figure 27 identifies each of the components, services, hardware and OS for POC2, the one regarding the "Service Announcement and Social-Routine Scenario".



Interests FOOD
USER (Android device)
END USER SERVICES (Oi!/NDN-Opp)
Message shared via Wi-Fi Direct
Trusted Circle
USER (Android device)
END USER SERVICES (CM/NDN-Opp)
GEOLOCALIZATION – TEXT – TAG: FOOD
UMOBILE HOTSPOT (Laptop – NDN aware AP)
Interests FOOD
USER (Android device)
END USER SERVICES (CM/NDN-Opp)
Message shared via Wi-Fi Direct
Interests FOOD - FASHION
USER (Android device)
END USER SERVICES (CM/NDN-Opp)
Message shared via Wi-Fi Direct
Only NDN pkt (Interest and Data)
Non NDN "standard" messages used by NDN-OPP, such as pData will be encapsulated
Message shared via Wi-Fi Direct
Interests FOOD – EVENTS
USER (Android device)
END USER SERVICES (CM/NDN-Opp)

Figure 27: POC2 block diagram scheme

5.1. Integration

POC1

Figure 28 represents the UMOBILE hotspot and all the components that it integrates, such as KEBAPP module, the NDN service. The UMOBILE hotspot uses an OpenWrt OS, which is a lightweight Linux distribution. This assures compatibility with many of the market systems.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

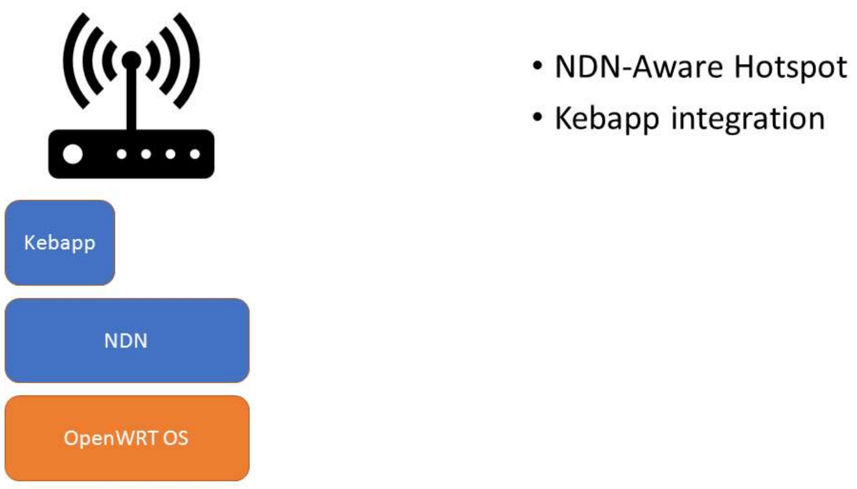


Figure 28: UMOBILE hotspot integration

This section explains how the integration of the service migration platform with the DTN framework can help to deploy services in an area partitioned from the main network (shown in Figure 29). The aim is to demonstrate how the service migration platform can be integrated with the DTN framework to mask network problems which is the goal of PoC1. This integration is further adopted from the demo presented at ACM ICN 2017. As presented in the block diagram of Figure 26 (PoC1), we assume that the authority user (e.g., Linux based machine) is at the local authority area and that the disaster area network is miles away and disconnected. The aim is to push a service (S) from the authority user machine and deploy it in the remote area (UMOBILE hotspot). We assume S to be a stateless service, e.g., a self-contained web server. To support this operation, we present an ICN-based approach to deploy dockerized services closer to end-users located in a network section impacted by a network partition event.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

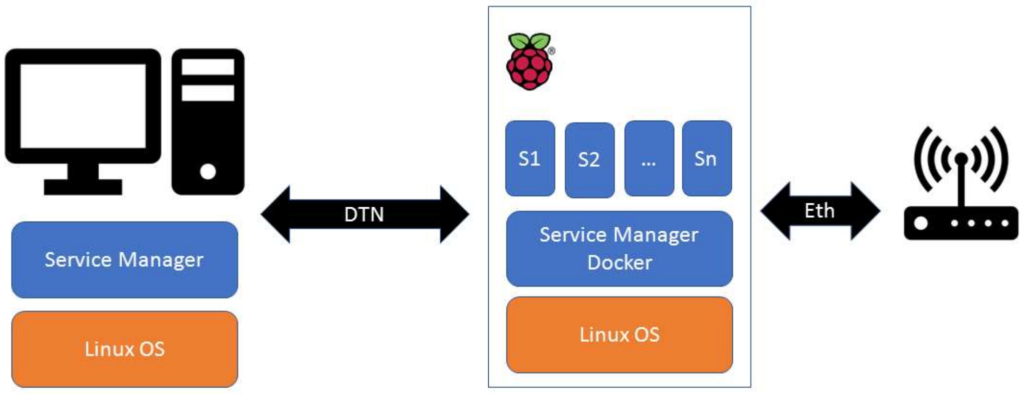


Figure 29:Service migration integration

To facilitate intermittent communications, we have implemented a special DTN face. This new face communicates with an underlying DTN implementation that handles intermittence by encapsulating Interest and Data packets in DTN bundles. Store-and-forward techniques are used by the underlying DTN implementation to reliably deliver the bundles utilizing standard DTN routing protocols. Upon reaching their destination, the bundles are decapsulated and the original Interest/Data messages are forwarded, resuming typical NDN communications. A possible deployment related to the PoC1 scenario includes a UMOBILE capable edge node (e.g., hotspot, service manager) that maintains a FIB entry towards the next edge node through the DTN face and one or more intermediate DTN mobile nodes that forward packets between the UMOBILE nodes, essentially forming a tunnel. By tunneling traffic through DTN, we create an alternative, reliable communication channel for NDN in situations where typical TCP and UDP faces would fail. We also enable data forwarding between two remote UMOBILE nodes by using different upstream and downstream intermediate (DTN-capable) routes/nodes, without any modification of NDN semantics. This can alleviate the fact that the NDN breadcrumb routing approach requires the same node that forwards an Interest to return the Data as well, inhibiting data delivery capabilities in such environments.

To support the operation in PoC1, we assume that both authority user machine and UMOBILE hotspot are equipped with the UMOBILE protocol stack. The authority user machine operates the service manager module while the UMOBILE hotspot runs the service execution function. Figure Hotspot-SM presents the abstraction of service migration integration for PoC1. The service manager (Linux based machine) communicates with UMOBILE hotspot (HS) through DTN tunneling. Notice that there is no physical connection between the service manager and the UMOBILE hotspot, our

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

solution exploits the DTN functionality by using a DTN node (e.g., Android phone) to carry the message between two nodes in different location (local authority and remote network). Android phone is a physically mobile, DTN-enabled node that can travel backwards and forwards between service manager and the HS. In fact, any smart device attached to a UAV or any other means of transport can replace the Android phone, as long as it supports a DTN framework (e.g., IBR-DTN).

In the last step of PoC1, shown in Figure 30, an emergency service is migrated and deployed in a UMOBILE hotspot placed in the rescue area. The devices and functions involved in the emergency service migration and user access are detailed in the following, with the correspondent interfaces between them explained.

- Service manager (Server): centralized entity that is able to migrate any service to a service execution function.
- Service execution Function (Raspberry Pi): a service execution function that is capable of instantiating services out of Docker images migrated by the Service Manager.
- UMOBILE Hotspot (Fonera): Provides network connectivity to users and advertises available service sin the service execution function.
- KEBAPP User (Smartphone): Smartphone user with a KEBAPP-enabled app installed.

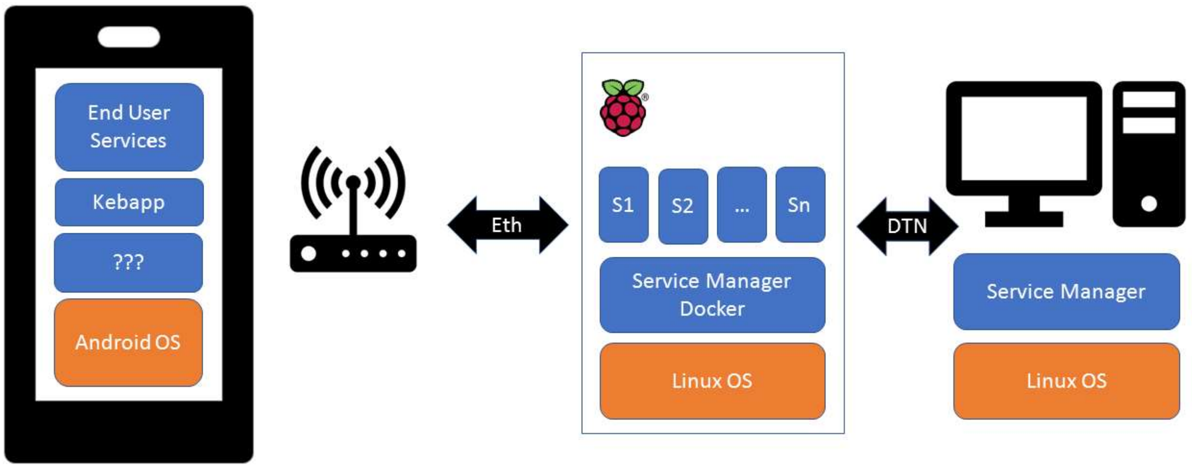


Figure 30: KEBAPP integration

UAV scenario

In order to allow a better comprehension of the complete data transmission process, the demonstration developed by UAVs is intended to support the elaboration and schematization of the POC1, separating however the diagram scheme represented above into the two main directions of data transfer. Indeed, POC1.1 illustrates the first stage of the scenario, where the message source begins with the user, which possesses an Android device and needs to deliver certain information (which can be, for instance, an emergency request) to the local authority.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Then, on the other hand, POC1.2 consists in the reply to that message that is expected to achieve a specific rescue team, depending on the emergency case.

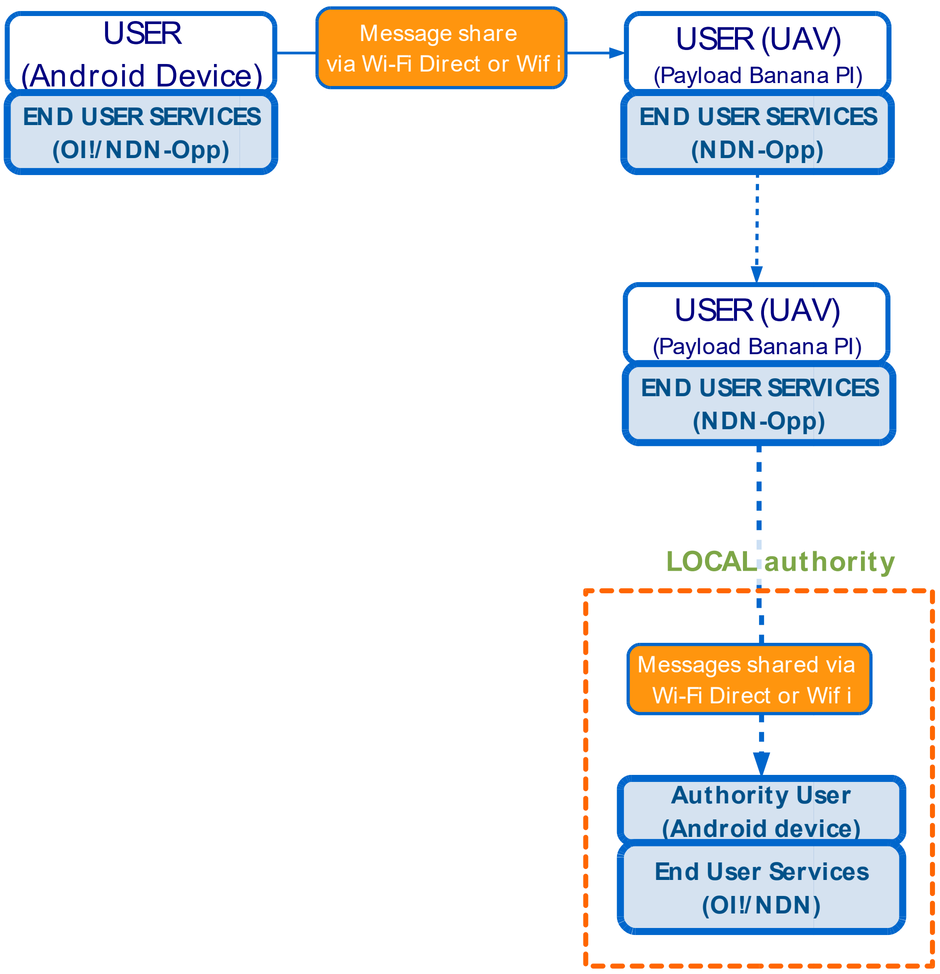


Figure 31: Block diagram for the support tests to POC1.1

Figure 31 represents the process of data transmission from the user to the local authorities – the process implied in POC1.1. Using an Android device (such as a smartphone), a message is transmitted and shared with the UAV that is flying in the surrounding area that originally did not have the required connectivity to cover the region. This message is sent via Wi-Fi Direct and is received directly by the UAV, which is integrated with the appropriate tools to capture the signal and receive the information. For that, this UAV (can be either an AR3 or VR1) is indeed equipped with the correct devices, in which it is included the Banana PI – the embedded Android device that is able to substitute the smartphone and allows an easy integration into the UAV (using power connections and connection data to Wi-Fi device – such as Ethernet). This incorporation is also ensured to be low-weight, given the normal payload limitations of the UAV. Operating via NDN services, the Banana PI is a fundamental part of the payload responsible for receive and send data/messages, turning the UAV into a UMOBILE hotspot with data mule. After receiving the information, the UAV continues the mission and, as it gets closer to the connectivity range of the Ground Control Station (GCS) – which is installed in the local authority command station/center –

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

will then transmit the urgent message to the responsible responder services. This message is shared via Wi-Fi Direct or Wi-Fi, and the local authorities can obtain the request initially sent by the user (the civilian that eventually suffered or detected an emergency).

After the obtainment of the message, the local authority may have the responsibility to reply to the initial request depending on its content. This replying process is represented by the schematization of the POC1.2, illustrated in Figure 32.

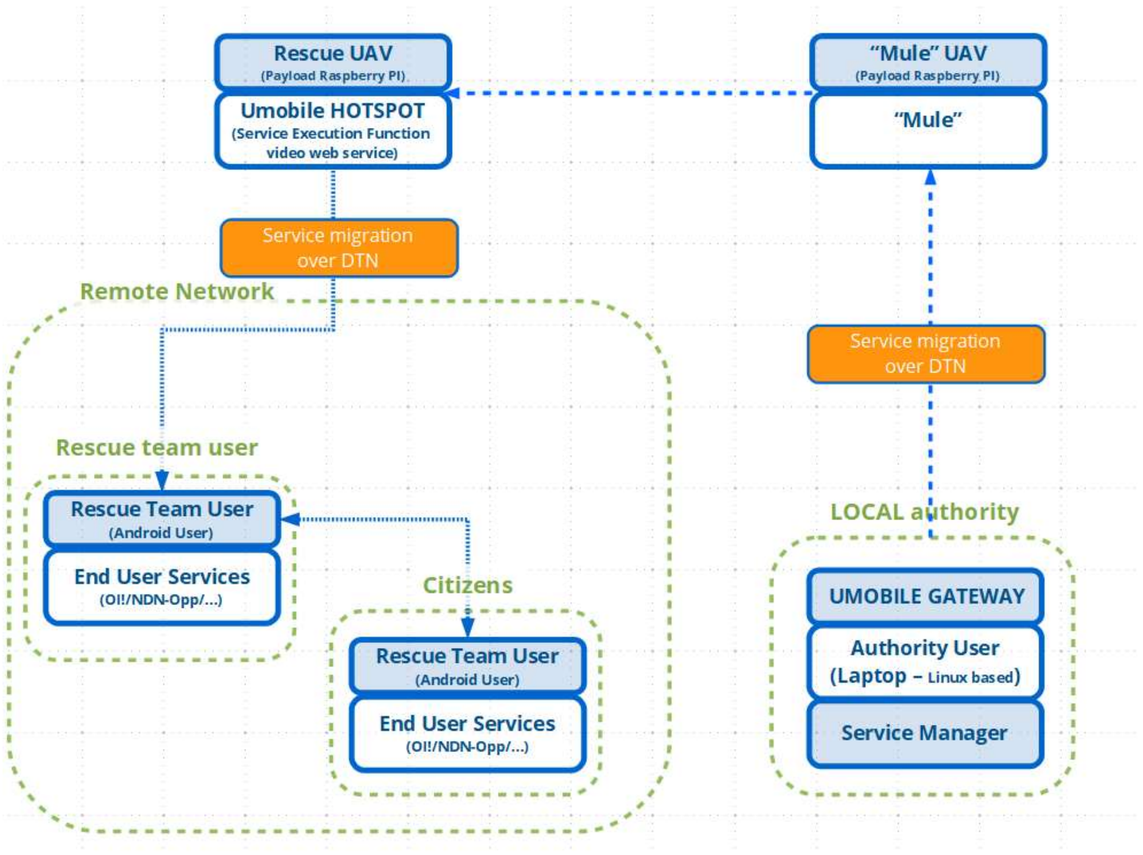


Figure 32: Block diagram for support tests to POC1.2 (service migration)

POC1.2 begins as the authority users, also known as the service managers, have the awareness of the situation faced by the initial user, and decide to take the appropriate measures to intervene. With the use of Linux based laptops, it is then established the over DTN transmission with certain rescue indications to the UAV. This is achieved with the evaluated service migration, using this time the UAV as an UMOBILE hotspot and data mule – that might include the transmission of heavy data such as videos or images, between others. This UAV (can also be either the VR1 or AR3) has, by its turn, the appropriate devices for the replying scenario, which consists in the inclusion of the Raspberry PI. This small device is what allows the UAV to be used as a hotspot and constitutes the service execution function. UMOBILE hotspots can also be deployed in isolated UAVs, providing local services and DTN capabilities, or in connected UAVs (through a data uplink), providing connectivity services. Due to the size of the data that needs to be transmitted, it is possible for this process to last longer, on the order of 10 minutes. Following this step, the UAV is then ready to communicate with the remote network, which includes the rescue team services. The same information is delivered to the rescue team users on the ground, using the same DTN

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

service migration, resulting consequently in a process with nearly the same time duration as the anterior, since the same data must be acquired by this end user services (rescue team that has Android devices). Once the information is received on the ground, the rescue team is then able to perform the assistance mission in the emergency area and support the first user initial request. This process eventually leads to a connection between the rescue team and the citizens that need to be contacted (for example, to send indications to leave a certain risky area). Those citizens, that can include the initial user that sent the emergency request, are also able to be represented as end users if they also possess Android devices (such as the smartphone), and may even reciprocate their reply to the rescue team when needed.

The block diagram in Figure 32is intended to demonstrate the data migration through the whole set of systems that are related to the UMOBILE – from the user, passing through the UAV devices and achieving the authority services that, by its turn, can eventually proceed with the rescue and interact with the citizens.

POC2

In the context of PoC2 integration, shown in Figure 33, we shall consider two applications: Oi! and the UMOBILE End-user Service (UES), which integrates the CM (Context, node availability, centrality, and similarity) , NDN-Opp (Direct Communication) as well as the routing module DABBER. These are in fact 3 independent Android applications which are being developed in Java. The module DABBER runs in Java/C++. In terms of software all these software modules are independent and running in the same device (end-user device).

The Oi! Application requires NDN-Opp (dependence) to run. For the demo it will rely on DABBER to perform message exchange across multiple hops (even though it is not a dependency). The DABBER routing module shall make use of an interface of the CM currently being developed to obtain the cost for node availability; node centrality, as well as node betweenness. This is a public interface, which other UMOBILE software modules can rely upon to acquire the specific node costs.

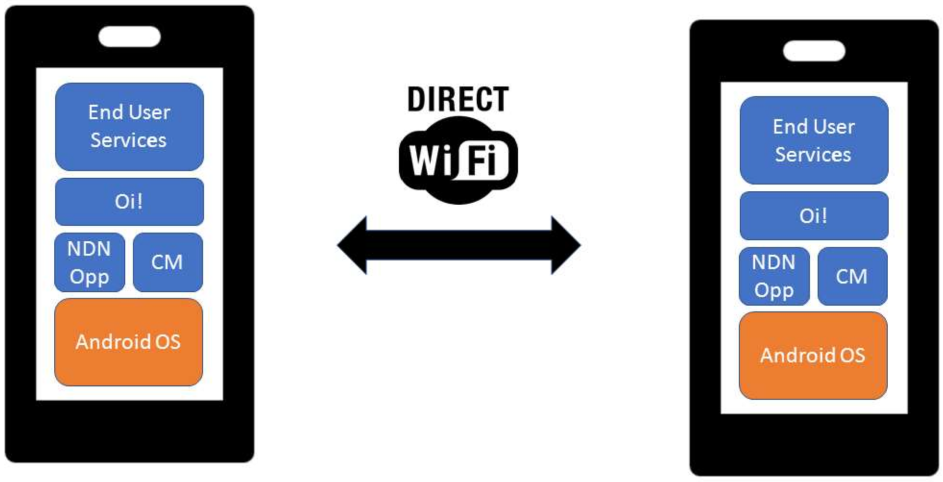


Figure 33: Oi!/NDN-Opp/Contextual Manager integration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

5.2. Validation

For a thorough validation of the POCs please refer to D5.4 [3]. Here we focus on the overall system validation as a basis for the final demonstration.

POC1

For the demonstrations of service migration and app sharing please refer to Sections 4.6.2 in this deliverable and 4.4.2 in D5.1.

For the demonstration of the NDN routing support in opportunistic environments POC1 corresponds to a simplification of POC2. We shall consider only the push model, to exemplify the benefits of such a model in a realistic environment. As illustrated inFigure 34, Oi!/DABBER Is installed in end-user devices. User (1) sends a message via Oi! to a user in the Local Authority premises. The message is opportunistically transmitted via the bikers (2)-(3), as illustrated inFigure 34, until it reaches the infrastructure network of the local authority (4). Here, we demonstrate how messages can be routed in opportunistic environments, with NDN support.

Afterwards, and within the rescue team, we demo how the different elements can exchange messages via Oi!/DABBER directly, even with intermittent connectivity (even if there is, at an instant in time, no end-to-end path between the involved source and destination).

POC2

The two different models shall be tested under different sub-scenarios of the POC2 demonstration. Figure 34 shows the topological scheme for the demo. In Lisbon, the main tests shall be developed in the COPELABS facilities and wireless testbed. This testbed is interconnected to the NDN worldwide testbed. Hence, validation between COPELABS and the UMOBILE Lab in AFA, Italy is also expected. Validation over the Internet shall be done to provide an understanding on interoperability features.

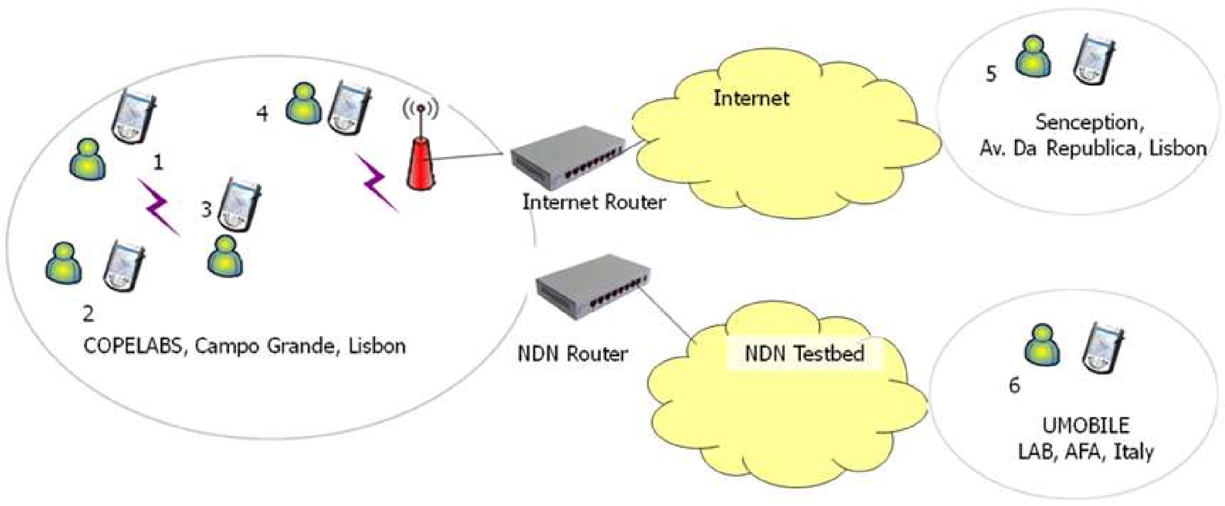


Figure 34: POC2 scenario

Then, the two different data dissemination models – push and pull – shall be demonstrated and validated under three different situations: i) Direct device-to-device communication; ii) Local communication with support for intermittent connectivity; iii) end-to-end communication (both via the Internet and directly via NDN).

Direct communication corresponds to sending messages between devices 1 and 2 and 3, which are in eachothers range. Hence, this first approach simply compares the push vs. Pull model.

Support for intermittent connectivity shall be demoed via the exchange of data from device 1 to device 4. This device is in the range of 3 but not in the range of 1. We shall show here the difference in performance between regular flooding against our contextual-based approach (DABBER).

End-to-end communication shall be performed both via plain NDN (between COPELABS and a device in the UMOBILE Lab) as well as to Senception.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

6. Conclusions

The main objectives of this deliverable are to provide the validation setup for the system and its individual components and to describe the evaluation of the platform through simulations.

This deliverable is a second and final version of D5.1. Here we have updated the information provided in D5.1 and have included conclusions on the validation and simulation works that the consortium has performed during these months. Moreover, thanks to the work done regarding integration in this document we are able to provide figures that explain the final deployments that we will do to demonstrate different use cases with UMOBILE system.

For that purpose, first we have presented the functional blocks of the UMOBILE platform and each of the network and end-user devices that are part of the platform. Based on these two types of devices we have provided a detailed description of each of them explaining also a full image of the whole UMOBILE platform.

Secondly, in Section 3 we briefly describe the simulation methodology and the tools that are being used to evaluate the different parts of the solution through simulation, as they were widely presented in D5.1. Several blocks of the UMOBILE platform have been validated through this mean.

Next, in Section 4 we have assessed and updated the validation of UMOBILE platform by explaining the validation methodology of each of the UMOBILE components individually. Besides, we have mapped each of the components with one of the use cases defined previously in the project.

In Section 5 we have updated the description of the functional blocksfor each of the POCs and also presented the integration and validation of the UMOBILE system platform among different UMOBILE components.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

7. References

[1] UMOBILE Project, “D.5.1–Validation methodology and evaluation report (1)”, January 2017

[2] UMOBILE Project, “D.5.3 - Proof-of-Concept (1)”, January 2017

[3] UMOBILE Project, “D.5.4 - Proof-of-Concept (2)”, November 2017

[4] UMOBILE Project, “D.3.3 - UMOBILE ICN layer abstraction initial specification”, February 2016

[5] UMOBILE Project, “D.4.1 –Flowlet Congestion Control (1)”, January 2016

[6] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, “A survey of informationcentric networking research,” IEEE Communications Surveys Tutorials, 2014.

[7] I. Psaras, K. V. Katsaros, L. Saino, and G. Pavlou, “Lira: A location independent routing layer based on source-provided ephemeral names,” arXiv preprint arXiv:1509.05589, 2015.

[8] V. Sourlas, P. Flegkas, and L. Tassiulas, “A novel cache aware routing scheme for information-centric networks,” Computer Networks, 2014.

[9] D. Trossen and G. Parisis, “Designing and realizing an informationcentric internet,” IEEE Communications Magazine, 2012.

[10] L. Saino, I. Psaras, and G. Pavlou, “Hash-routing schemes for information centric networking,” ACM ICN, 2013.

[11] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM 2.0: A new version of the NDN simulator for NS-3,” NDN, Technical Report NDN-0028, 2015.

[12] A. Keranen, J. Ott, and T. Karkkainen, “The ONE Simulator for DTN Protocol Evaluation,” in SIMUTools, 2009.

[13] UMOBILE Project, “D.3.4 –UMOBILE ICN layer abstraction final specification”, July 2017

[14] “NS-3, A Discrete Event Simulator.” http://www.nsnam.org.

[15] “ndnSIM: NS-3 based Named Data Networking (NDN) Simulator”, http://ndnsim.net/2.3/, last accessed 21/01/2017

[16] “The One, The Opportunistic Network Environment simulator,” https://akeranen.github.io/the-one/, last accessed 21/01/2017

[17] Upeka De Silva, Adisorn Lertsinsrubtavee, Arjuna Sathiaseelan, Carlos Molina-Jimenez, and Kanchana Kanchanasut. 2016. Implementation and evaluation of an

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

information centric-based smart lighting controller. In Proceedings of the 12th Asian Internet Engineering Conference(AINTEC '16). ACM, New York, NY, USA, 1-8.

[18] Llorenç Cerdà-Alabern, Axel Neumann, and Pau Escrich. 2013. Experimental evaluation of a wireless community mesh network. In Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems (MSWiM '13). ACM, New York, NY, USA, 23-30.

[19] UMOBILE github repository: https://github.com/umobileproject

[20] UMOBILE Project, "D.4.4 -Set of Qos intefaces and algorithms", July 2017

[21] UMOBILE Project "D4.2 – Flowlet congestion control (2)", July 2017

[22] G. Carofiglio, M. Gallo, and L. Muscariello. Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks. In ACM ICN workshop, 2012.

[23] Klaus Schneider , Cheng Yi , Beichuan Zhang , Lixia Zhang A Practical Congestion Control Scheme for Named Data Networking ACM ICN Conference 2016

[24] UMOBILE Software Oi: https://play.google.com/store/apps/details?id=com.copelabs.android.oi

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

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124