Action full title:

Universal, mobile-centric and opportunistic communications architecture

Action acronym:

UMOBILE

Deliverable:

D5.1 “Validation methodology and evaluation report”

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 the validation setup for the system and its individual components of the UMOBILE architecture. The report will also include the results of the detailed evaluation of the platform through simulations. A final validation and evaluation report will be provided by the end of the project (month 36).

Deliverable Number-Title	D5.1 Validation Methodology and Evaluation Report
WP Number	WP5
WP Leader	FON
Task Leader (s)	Alberto Pineda (FON)
Authors	**COPELABS**: Paulo Mendes **ATHENA R.C**: Sotiris Diamantopoulos, Christo-Alexandros Sarros, Vassilis Tsaoussidis **UCL**: Ioannis Psaras, Sergi Rene **UCAM**: Adisorn Lertsinsrubtavee, Carlos Molina-Jimenez **SENSECPTION:** Rute Sofia **TECNALIA**: Iñigo Sedano **TEKEVER**: Francisco Almeida **AFA**: Angela D'Angelo, Francesco Amorosa, Giammichele Russi **FON:** Alberto Pineda, Pablo Salvador
Reviewer	Alberto Pineda, Pablo Salvador
Contact	alberto.pineda@fon.com
Due date	M24: 31/01/2017
Actual date of submission	04/04/2017

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

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	11/11/16	Initial template with partners' input
Version 0.2	14/11/16	Circulate among partners first draft proposal and ask for extra input
Version 0.3	07/12/16	Revised first version based on feedback
Version 0.4	21/12/16	Added new sections
Version 0.5	13/01/16	Added some partners' contributions
Version 0.6	23/01/17	Finalized preliminary version of the deliverable and circulated for internal review
Version 0.7	30/01/17	Finalized preliminary version of the deliverable and circulated for internal review
Version 1.0	31/01/17	Final version after reviewers' comments
Version 1.1	31/03/17	Updated addressing interim review feedback
Version 2.0	04/04/17	Final version

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 Tasks 5.1 "**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 focuses on the description of the UMOBILE system components from a functional point of view. Furthermore, this deliverable presents the validation setup of the overall system. 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.

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

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 deliverable D5.3 [1]. The aim of this document is to present each of the individual components of the UMOBILE platform, their features and their corresponding validation and evaluation.

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. Finally, the conclusions are drawn in Section 5.

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

2. UMOBILE Platform

In this section we describe the UMOBILE platform and where each of their elements are located. In addition, we outline all the elements of the platform and explain them.

2.1. Overview

Figure 1 is well-known as it has been used during the whole project to illustrate the UMOBILE platform. It shows a high-level description of the UMOBILE platform in a specific scenario. There are two domains: the UMOBILE domain; this is the NDN-DTN area where the UMOBILE platform provides services to end users, and the Internet domain where we can find IP communications. Following Figure 1, thanks to UMOBILE Hotspots and UAVs and thanks to D2D communications and WiFi, the platform provides access to Internet based on NDN-DTN paradigms. UMOBILE routers create an NDN network. UMOBILE Gateway allows reaching Internet thanks to the de-encapsulation of NDN traces. The service manager is in charge of the service migration to UMOBILE hotspots.

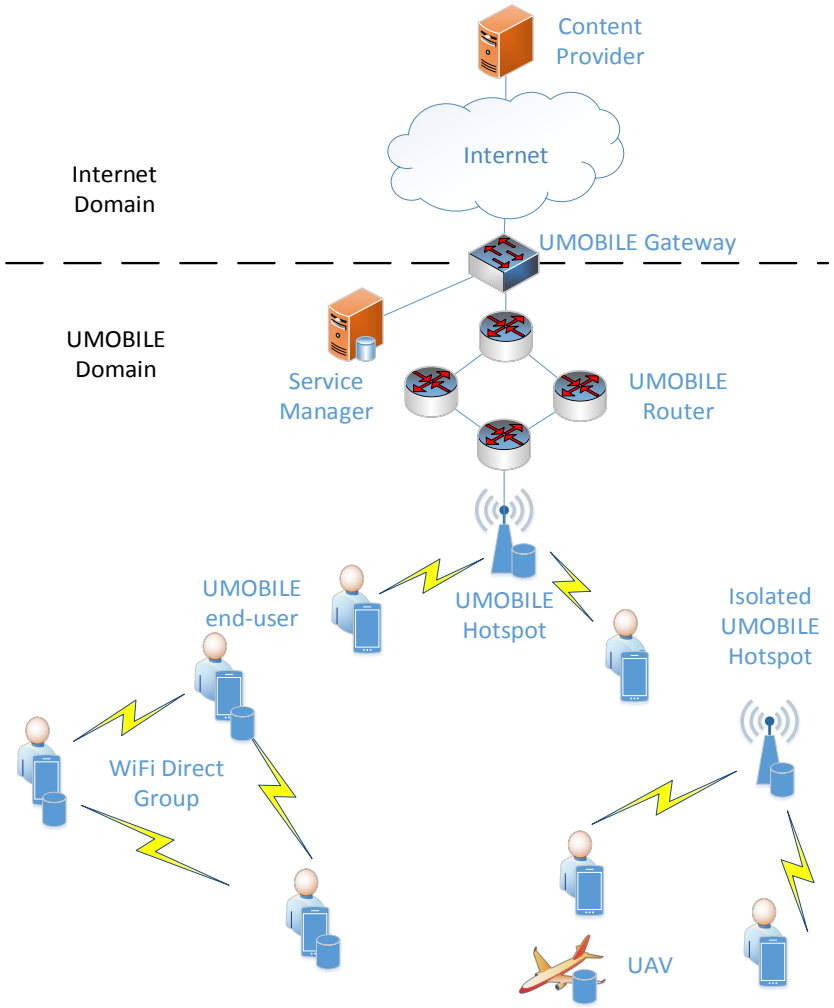


Figure 1: Overview of the UMOBILE platform

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

Figure 2 extends the concept of Figure 1 as it contains different scenarios where the UMOBILE platform can operate including the one that Figure 1 describes. Moreover, it adds a new actor, the ISP, locating its premises in the UMOBILE ecosystem. In order to provide compatibility with existing mobile user devices, we devise the UMOBILE platform as a layer working on top of IP, where IP is used as a network enabler but only on a hop-by-hop basis and is linked to the wireless connectivity.

In Figure 2 we illustrate three different ways of connecting the area where UMOBILE is providing service to end-users with UMOBILE services. As, in UMOBILE, NDN is encapsulated on top of IP, the first way of connecting the area is through the IP network of the ISP and through the internet. The traces reach ISP's premises and they are de-encapsulated in the UMOBILE Gateway. A second way is directly through an entire NDN public network. The last one corresponds to Figure 1 and proposes to build an NDN network thanks to UMOBILE routers.

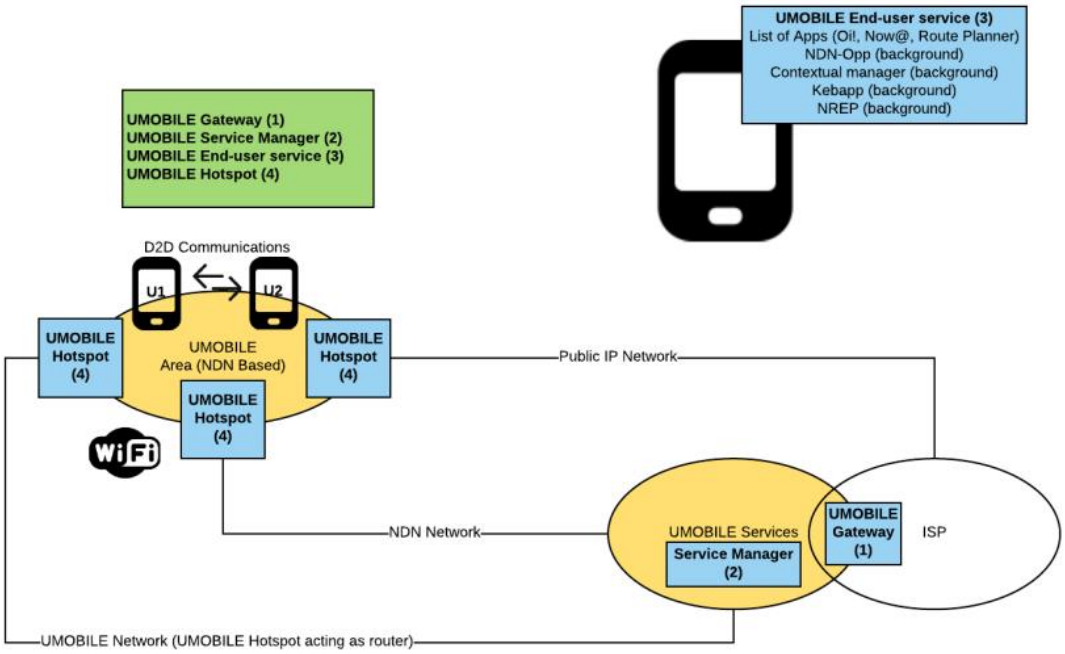


Figure 2: Functional blocks of the UMOBILE platform

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

The actors involved in the UMOBILE architecture include:

- **UMOBILE Hotspots**: 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. Some of these hotspots will be isolated access points (APs) providing specific services
- **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.

The demonstrators, detailed in D5.3 [1], highlight use cases covered by the UMOBILE platform, which is characterized by a WiFi network set-up in which some UMOBILE hotspots are collocated with a gateway.

UMOBILE is being developed as a modular software architecture, where some modules may or may not reside in a specific hardware element. For instance, we envision that the Service Manager will only run on end-user devices, while the contextual manager will be developed to run in end-user devices, as well as APs. Nevertheless, modules such as the Routing Module or the Forwarding Strategies will be applicable in different locations.

2.2. UMOBILE Hotspots

The UMOBILE hotspot is deployed in the fixed part of the network or on-board mobile nodes (e.g., UAVs), enabling a multitude of network-related operations. In particular, these devices employ the UMOBILE platform to cache content, to execute services, to allow for granular QoS and QoE application interfaces as well as to ensure data delivery even under adverse network conditions. UMOBILE-enabled hotspots collect and relay relevant information (e.g., alert messages, instructions from emergency authorities), host some instantiated services or store collected data, check its validity and perform computational functions (e.g. data fusion) to increase the value of the information to the civil authorities.

2.2.1. UMOBILE UAVs

UMOBILE-enabled UAV devices collect and relay relevant information and connect two isolated areas, 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 area covered and enable communication with other

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

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.

The selected equipment for UMOBILE is the AR4 from TEKEVER presented in Figure 3. The AR4 Light Ray is a small tactical, hand-launched UAV for ISR (Intelligence, Surveillance, and Reconnaissance) targeting the defence market. With an endurance of 2 hours and a range of 20km it is designed essentially for over-the-hill operations. It weighs about 5kg and is capable of carrying up to 1kg payload, having been validated and battle proven by multiple security and military forces.

Figure 3: UAV setup

The AR4 can carry multiple options for Electro-optical, sensors, up to 4K definition, which could enable transmission of high-resolution image and video that could help in the management of the scenario.

In order to assess UAV use as hotspots, the WiFi signal strength in the ground will be measured when the UAV is flying in different altitudes and locations. The transmission of short messages through the UAV will be the main objective to be achieved. However, if possible, the UAV could be used to permit the transmission of other data, namely in coordination with the data link used for telemetry and control.

2.3. UMOBILE Gateway

UMOBILE-enabled gateways provide interconnectivity between any opportunistic UMOBILE device and the fixed IP network by translating interest packets to HTTP requests and vice versa. Such devices can be employed by service and content providers

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

to act as repositories, being able to store data received through the IP network and then share it over the UMOBILE network (or vice-versa) upon request. In the framework of the UMOBILE project, the focus is placed only on service and content sharing over the UMOBILE network.

UMOBILE Gateway can be part of the infrastructure of service/content providers or civil authorities, being employed as repositories supporting both the IP and the UMOBILE part of the network. In particular, UMOBILE Gateway can be utilized by providers as their point of entry into the UMOBILE network, connecting their legacy IP-based infrastructure with other UMOBILE nodes, as depicted in Figure 2. This point of entry is normally expected to reside in the provider's premises. Depending on the deployment model (an option is to follow the one devised in [5]), though, any UMOBILE hotspot can act as Gateway, if it is equipped with an IP interface and sufficient storage and processing power. The gateway assumes its role by providing universal access to content or services located in both the host-centric and information-centric domain. For example, when a service exists in the IP domain that can be deployed in the UMOBILE part of the network, it can be "fetched" by the UMOBILE gateway – with the help of the Service Manager – and stored there. It is then made available to the UMOBILE users by the repository, who can choose to download and deploy it as they like. In essence, the gateway provides a common pool of shared services and information between the two domains, assisted by the service migration mechanisms.

As a UMOBILE node itself, a UMOBILE Gateway is employing the full UMOBILE platform, being, thus, able to forward data over the DTN interface if required.

2.4. UMOBILE Service Manager

UMOBILE is being developed as a modular software architecture, where some modules may reside or not in a specific hardware element. For instance, we envision that the Service Manager will only run on end-user devices, while the contextual manager will be developed to run in end-user devices, as well as APs. Nevertheless, modules such as the Routing Module or the Forwarding Strategies will be applicable in different locations.

The Service Manager is a piece of software that implements all the functionality that the Service Provider needs to deploy his services, including disk space to store both services and compressed images. In Figure 1, the Service Manager is strategically deployed on a computer directly connected to the Gateway. This deployment simplifies the task of transferring compressed service images from the global Internet to the UMOBILE Domain.

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

2.5. UMOBILE End-User Service

In this section we specify the different software components that are part of the UMOBILE platform.

The UMOBILE End-User Service (UES) integrates different software modules that allow UMOBILE services to be delivered with adequate levels of QoE and QoS to end-users. The UES integrates the following:

- **UMOBILE Background services** that assist in better data transmission, and service delivery with adequate QoS and QoE levels.
    - **KEBAPP,** to enable users to share the applications present in their mobile devices.
    - **NREP,** name-based replication system limited by time and space within a certain geographic area and with specific life expectancy.
    - **NDN-OPP**, to assist opportunistic routing (device-to-device communication) thus taking advantage of opportunities and interest/contacts in proximity, to exchange data.
    - **Contextual Manager**. The contextual manager is a UMOBILE module that resides on end-user devices and that captures both external (roaming) and internal (usage) data, to assist a better network operation (e.g. opportunistic routing; priorities in terms of resource management; interest match in terms of data dissemination).
- **List of UMOBILE native applications**. In addition to services migrated, UMOBILE integrates a set of novel applications such as Oi!, Now@, RoutePlanner. Via the UMOBILE End-User Service, the user can then select specific applications, which support natively intermittent connectivity.

The UES is being devised in a modular way, to ensure that it can be run in regular end-user devices such as smartphones, tablets, laptops and, even, Raspberry PIs.

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

In this section, we describe the different simulation tools that are used to evaluate the efficiency of the different UMOBILE components. As a fully integrated UMOBILE platform does not exist at this stage of the project, 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. We note that the current deliverable also includes module evaluation reports provided by real deployments, but these are not the focus of this section; instead, we focus only on simulations. 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 leveraged 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. Below, we provide some basic information about the simulators and enumerate the specific UMOBILE protocols being evaluated on each one, while stating the reasons for the particular choice.

3.1. ns3 and ndnSIM

Network Simulator 3 (ns3) [19] is a discrete event network simulator primarily used for research and education, one of the most extensively used in the research community to test new network protocols. It is rooted on an open-source project started in 2006. It is the successor of the popular ns2 simulator that is no longer being developed, although it was not designed to be backwards compatible.

NdnSIM is an extension to the ns3 simulator [21], implementing the basic components of the Named Data Networking architecture in a modular way. The design of ndnSIM follows the philosophy of network simulations in ns3, which devises maximum abstraction for all modelled components. This way, diverse and flexible experimentation over NDN is achieved allowing for different deployment needs (NDN over link-layer, NDN over IP etc.) which would be hard to test on current testbeds.

ns3 is used to evaluate the In-Network Resource Pooling Protocol (INRPP). This decision is made as INRPP is a congestion control protocol primarily used for wired networks and ns3 is widely used for wired network topologies. There is also an extensive knowledge of the ns3 simulator at UCL, so the development and simulation process was greatly sped up by using the particular tool.

NdnSIM evaluates the Opportunistic Off-path Content Discovery protocol (OOCD). Since OOCD is an improvement of the NDN architecture, a simulator designed for Named Data Networks is needed. To the best of our knowledge, ndnSim is the only simulator designed to simulate NDN networks. The choice is supported by the fact porting the ns3 code to a

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

real Linux NDN implementation is really straightforward when using ndnSIM. This is because the ndnSIM extensions required to support the proposed content discovery mechanism involves enhancements to the ndn-cxx library and the NDN Forwarding Daemon (NFD) components, which are exactly the components used in the real NDN implementation – also used in ns3 along with added wrapper libraries.

3.2. ONE

The Opportunistic Networking Environment (ONE) simulator [22] 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. The ONE simulator offers an extensible simulation framework itself supporting mobility and event generation, message exchange, DTN routing and application protocols, a basic notion of energy consumption, visualization and analysis, interfaces for importing and exporting mobility traces, events, and entire messages.

In order to evaluate KEBAPP and NREP (the smart routing module might be also evaluated by these means, but this is still under study) we use ONE for its simplicity, flexibility and mobility traces provided. In this case, we do not require a full stack network simulator, like ns3, to evaluate NDN networks, but a simple simulator to evaluate our proposals with different traces for opportunistic networks. The ONE simulator is designed in a modular fashion, enabling extensions of virtually all functions to be implemented using well-defined interfaces, which allowed us to extend it to develop NREP and KEBAPP approaches.

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 we present the validation of the different components of the UMOBILE network devices and end-user services corresponding to software modules by means of experiments or simulations.

4.1. In-Network Resource Pooling Protocol (INRPP)

4.1.1. Description

UMOBILE is developing in task 4.1 of WP4 a new congestion control protocol named In-Network Resource Pooling Protocol (INRPP) aimed at improving congestion control in the fixed network by using caches as temporary custodians to deal with congestion locally, and using backpressure techniques to slow-down senders when necessary. D4.1 [3] 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 will be completed in D4.2, including the development of the Flowlet Congestion Control to be used in the UMOBILE domain.

In this first version of INRPP detailed in D4.1 [3], we assume that IP network routers possess caches, which are essentially large (i.e., bloated) buffers that act as custodians for the data transferred along the network paths. Data senders push data in the network in an open-loop, processor sharing manner, using the typical TCP protocol, but using a rate-based congestion control set at the available rate at the sender interface. Data is stored in in-network caches using a FIFO policy when it hits bottleneck links and progressively moves (from one cache to the next) towards the destination in a hop-by-hop manner. At the same time detour paths are exploited to move data faster towards the destination. In case of increased demand, the system enters a closed-loop, hop-by-hop backpressure mode in order to avoid packet loss. When the backpressure reaches the last hop (the sender), the TCP protocol adapts its rate to the available throughput at the bottleneck of the network. In-Network Resource Pooling is, therefore, realised in terms of both in-network storage and in-network detour decisions to exploit unused bandwidth in network subpaths. In that sense, feedback signals stay local (instead of having to propagate e2e) and move hop-by-hop together with the data. Per-flow fairness is assured at routers indexing the cached data by flow and pulling packets from the cache to be sent through the output interface from a different flow each time using a round-robin scheduler. The resulting INRPP eliminates packet drops, completes flows faster, allows senders to push all data of a flow faster inside the network and responds rapidly to changing network conditions. Our results show that flow completion time and fairness

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

improves by as much as 50% compared to RCP, MPTCP and TCP, under realistic network conditions.

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, 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.

4.1.4. Validation & KPIs

We next engage in a detailed investigation of the performance of the proposed INRPP 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 [19], 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), where communications take place in the Internet domain between central servers and service instances placed in the edge of the network.

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 4. 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

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

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.

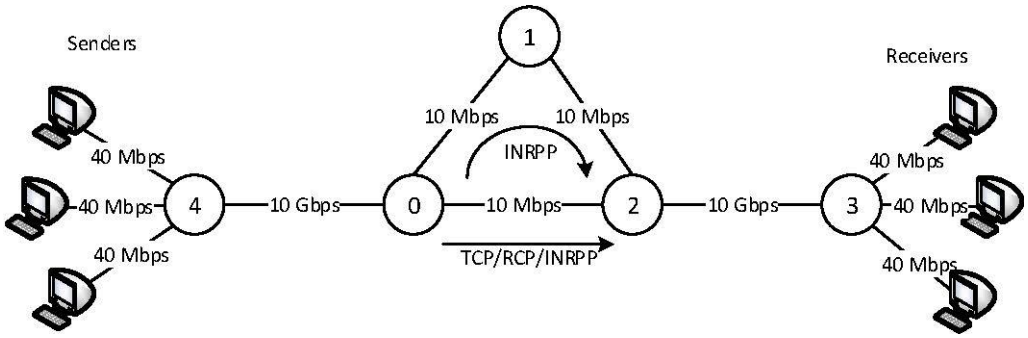


Figure 4: Detour simple scenario

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

In Table 1, we 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 5, 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 4. 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

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

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.

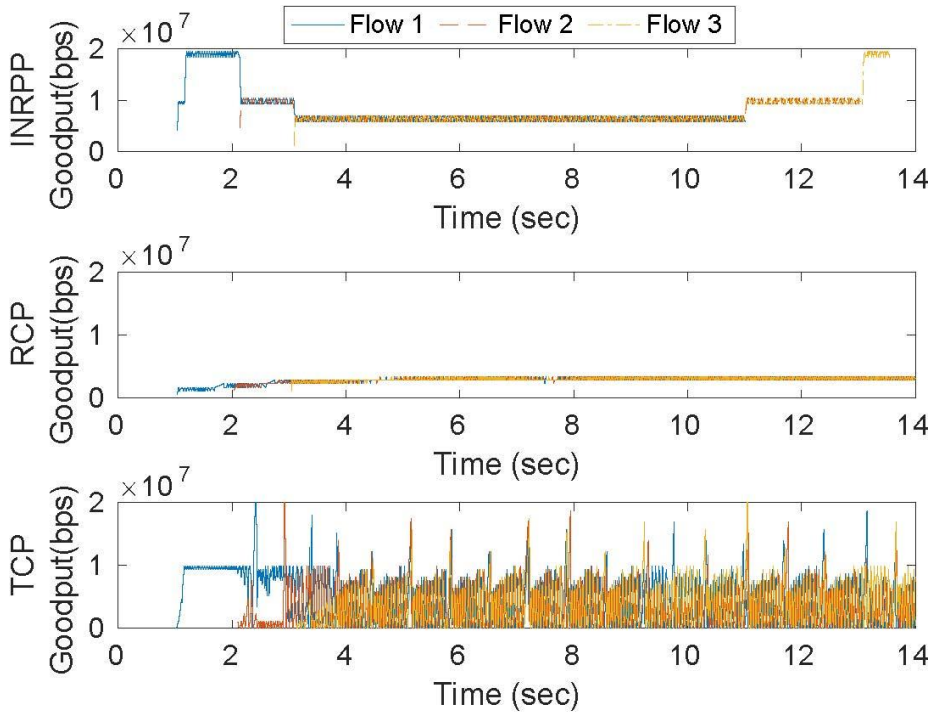


Figure 5: Receiver goodput

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

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

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.

Multi-homed topology

In this second scenario, we aim at evaluating INRPP in a multi-homed scenario where we can compare it with a multipath transport protocols, such as MPTCP. MPTCP can exploit more than one path and establish multiple sub-flows to take advantage of available sub-paths. It is, however, constrained to end-host multi-homing and can therefore, take advantage of e2e paths only. We evaluate the scenario depicted in Figure 6 where two paths can be used in parallel when using MPTCP (link 0-1 and link 2-1). MPTCP senders are multi-homed and are connected to node 0 and 2 at the same time, while the MPTCP receivers are single-homed and connected to node 1. MPTCP users establish two sub-flows, one for each pair of source, destination IP addresses. In addition, we also evaluate INRPP/TCP/RCP connecting the senders to node 0 and the receivers to node 1 only. This way, TCP and RCP will use only the path across link 0-1 (the shortest path) and INRPP will use this path as the main option, but will also be able to use the detour paths 0-3-1 and 0-2-1. The network parameters are the same as in Section 4.1, however, here, we increase the cache size to 12.5MB (size proportional to the link bandwidth equivalent to 10 seconds of traffic); the end-points have 100 Mbps links.

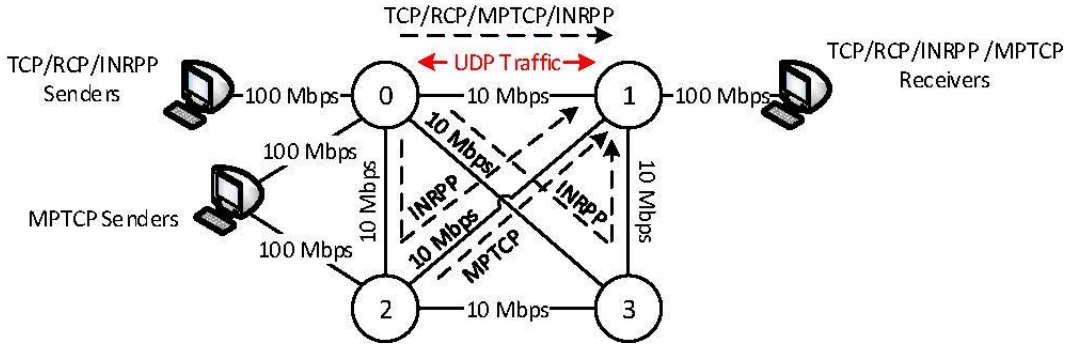
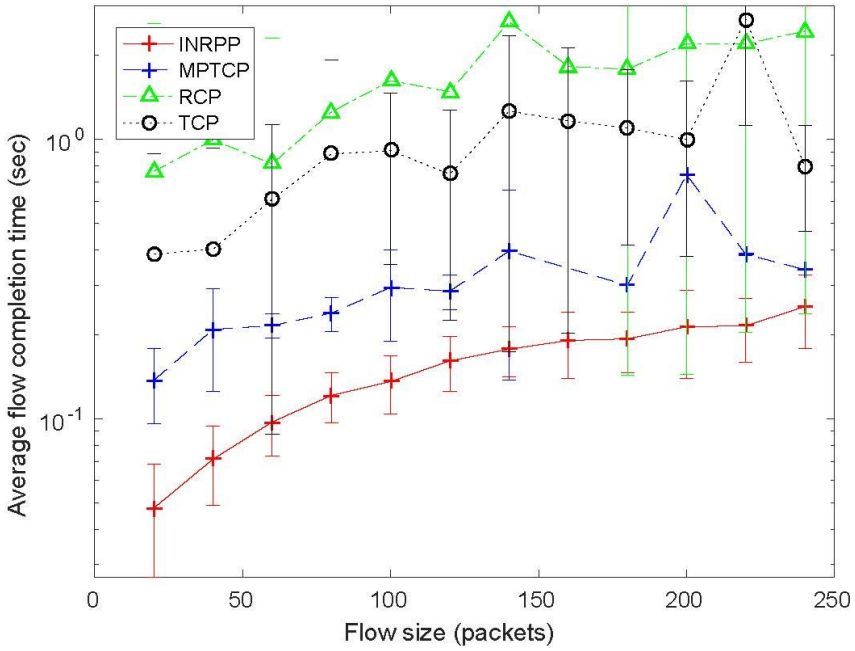


Figure 6: Multi-homed topology scenario

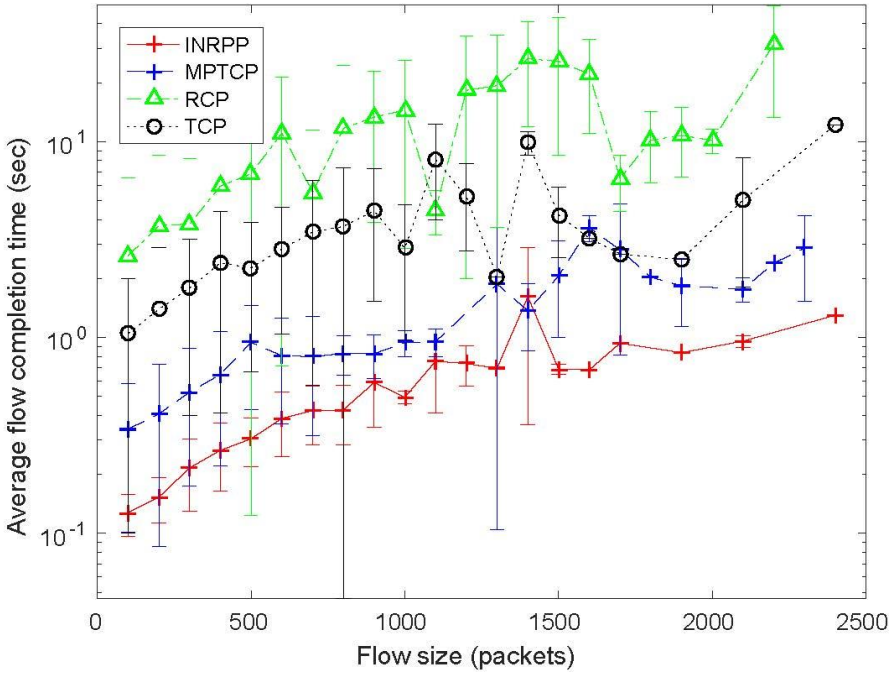
We evaluated this scenario using a Poisson Pareto Burst Process (PBPP) to model Internet traffic. We used 1000 flows with Poisson arrivals with a λ rate determined by the offered load of the network ρ, where $\rho = \lambda \times E[L]/C$ ($E[L]$ is the average flow size and C the capacity of the link), that we set to 0.9. Flow sizes are Pareto distributed with shape equal to 1.2. Note that this scenario is beneficial for MPTCP because it is a symmetric scenario. MPTCP behaviour can be worse when using highly asymmetric paths (in terms of bandwidth or latency) (such as 3G and WiFi) because of issues caused by disordering. In some of the simulations we also added cross-traffic in the link 0-1. Cross-traffic is

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

added through UDP flows that come in every 15 s, consume half of the bandwidth (5Mbps) for 5 s and then leave.



a) AFCT for E[L] = 30



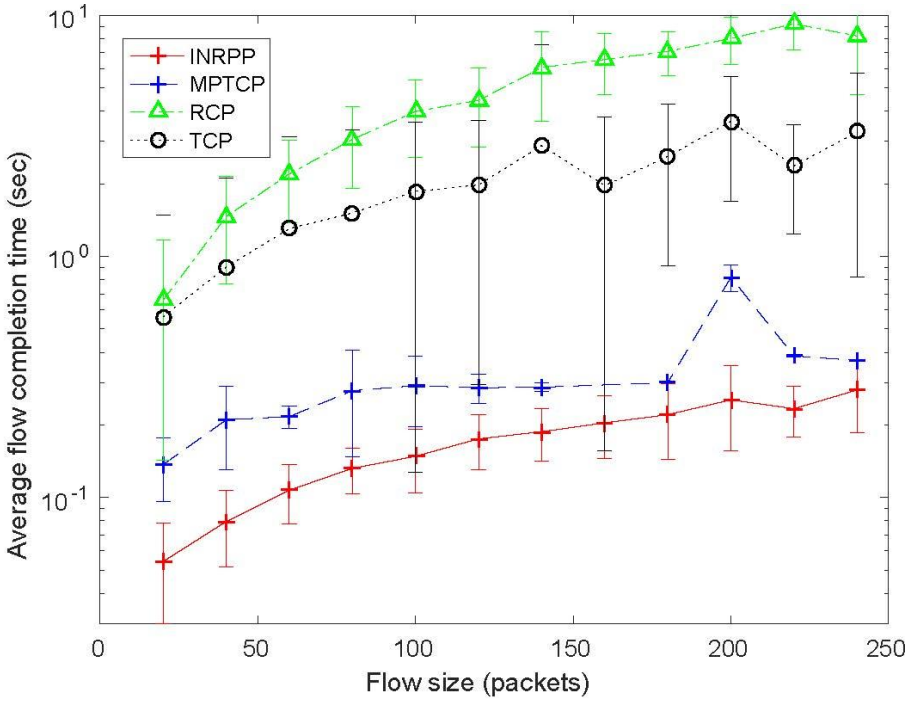
b) AFCT for E[L] = 500

Figure 7: Multi-homed AFCT w/o cross traffic

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

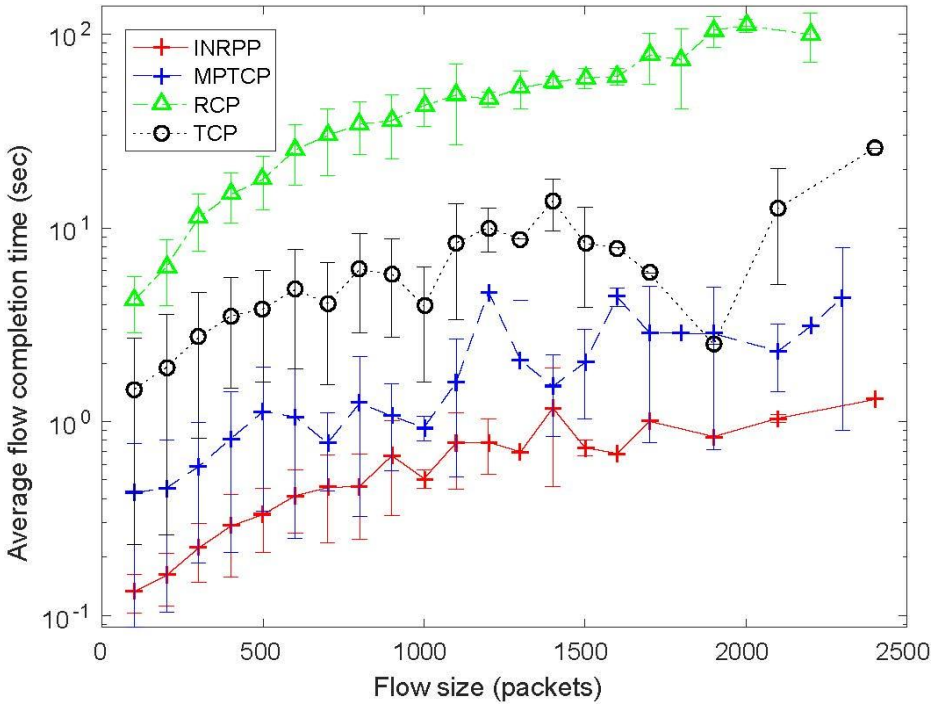
Figure 7 shows the AFCT for different flow sizes when there is no cross-traffic in any of the links. We present the AFCT for all protocols using $E[L] = 30$. In Fig. 14(b), we present results using $E[L] = 500$. In Fig. 14(a), we observe that the AFCT of both MPTCP and INRPP are much lower than TCP or RCP. In this case, TCP does not differ significantly from RCP since we have a larger number of flows in the simulation and RCP's estimation of number of active flows is more accurate. However, RCP does not outperform TCP because the number of active flows is still not large enough for RCP to reduce its error rate in estimating the number of active flows.

The performance of RCP and TCP is substantially inferior compared to MPTCP and INRPP, because neither RCP nor TCP can use more than one path to send data. When comparing only MPTCP and INRPP we can see that INRPP clearly outperforms MPTCP, providing shorter flow completion times (up to around 50% in some cases), and therefore using the network resources more efficiently than MPTCP. First of all, MPTCP is not able to use all the detour paths available. MPTCP can use more resources than TCP, but at the same time it also inherits its limitations: First of all, MPTCP is an end-to-end resource pooling mechanism, and therefore, cannot exploit mid-path resources as INRPP does with the residual bandwidth available in detour paths. Secondly, AIMD-based MPTCP faces drops and timeouts, that most of the time does not imply a significant increase in AFCTs, but mainly causes poorer fairness performance – see Table 2. In fact, the chances of packet drop and timeouts in MPTCP increase linearly with the number of sub-flows. This is shown by the substantially worse fairness performance in case of short flows (see $E[L] = 30$ in Table 2).



a) AFCT for E[L] = 30

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



b) AFCT for E[L] = 500

Figure 8: Multi-homed AFCT with cross-traffic in link 0-1

Figure 8 shows the AFCT, but this time with the cross-traffic pattern previously explained. In Figure 8(a), we present the AFCT for all protocols using $E[L] = 30$ and in Figure 8(b), we provide results using $E[L] = 500$. In this figure, we observe again that INRPP significantly outperforms all other protocols. In this case, the flow completion time of RCP and TCP clearly gets worse due to the cross- traffic. However, the AFCT increment because of the cross-traffic, using MPTCP or INRPP, is almost imperceptible compared with RCP and TCP. When using RCP or TCP, the cross-traffic generates more load than the link can absorb. When using MPTCP or INRPP alternative path(s), which do not use link 0-1, can be used to forward traffic in excess. Therefore, the multipath capability diminishes the impact of cross-traffic on AFCT.

In Table 2, we show the average fairness for the previous simulation. In all cases, INRPP presents the best performance, except for $E[L] = 500$, where RCP provides only marginally better fairness than INRPP.

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

Table 2: Average fairness using random flow sized in the multi-homed topology

Protocol	No cross-traffic		With cross traffic	
	$E\ L\ = 30$	$E\ L\ = 500$	$E\ L\ = 30$	$E\ L\ = 500$
TCP	0.8205	0.8575	0.4321	0.8883
RCP	0.9301	0.9589	0.7569	0.9959
MPTCP	0.6103	0.8947	0.6104	0.8515
INRPP	0.9298	0.9895	0.8225	0.9897

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 topology[1] 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 a $E\ L\ = 125$. All the links have the same capacity of 1 Gbps except the links connecting edge 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.

[1] 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

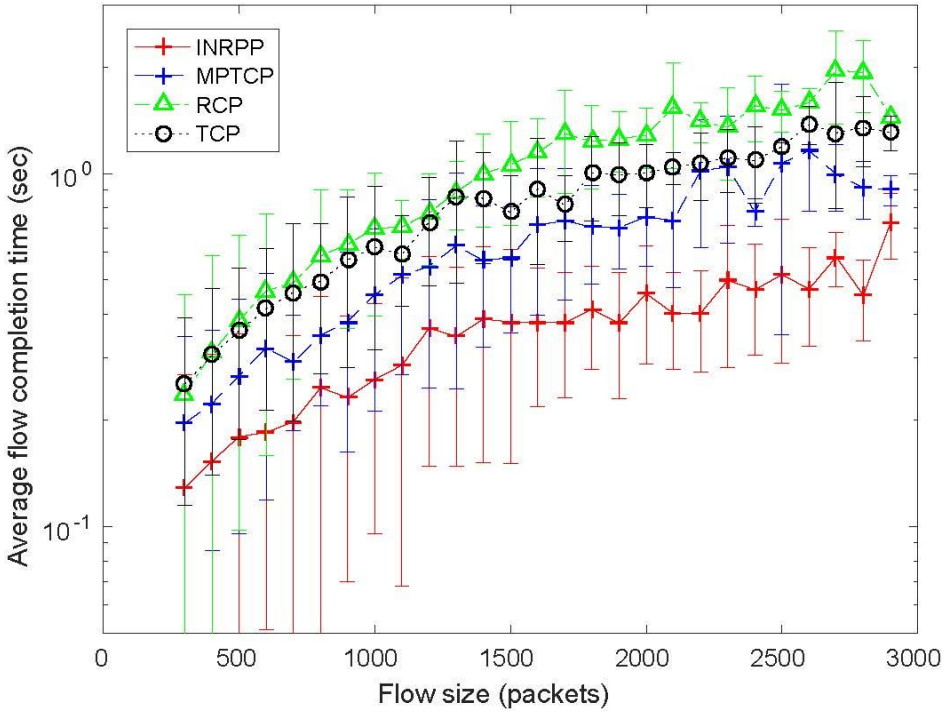


Figure 9: AFCT for the rocketfuel topology

In Figure 9, 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.

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]. Thus, in ICN, content can be transparently delivered from any cache-enabled in-network device (e.g., router), as long as this device holds a valid copy of the requested content and the request traverses it.

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

Apart from the optimization of the caching strategy, another 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. Content discovery, or else, request to cache routing can follow one of two approaches, either opportunistic on-path routing, where content is searched on-path as the request is travelling towards the content source, or coordinated off-path resolution-based, where requests are forwarded off the shortest path to some designated cache that is likely to 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].

Though off-path resolution-based content discovery mechanisms produce relatively small extra traffic in the network, they introduce considerable amounts of coordination/communication overhead and delay and as such, result in inherently less scalable solutions. On the other hand, on-path content discovery mechanisms provide a probabilistic solution, but with inherently limited gain that can be increased by decreasing the correlation of the cached items in the routers of the network. 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. The newly introduced D-FIB is combined with scoped interest forwarding techniques in an attempt to decrease the delivery latency of requested items and increase the cache hit rate of the system, by exploring the caching capabilities of nearby routers. In order to support the proposed operation, we also introduce an Off-path Forwarding Flag (OFF) bit to the Interest packet to distinguish whether it is following the FIB entries towards the content origin (OFF is set to zero by the corresponding forwarding router), or is heading towards the direction of users with similar satisfied interests. For the second case, when OFF is set to one by the forwarding router, the Interest packet follows matching entries in the D-FIB of each passing-by router. We also introduce a Total Forwarding Counter (TFC) to control the total number of Interests that will be generated from a single data request as explained below. More details of the OOCD proposal will be provided in the final version of the architecture design in D3.2 and D3.4, to be submitted at the end of the project.

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

4.2.2. Demo Key Features

This service is not planned to be implemented in real devices and deployed in the final demo. In order to evaluate and demonstrate the good performance of OOCD we require a large scale network, therefore we will evaluate and validate OOCD only by simulation

4.2.3. Hardware and Software Requirements & Experimental Settings

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

4.2.4. Validation & KPIs

For the evaluation of the proposed discovery scheme we used, after adequately extending, ndnSIM [11] (version 2.1) —an ns3 based, packet-level simulator. The extensions to the ndnSIM simulator required to support the proposed content discovery mechanism involved enhancements to the ndn-cxx library and the NDN Forwarding Daemon (NFD) components. Enhancements to the ndn-cxx library allow the Interest packets to carry an OFF bit, which is set when an interest is forwarded downstream and a Total Forwarding Counter (TFC) field. The NFD library was upgraded to include the D-FIB interest forwarding mechanisms. Each of the new Interest forwarding mechanisms is implemented as a new strategy in the NDN forwarding strategy layer. Finally, the data processing portion of the NFD was slightly modified to add new entries to the D-FIB table as data packets are processed.

We use the 4755.pop.cch network topology as provided through the Rocketfuel dataset [12]. Spanning over the Asian continent, the 4755.pop.ch topology has V = 191 routers and 242 bidirectional links. Out of the 191 routers, 148 are edge routers (with degree less than 3), 39 are gateway routers (i.e., connected to an edge router and has degree larger than 2), and 4 backbone routers. We consider a scenario where the content item population is $M = |M| = 105$ and requests are generated to the network with a rate equal to $\xi = 100 reqs/sec$.

Requests are generated in the network with rate $r = \{r^1, \ldots, r^M\}$, where r^M denotes the aggregated incoming request rate (in requests per second) for content item $m \in M$. The request rate for each item is determined by its popularity. Here we approximate the popularity of the items by a Zipf law of exponents z [13]. Our evaluation is based on the following metrics:

- **Cache Hit Ratio** (in % of issued interests): It is the percentage of the interests that have been satisfied (found the requested item) from a cache/router of the network (off-path and on-path).

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

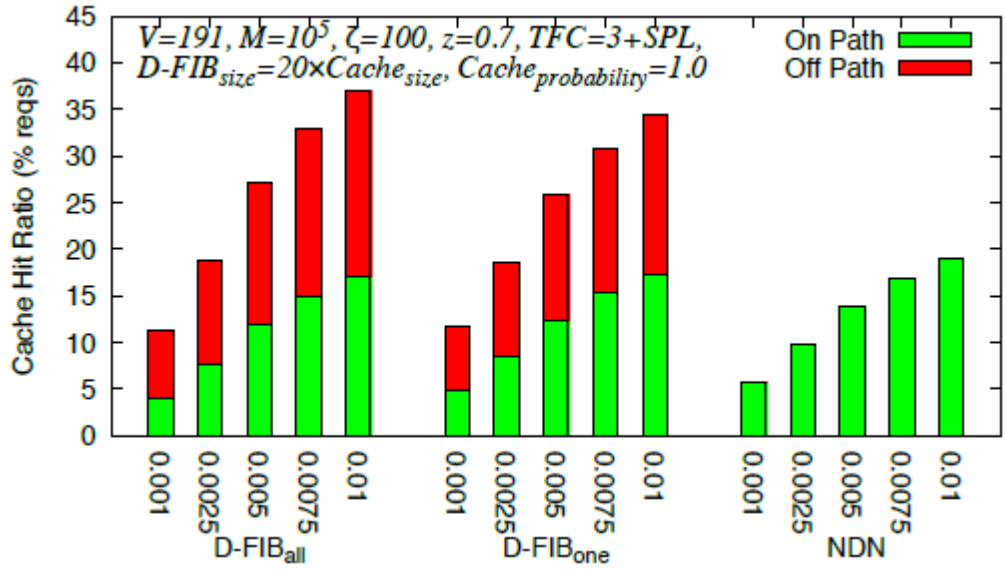
- The **Minimum Hop Distance** (in hops): It is the number of hops travelled by the (first) data arriving at the user from a responding router or the content origin for each successful request. This metric is indicative of the transfer delay as a function of hops in the network.
- The **Mean Traffic Overhead** (in hops): It is the mean number of hops that the initiated Data packets travel in the network, until they are discarded or reach the consumer, for each satisfied interest.

Our simulation setup consists of a set of consumers (i.e., hosts) each connected to exactly one edge router, and for each request, one of the consumers is selected uniformly random as the originator of the request. We use a simplified scenario with a single producer (e.g., large content cache) acting as the content origin for all the requests, and it is conveniently located near the core of the network1 with direct links to two randomly chosen backbone routers. In a real deployment scenario, the distance between the content origins (e.g., CDNs, content providers) and consumers can be more than just a few (router-level) hops. In fact, the content origins can be one or more domain-level hops away from the consumers, which could translate to significant latency in the end-to end communication. Therefore, our results comparing the hop distances when fetching content from the origin with the hop distances when fetching the content from the routers' caches can be thought of as the best-case scenario; in reality the additional distance that needs to be covered to reach a content origin is expected to be more than just a few router-level hops.

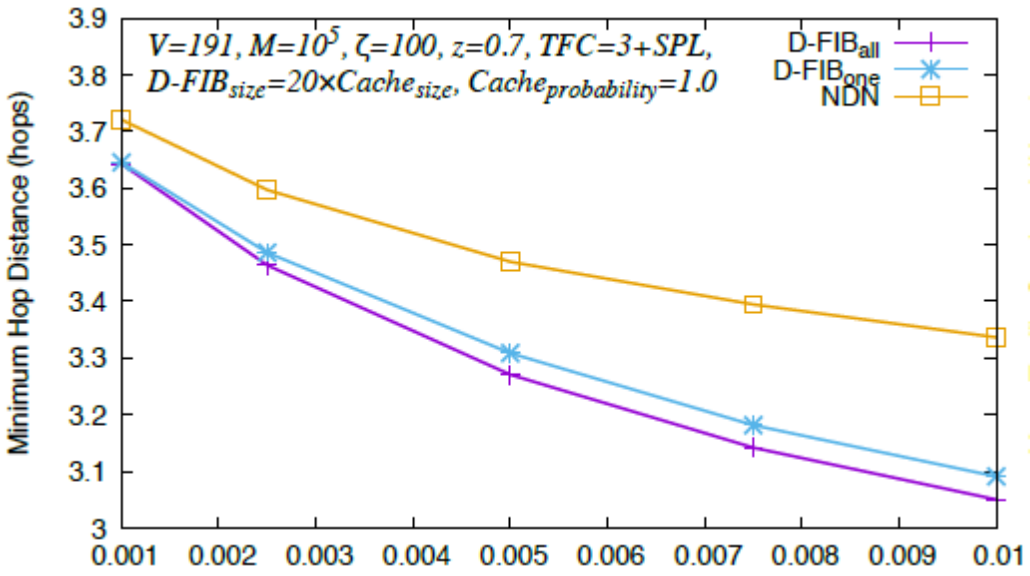
In all the experiments presented below, we observe the network over a duration of one hour, following an hour of "warm-up" phase during which the requests are forwarded using NDN's default strategy and the D-FIB table and content caches are populated during data processing. Finally, we assume without loss of generality that the routers have the same storage capacity and that each responded packet is cached in all routers along the return path. This caching policy is also known as Leave Copy Everywhere (LCE) and is the default policy in NDN [14] ($\mathrm{Cache}_{\mathrm{probability}} = 1$).

In Figure 10 we depict the impact of the cache capacity, expressed as the fraction of the items population that can be stored in the cache (i.e., CS component) of a router. A first and straightforward observation is that the usage of the proposed forwarding strategies leads to ↑100% increase in the cache hit ratio compared to the traditional NDN strategy. Particularly, we observe that when the cache capacity of each router is only 0.5% of the total item population, almost 30% of the total issued requests can be satisfied from the caches within the network, whereas the NDN's default strategy only manages to satisfy less than 14% of the requests from the on-path caches.

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

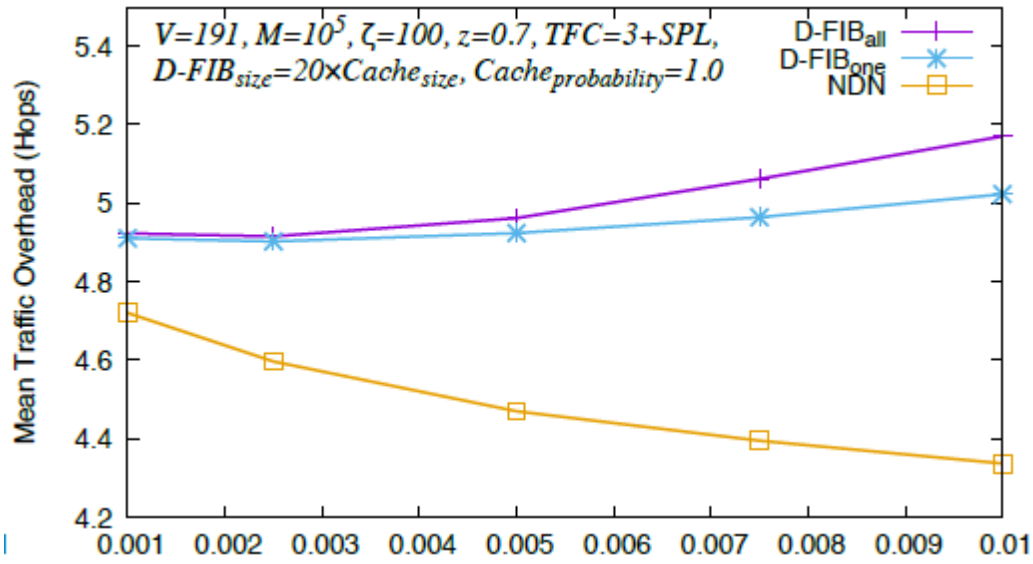


a) Cache size / Content population



b) Cache size

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

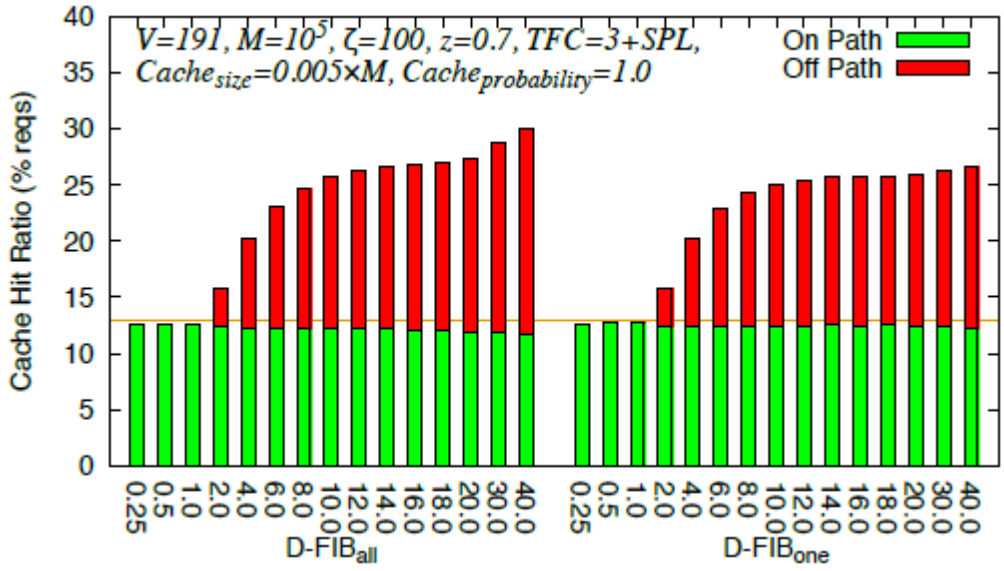


c) Cache size / Content population

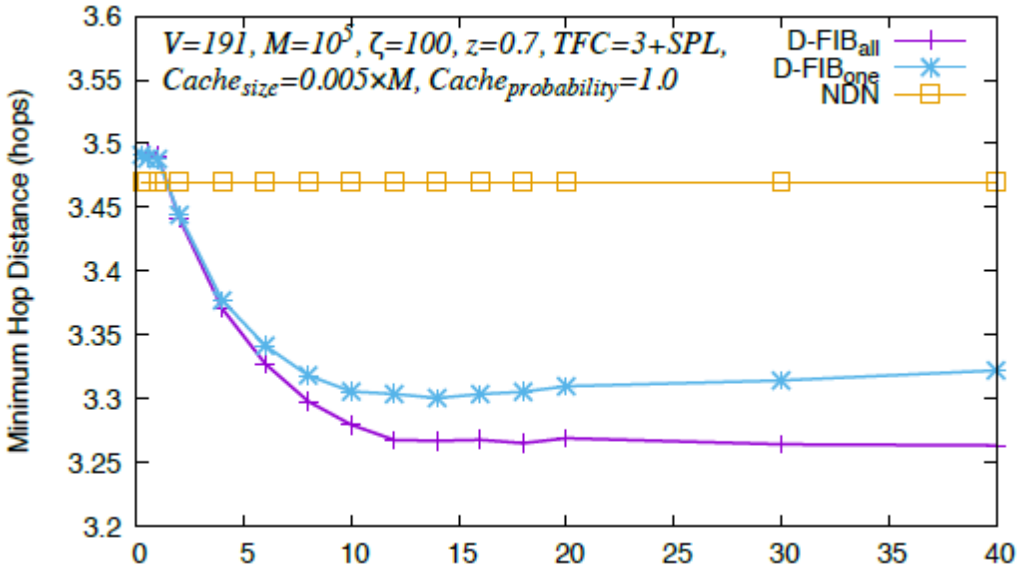
Figure 10: The impact of the cache capacity of each router in the performance of the examined forwarding strategies

Additionally, the newly proposed strategies also manage to decrease the hop count (i.e., latency) for the retrieval of content items. For both of the proposed strategies, the minimum hop distance is reduced by approximately 10% when the cache size of each router is only 1% of the content population. In a real deployment scenario, we expect the difference to be larger as the requests will travel outside the network to fetch content. Therefore, the consumers would potentially accrue more benefit from the increase in cache hit rates as this translates to better QoS for users, namely the latency in obtaining data. From the comparison of the two proposed downstream interest forwarding strategies we observe that the all strategy performs slightly better regarding hit ratio compared to the one strategy, but at the cost of slightly increased network overhead. Note here that we assumed only three extra quotas (i.e., TFC is equal to three plus the shortest path length (SPL) in hops to the origin). This means that both strategies manage to double the cache hit ratio of the system initializing only a very small number of downstream interests. In the rest of the experiments presented below, we set the capacity of the routers to 0.5% of the total item population.

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

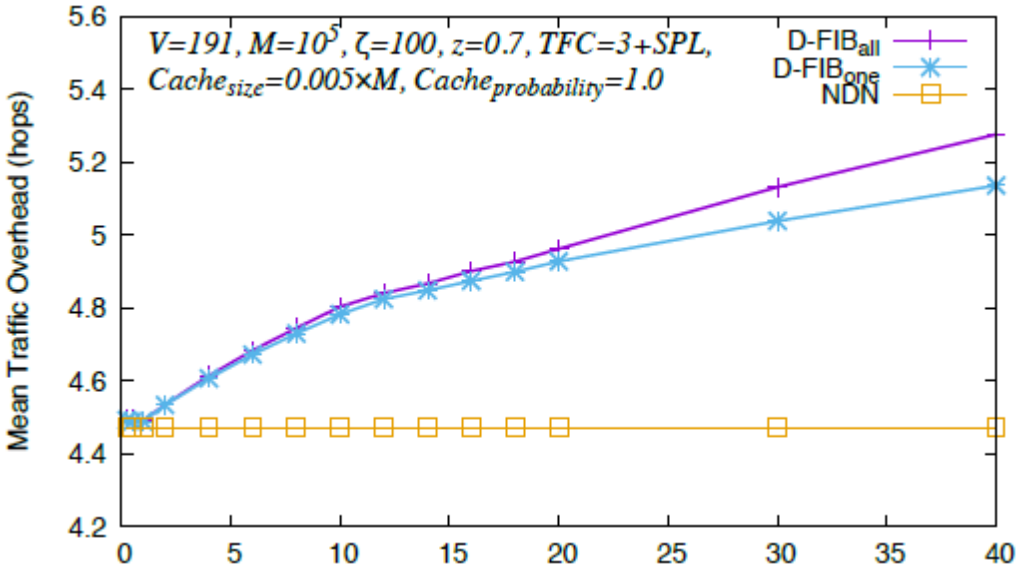


a) D-FIB/Cache size



b) D-FIB/Cache size

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

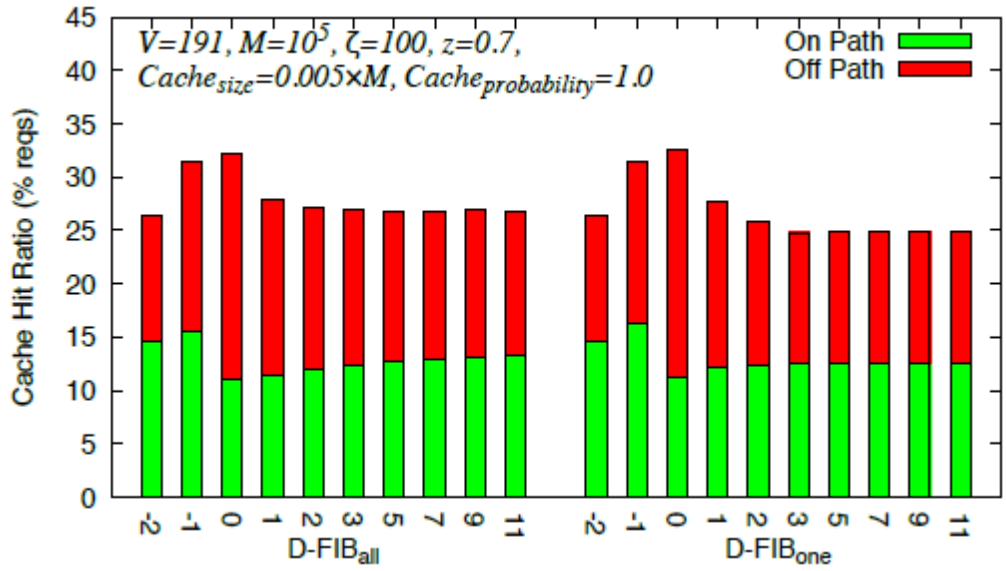


c) D-FIB/Cache size

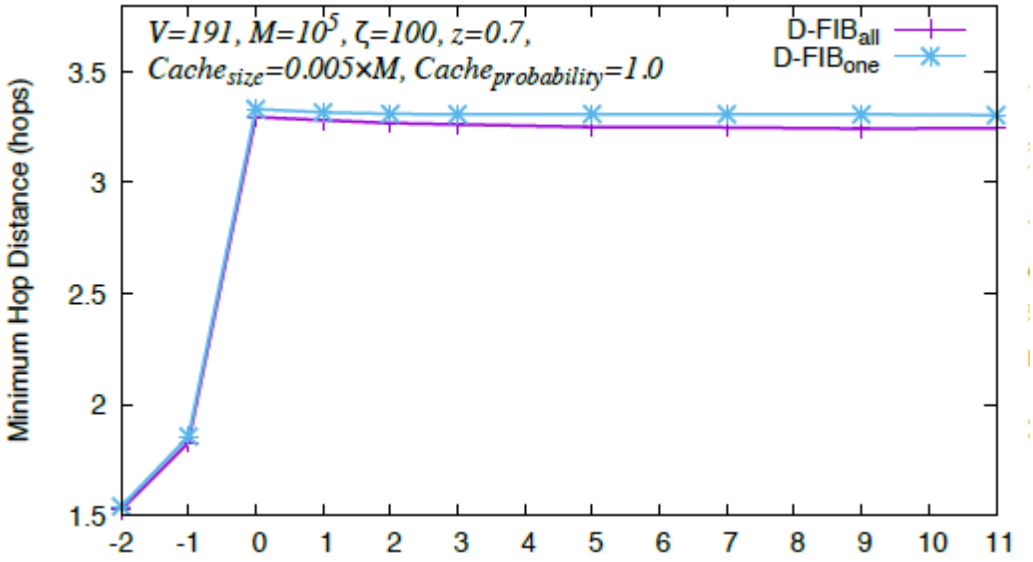
Figure 11: The impact of the D-FIB size in the performance of the examined forwarding strategies

The D-FIB table on each router has a limited size and entries follow an LRU replacement policy. As such, the size of D-FIB also affects the performance of the proposed forwarding strategies, since small D-FIB sizes means that even if there are cached copies at nearby routers, the lack of entries in the D-FIB of routers cannot exploit them. In Figure 11, we depict the impact of the DFIB size, expressed as the ratio of the router's cache size. From the cache hit rate plot we observe that when the D-FIB size is less than the cache size of a router, the proposed strategies cannot retrieve any content exploiting off-path caches. This is due to the rate at which D-FIB entries are updated, which does not allow an interest for a piece of content to establish a trail for a similar future interest. Both strategies start to perform significantly well in terms of cache hit rates for sizes of DFIB that are at least 10 to 16 times larger than the cache size (i.e., in number of entries and not actual disk capacity).

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

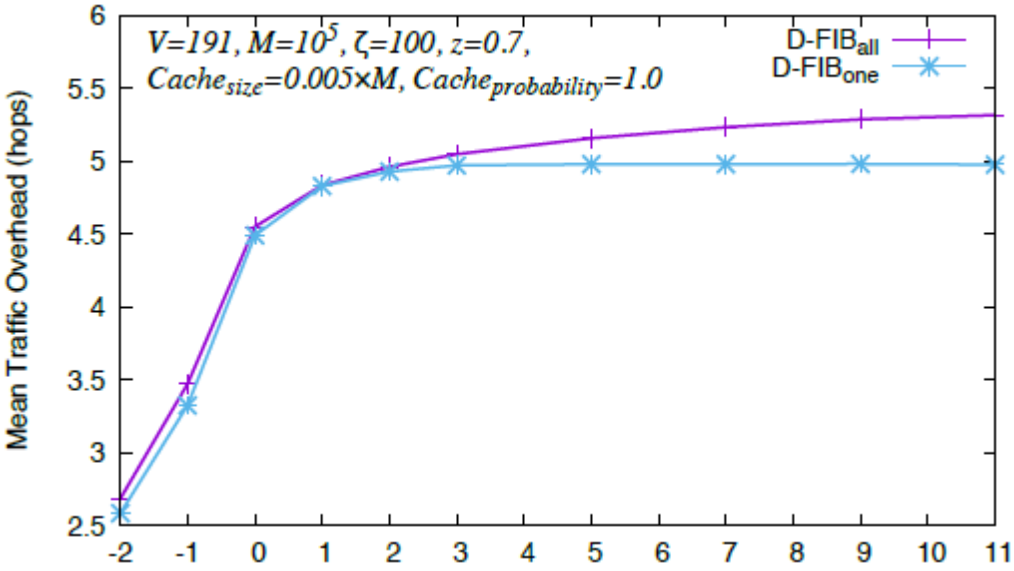


a) Extra Quota



b) Extra Quota

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



c) Extra Quota

Figure 12: The impact of the TFC value in the performance of the examined forwarding strategies

The TFC of each Interest packet is set to a predetermined value, by the first-hop router where the user who issued the request is attached to. In the above experiments this value is set to TFC = 3+SPL, where SPL is the shortest path length in hops from each router to the origin server and extra quota is set to 3. In this section we examine the performance of the proposed forwarding scheme for different values of extra quota (note that the values in the x-axis of Figure 12 are additional to the SPL, which is obtained by the first-hop router from its FIB table). From Figure 12 we observe that increasing the TFC value does not increase the hit ratio. On the contrary, a large TFC value slightly decreases the hit ratio of the system. This is mainly due to the LCE policy, since a large TFC value means a larger number of downstream interests. This, in turn, results in more duplicate responses and merely in a higher rate of cache replacements. In other words, a single interest that is multicast towards many different directions results in duplicate responses of the same data packets that will be cached along the paths towards the user and replace already cached content. This increases the correlation and redundancy of cached data in the network and minimizes the probability of future interests for different content to find matching cached content. For cache probabilities set to around 50%, the hit ratios slightly increase with increasing TFC value, but we do not present these results due to space limits.

Table 3: Request satisfaction rate of the first requests for different extra quota values

Extra Quota	Sat. $Rate_{all}$	Sat. $Rate_{one}$
-2	26.3%	26.3%
-1	31.3%	31.3%

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

0	91.3%	91.3%
1	98.5%	98.5%
2	99.2%	99.7%
3	99.7%	99.9%
...	...	...
11	100%	100%

In Figure 12, we also consider negative and zero extra quota values where TFC in each packet is set to extra quota+SPL. A request packet carrying a negative quota can only retrieve the content from a router's cache since the TFC allowance will be spent before reaching the content origin even when the request follows the shortest path. As mentioned before, the first hop router is responsible for setting the TFC value in the packets and also for retransmitting interests when the previous attempt is not successful. The TFC value can be set according to different goals. If the primary goal is to reduce the percentage of requests that reach the content origin and minimize the overhead of duplicate data packets, then the first hop router can initially try a negative quota value (e.g., -1). When the request is not successful in returning data, larger quota values can be used in the retransmitted requests. On the other hand, if the primary goal is to obtain the data at first attempt and thus increase QoS, the TFC value in the request can be set to a positive value to guarantee retrieval of data. Table 3 shows the satisfaction rate of first requests (i.e., excluding retransmissions) for various extra quota values for the "one" and "all" forwarding strategy. For negative extra quota values, the satisfaction rate of the first requests is below 32%. However, for zero extra quota value, the satisfaction rate reaches 91% and only 59% of the queries reach the content origin, whereas 32% of the queries are satisfied from the caches. The overall performance and overhead of the proposed forwarding mechanisms do not differ significantly, due to the sparseness of the ISP topology that we used in our experiments. However, their performance differs more significantly for denser topologies that we have also examined, but not depicted due to space limitations.

4.2.5. Use case mapping

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

4.3. Named-based replication

4.3.1. Description

UMOBILE will develop 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

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

and with specific life expectancy, optimised by prioritisation rules considering emergency messages more important than other messages, e.g., messages between friends.

Based on the algorithms designed in the task, we plan to prepare a demo to show the key features of the proposed system. The demo would consist in some mobile users in a certain area and some access point (AP) or fixed devices. These mobile devices would have restricted capacity (memory and/or battery). During the demo an emergency service will coexist with other services such as instant messaging or a video delivery service. We plan to use WiFi Direct as a main communication technology, but we don't discard yet the use of Bluetooth or WiFi Aware capabilities for this task.

4.3.2. Demo Key Features

This service will demonstrate in the proof-of-concept:

- Successful emergency message delivery rate
- Message delivery restriction in a certain area and time space

4.3.3. Hardware and Software Requirements & Experimental Settings

First, we will provide a validation of this service by means of simulation. At the end of this project, this service will be integrated and showcased in the final proof-of-concept. To that aim, the following hardware and software will be required:

- Smartphone devices with Android OS: Samsung Galaxy Tab A and Android 6.0.1
- Mikrotik routers with OpenWrt

4.3.4. Validation & KPIs

We will include an evaluation performed with ONE simulator, once ready for D4.3.

4.3.5. Use case mapping

This service will be showcased in the emergency scenario use-case proof-of-concept presented in deliverable 5.3 [1].

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.

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

4.4.2. Demo Key Features

KEBAPP will provide access to processed information to mobile users in connectivity restricted scenarios enabling mobile application sharing between users and/or users and AP, when users have no Internet connectivity but other users or UMOBILE devices can provide this information locally. This way, UMOBILE users will be able, for instance, to recompute paths (using a developed UMOBILE service) under emergency situations even without Internet connectivity or with connectivity restrictions, when the user discovers an AP or UAV offering the service. KEBAPP we will use WiFi and WiFi Direct communications to enable application sharing between users, and we will explore the possibility of using WiFi Aware capabilities to improve the discovery process.

4.4.3. Hardware and Software Requirements & Experimental Settings

KEBAPP is thoroughly validated in terms of simulations. Furthermore, we also experimentally validate it using the following requirements:

- Smartphone devices with Android OS: Samsung Galaxy Tab A and Android 6.0.1
- Mikrotik routers with OpenWrt

4.4.4. Validation & KPIs

KEBAPP is implemented over Android devices. Currently, we implemented KEBAPP in Android 6.0.1 and developed a sample application using Samsung Galaxy Tab A Tablets. We also modelled the application sharing framework approach in the ONE simulator [15], in order to evaluate the feasibility of the KEBAPP application sharing framework.

For the evaluation of the proposed framework, we consider a KEBAPP-enabled tube map application that provides information on train lines and their respective schedule, as well as real-time information regarding delays, closed stations etc. We assume that a significant percentage of users commuting by tube have installed at least KEBAPP on their smartphones. In particular, we consider a common scenario where a commuter wants to know if there are any delays on the lines that he/she needs to take in order to reach his/her destination, as well as the estimated time of arrival to the destination. By exploiting the proposed shared application framework, a KEBAPP user can take advantage of different applications (within the same context) shared by other users that have recently entered the tube station. In particular, by using as input simple information such as current location and destination of a user, the devices of nearby users compute the estimated arrival time, along with the route that the user needs to follow, and send a reply back to the user. To evaluate the proposed application, we perform simulations based on mobility traces, from a subway station in downtown Stockholm2.

[2] Trace was obtained from http://crawdad.org/kth/walkers/20140505/

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

For different numbers of KEBAPP users (which reflect the KEBAPP-enabled application penetration rate), we examine to what extent route calculation requests are successfully responded by other user applications. In particular, we randomly select a subset of the trace nodes, as the KEBAPP users, and randomly generate route calculation requests during their short stay in the station. We assume that a request can be successfully responded by only a small percentage (1, 5, or 10\%) of the other KEBAPP users, which are also selected in random; this percentage corresponds to the users that have enabled the sharing capabilities of their KEBAPP application, and have access to the requested data, as well as sufficient battery level. We should also note that each commuter visits the tube station for a short period of time (i.e., with an average stay of around three minutes), so the probability that two users that run a KEBAPP application do not exist in the subway station simultaneously is high.

All sets of simulations are repeated 1000 times. We measure the successful responses of route calculation requests, in terms of probability and response time. In particular, we measure: i) the average Response Ratio (RR), i.e., the fraction of the total generated requests that receive a successful reply, and ii) the average First Response Time (FRT), which is the average time between each request issue and the first reply, for those that are successfully responded. Note that in this preliminary experiment, we do not consider computation processing time and protocol-specific delays, as we do not focus on protocol implementation-details.

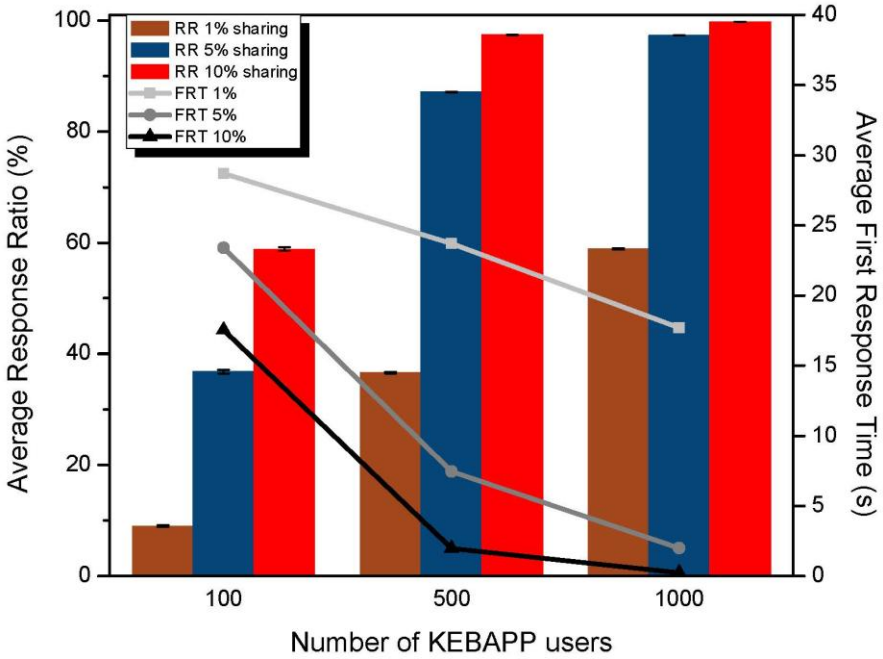


Figure 13: Evaluation results

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

Our evaluation results, presented in Figure 13, show that, as expected, the response ratio increases with the KEBAPP-enabled application penetration rate, as well as the percentage of the sharing users. In particular, even with a small penetration rate, (when only 100 out of the 3300 commuters are KEBAPP users), the KEBAPP users have a chance to get a successful response. As shown in Figure 13, the lowest response ratio of 9% is achieved when only one other passenger (during the simulation period of one hour) can successfully respond to the user request. The probability of a successful response gets significantly higher when more users are willing to share their resources; this probability is increased by a factor of ~6.5 (from 9% with only one user sharing resources to 58.8% with ten users sharing resources out of 100 KEBAPP users). As the number of users increases, the response ratio approaches 100%.

As far as average response time is concerned, we notice that response time decreases as the number of users that are sharing resources increases, since the user that issues a request is more likely to encounter such a user. In particular, the maximum average delay is observed at the lowest sharing ratio of 1%, and it spans from 17-29 seconds. For increased information availability, a user request can almost immediately find a match, within a few seconds.

In that case, we have used a variety of movement models/patterns (included in the ONE simulator) in order to assess the efficiency of mobile content dissemination using KEBAPP. In that case, there are a set of users considered source nodes that are providing content to a set of users called destination nodes. We have used the Helsinki city population and city centre as the default urban environment (area of size equal to 8.3km x 7.3km). By default, we assume that the population of the destination nodes is equal to 1000, whereas the default population of the source nodes (nodes providing information to other nodes) is equal to 50. The distribution of source nodes is as follows: out of the 50 nodes, we assume that 18 are buses that follow predefined routes, whereas the rest nodes (i.e., 32 in the default scenario) are assumed to be users that follow the working day movement model [16]. Destination nodes (nodes receiving content from source nodes) are configured as follows: out of the 1000 nodes 20% are assumed to be tourists. Tourists choose random destinations (either total random points in the map or one of the seven "points of interest" (i.e., tourist attractions) in the city centre) to which they travel following the shortest path and wait randomly 2 - 15 minutes. The majority of the destination nodes, i.e., the remaining 80%, are assigned the working day movement model, which allows them to travel to designated office spaces on the map and travel for other evening activities later in the day. All nodes start at their base/home and travel to their office, either directly by car (50% of nodes) or by bus (remaining 50%). Once they reach the office, they spend 7 hours there and at the end of the office day there is a 50% chance the node will go for an "evening activity" and 50% chance it travels home for the rest of the day.

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

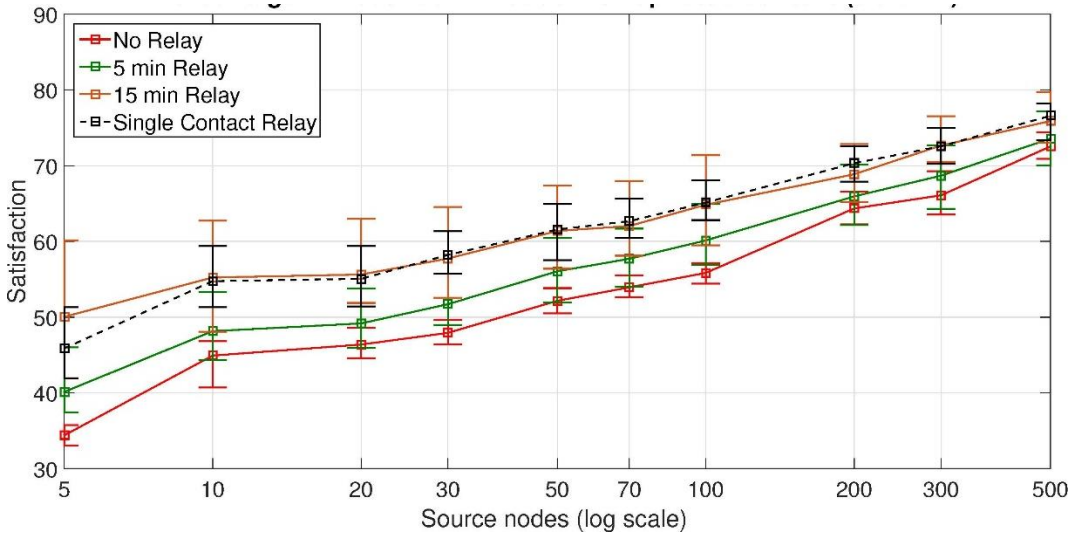


Figure 14: The impact of the number of source nodes relay time in the satisfaction rate

In Figure 14 we depict the impact that the number of source nodes have in the satisfaction rate (the percentage of the destination nodes with updated content over all destination nodes at the end of each application update period), that means the percentage of users that receive updates from a certain content directly using KEBAPP from other users without accessing the Internet. Note that the number of destination nodes is always equal to 1000 nodes, while the source nodes are always buses when source nodes are greater than 18, whereas when source nodes are less than 18 the remaining nodes are chosen to be nodes that follow the working day movement model. This option is more rational since tourists are usually visiting a place for a limited amount of time, whereas "workers" tend to stay in one place for longer periods. Destination nodes also relay messages with different configurations: No relay, Relay for 5 min only, relay for 15 only or relay the message only once. We observe exponential increase in the Satisfaction Ratio as the number of sources nodes increases. This increase is more pronounced for small relay times.

Last, but certainly not least, we look into the energy consumption of user-operated CDNs. Energy is the price paid by the system in order to disseminate content in a D2D manner and therefore, cannot afford to be overlooked in our feasibility study.

Table 4: Source nodes energy consumption results (1-hour summary, 41-62 messages)

Message size:	5MB	50MB	100MB
mWh	106.6 - 161.2	179.99 - 272.18	212.38 - 321.16
% Battery phone	0.86 - 1.3	1.45 - 2.19	1.71 - 2.59
% Battery tablet	0.41 - 0.62	0.69 - 1.05	0.82 - 1.24

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

Table 5: Relay nodes energy consumption results (1-hour summary, 3 messages)

Message size:	5MB	50MB	100MB
mWh	7.8	13.17	15.54
% Battery phone	0.06	0.1	0.12
% Battery tablet	0.03	0.05	0.06

In Table 4 and Table 5 we show preliminary results using real devices (Galaxy Tab A tablet and Samsung Note 3 smartphone) and transmitting different file sizes using WiFi Direct. These results are obtained with two static devices separated by around 10 meters and with a RSSI value close to -50dbm. We carried out a set of experiments using 5 MB, 50 MB and 100 MB of exchanged files and we extrapolated these results to the number of messages sent by source and relay during an update interval of 1 hour. We also extrapolated from this energy consumption the percentage of battery consumed using a tablet (i.e., 7000 mAh/25.9Wh capacity) or using a smartphone (i.e., 3200 mAh/12.4Wh capacity). The energy consumed by the WiFi Direct application is measured using the Trepn Profiler tool [17]. From Table 4 and Table 5, we can observe that the energy consumption for the ubiCDN source nodes goes from 106.6 mWh in the best case (41 messages sent) when sending 5 MB messages, to 321.16 mWh worst case (62 messages) when sending 100 MB. This means the percentage of the battery consumed is between 0.86% and 2.59% for a smartphone, and between 0.41% and 1.24% for a tablet. In case of relay nodes, the energy consumption goes from 7.8 to 15.54 mWh, meaning from 0.06% to 0.12% of the battery for a smartphone and 0.03% to 0.06% for a tablet, respectively. From this analysis we can consider that energy consumption is not a big issue.

The results obtained in this energy consumption analysis, despite being preliminary with simple tests, are in line with the results presented in [18]. In [18] authors report that an average smartphone can transmit up to 44GB of data before depleting the battery, with an average consumption of a 1 J/MB (i.e., 1.38 mWh for a 5MB file) in a walking speed mobility scenario.

4.4.5. 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

DTN is a technology that facilitates interoperable communications between different, intermittently connected networks. Like IP, DTN operates on top of existing network architectures, creating a DTN overlay. In particular, DTN extends internetworking in the

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

time domain: rather than assuming a continuous end-to-end (E2E) path as IP networks do, DTN operates in a store-and-forward fashion: intermediate nodes assume temporary responsibility for messages and keep them until the next opportunity arises to forward them to the next hop. While stored, messages may even be physically carried within a node as the node is transported: the model is also termed store-carry-forward. This inherently deals with temporary disconnections or disruptions and allows the connection of nodes that would otherwise be disconnected at any point in time. The core of the DTN architecture lies in the Bundle Protocol.

IBR – DTN is one of the most popular, well-documented and maintained implementations of the Bundle Protocol designed for embedded systems. IBR – DTN can be used as framework for DTN applications; its module-based architecture with miscellaneous interfaces makes it possible to easily change functionalities like routing or bundle storage. A new face connecting the UMOBILE platform with the IBR – DTN daemon has been developed in the framework of the project. This allows the natural integration of NDN and DTN networking platforms and enables a whole set of new communication capabilities.

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

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 UAV
- 1 hotspot

4.5.4 Validation & KPIs

We have performed a successful initial assessment of the IBR-DTN integration into the NDN forwarding daemon (NFD), 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:

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

- 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 in Figure 15:

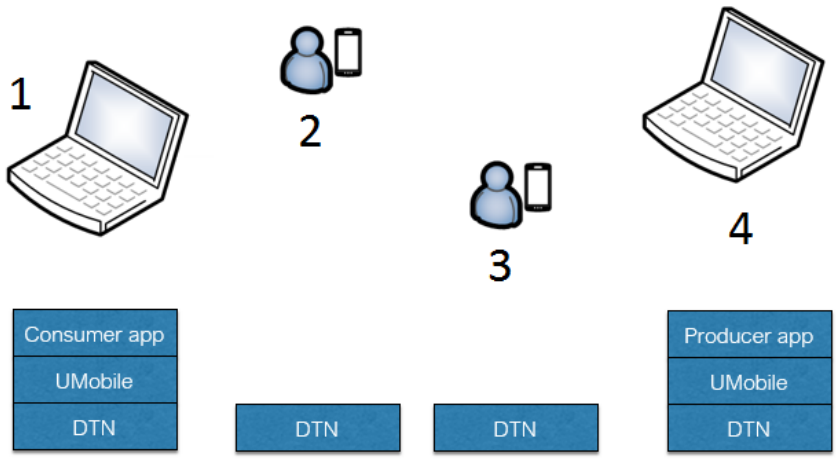


Figure 15: Evaluation setup for IBR-DTN

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 in Figure 16:

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

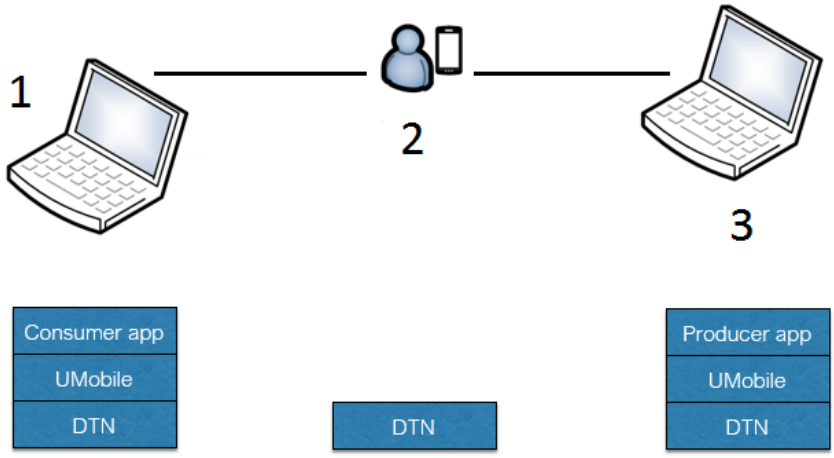


Figure 16: Scenario 1 for IBR-DTN

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.

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 in Figure 17:

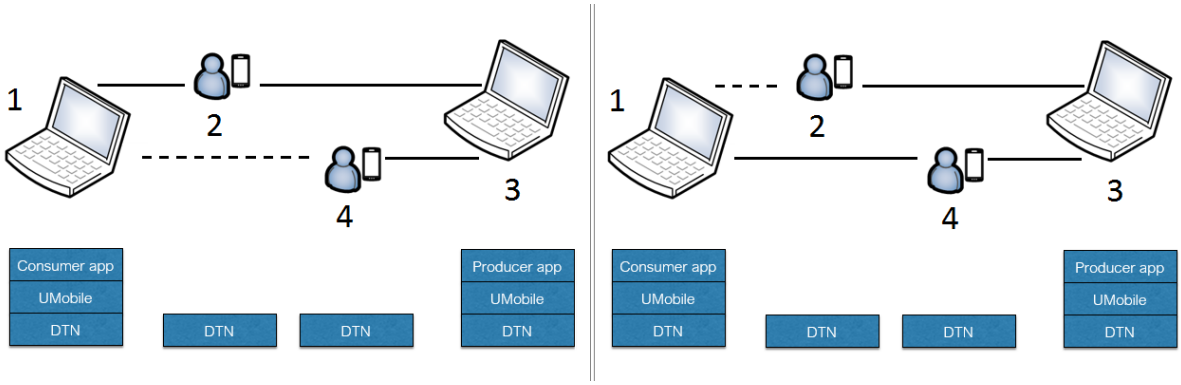


Figure 17: Scenario 2 for IBR-DTN

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

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

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 in Table 6:

Table 6: 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 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).

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

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.

4.5.5 Use case mapping

This demo maps to the emergency fire case in the first demonstrator.

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.

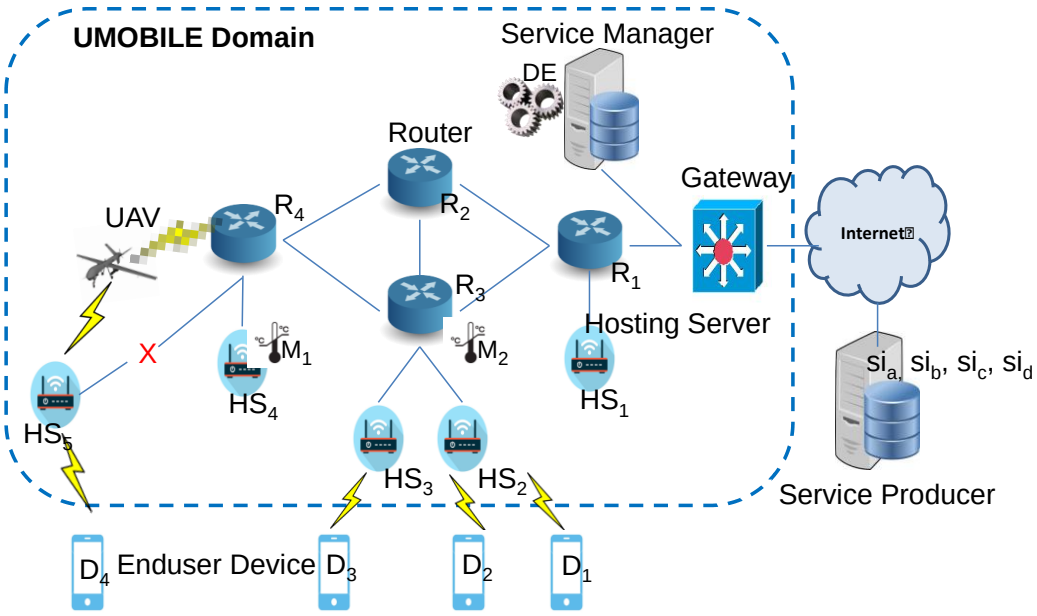


Figure 18: View of UMOBILE deployment with focus on service migration

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

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 18 (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 (sia, sib, sic and sid) and that he has delegated to the network operator the responsibility of delivering them to the end-user upon requests. For example, sa is the service that can be enacted from its compressed image sia. Let us assume that sa is a service needed by a rescue team involved in the emergency scenario. The compressed image sia 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).

4.6.2 Demo Key Features

This demo has two objectives:

- Objective 1 (O1): To measure and quantify the capability of service migration platform as well as identify the critical factors that impact the QoS as perceived by the end-users.
- Objective 2 (O2): To validate the correctness of the functionality of the service migration platform as a whole and of their main components such as its decision engine and monitoring system.

The methodology used is presented next:

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

- In pursuit of O1 we will conduct experiments to identify the performance metrics which impact the latency as perceived by the end-users.
- In pursuit of O2 we will conduct experiments that produce outputs from the main components of the service migration platform. In this manner, we validate that the underlying mechanisms (for example, Push-based and Pull-based communication models) function correctly.

4.6.3 Hardware and software requirements and experimental settings

In this preliminary demonstration of our ideas, we will use a laboratory implementation of the UMOBILE platform discussed in the Introduction of this deliverable (see Figure 1). The rationality behind our choice is that our laboratory setting is a simplified version of the architecture of Figure 1, yet it includes and highlights all the components needed in our experiments and can be mapped onto the architecture of Figure 1. The functionality and realization of the components used in our laboratory experiments is as follows (see Figure 18):

- **UMOBILE Domain:** It is a set of nodes deployed with UMOBILE software so that they are able to take advantage of the NDN facilities. It includes network routers (for example, R_3) and applications hosts (for example, HS_2). All the components shown in the UMOBILE domain require the UMOBILE platform software to support applications such as these demonstration scenarios. However, the deployment of UMOBILE platform software in end-user's devices is optional since these devices can use conventional IP interfaces to connect to the UMOBILE domain. The UMOBILE Domain is linked to the conventional Internet by a UMOBILE Gateway.
- **Gateway**: It is responsible for connecting the UMOBILE Domain to the Global Internet. Its functionality is to convert NDN Interest Requests to HTTP requests and HTTP responses to NDN Response. In Figure 18 it is deployed in one of the NDN routers (R_0).
- **UMOBILE Routers (R_1, R_2, R_3, R_4):** It is a standard NDN router enhanced with features developed by the UMOBILE project. We assume that UMOBILE Routers are in possession of storage that they use for in-network caching and for storing application level information such as dockerized service images. In the laboratory setting, we use Raspberry Pi computers to implement the routers shown in Figure 18.
- **Hotspots (HS_1, HS_2, HS_3, HS_4):** It is a conventional computer with wireless communication facilities that can offer connectivity to End-user Devices (D_1,...,D_4). It has disk storage facilities and virtualization software. In our experiments we use Docker virtualization technology. Likewise, to implement the hotspots we use Raspberry Pi computers that are capable of executing Linux containers.

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

- **End-users devices (D_1, D_2, D_3, D_4):** It is a mobile device with wireless facilities and interested in accessing services provided by the ISP provider. We assume that mobile devices communicate with the UMOBILE Hotspots over conventional HTTP.
- **Service Provider:** We assume the emerging business model where network providers are responsible for providing both network connectivity and access to services to end-users. In Figure 18, we assume that the service provider is the owner and in full control of the resources included in the UMOBILE Domain. Consequently, the Service Producer has delegated to the Service Provider the responsibility of deploying their services.
- **Service Producer:** It is an entity in possession of some arbitrary services of interest to the end-users. He stores them as compressed dockerized images si_a, si_b, si_c and si_d which are at the disposition of the Service Provider.
- **Services images (si_a, si_b, si_c, si_d):** It is a compressed dockerized images stored within the Service Producer.
- **Services (s_a, s_b, s_c, s_d):** It is an application of interest to the end-user that can be instantiated from a corresponding compressed dockerized image. We assume that the services demand different levels of QoS, for instance different latencies. For example, service s_a is latency-sensitive whereas s_d is latency-tolerant.
- **Service Manager:** It is a piece of software that implements all the functionality that the Service Provider needs to deploy his services, including disk space to store both services and compressed images. In Figure 18, the Service Manager is strategically deployed on a computer directly connected to the Gateway. This deployment simplifies the task of transferring compressed service images from the global Internet to the UMOBILE Domain. At the heart of the Service Manager you can find a decision engine responsible for the actual deployment of services.
- **Decision Engine (DE):** It is a piece of software with all the necessary logics to make decisions about service deployment and migration. The decision engine has access to monitors deployed strategically within the UMOBILE Domain and instrumented to collect metrics about conditions of interest. On this basis it makes decision about which services to deploy, when and where. For example, it creates a second replica of a given service in a neighbour Hotspot when the monitoring reports that the first replica is overloaded.
- **Monitors (M_1, M_2):** It is a software tool for collecting real time information about the status of the resources included in the UMOBILE Domain, such as network conditions and current usage of their Hotspot resources (CPU, memory and disk). Only two monitors are shown in Figure 18. A good example of a monitor is a

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

Python script deployed in the Raspberry Pi used to implement a Hotspot to measure and report its current free memory.

4.6.4 Validation of the service migration platform

Identification of the resource consumptions that impact QoS

These experiments are aimed at the fulfilment of objective 1, thus they demonstrate the proper functionality of the service migration platform and of its main components. Namely, we will validate the following components:

- The monitors (see M_1 and M_2 in Figure 18) used by the decision engine. The focus of this validation is on the correctness of the Pull-based model used by the monitors in metric collection.
- The actual deployment of the service over the NDN-based UMOBILE Domain. The focus here is on the validation of the Push-based model. Such a model is central since the service platform uses it to move a service image from its original location (for example, the Service Manager) to the selected Hotspot.

Regarding monitoring, we report the results of some experiments that measure current CPU load, memory usage and latency exhibited by a service deployed on a Raspberry Pi computer exposed to different work loads including exhaustion points.

Validate the scalability of number of Docker containers running on the UMOBILE hotspot (i.e., Raspberry Pi).

In this validation, we aim at finding the maximum number of containers running in parallel on a single Raspberry Pi. To scale up the containers, we consider that the Docker image could be prepared as small as possible to minimize memory footprint. Consequently, the memory allocation for kernel to handle a web server process can be optimized. For this, we use a nano web server image developed in assembly code in which size is less than 90 KB. We base our evaluation on memory consumption, CPU utilization and the creation time (time taken to create a container) by using *sysstat*.

The results for our optimized Docker are depicted in Figure 19-Figure 22 where we were able to scale up the number deployed containers to 2408. Specifically, memory usage is a key factor that limits the capability of running the containers on a PI. As shown in Figure 19, the memory usage increases gradually when a container is added. The initial memory usage before creating the first container was about 98 KB (9.89%). We hit the limit of 2408 containers where there is no space for available memory.

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

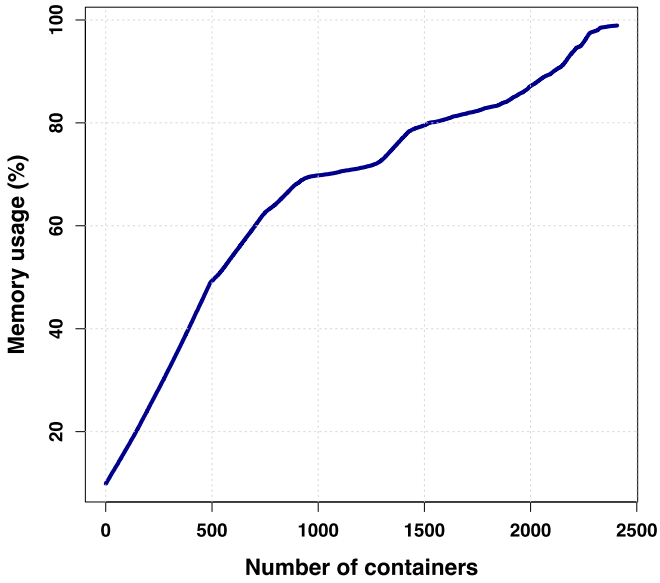


Figure 19: Measuring memory usage while scaling up the number of containers

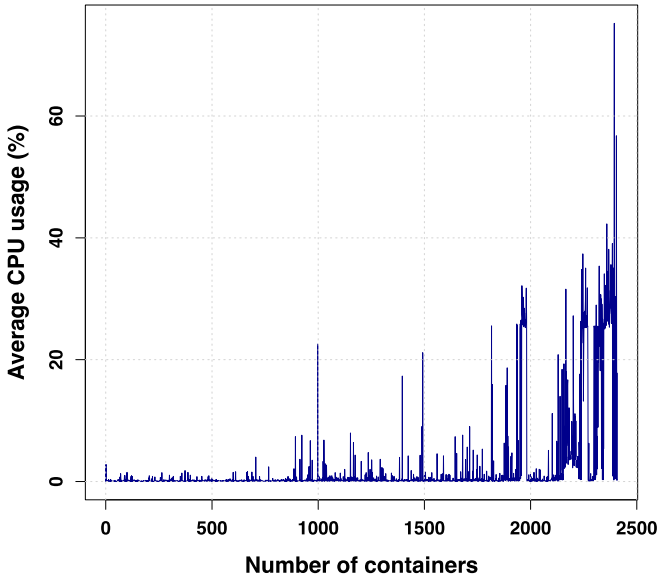


Figure 20: Measuring average CPU usage while scaling up the number of containers

Figure 20 shows the average utilization of the quad-core CPU of the Pi. Over the first thousand of deployed containers, the average CPU usage is very low (about 0.2%) with a few spikes. However as expected, the CPU usage increases significantly in the high load state (when the number of containers is larger than 2000). The Raspberry Pi consumes about 30% of CPU resources. The available space can be around 70% which is sufficient to provide multiple services.

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

Figure 21 depicts the results of creation time where each point denotes the time it took for the *n-th* container to start up. The creation time is varied from 0.62 s to 38.37 s depending on the current CPU load and memory usage. In addition, we also plot the cumulative creation time of the 2408 containers (Figure 22). The Pi spent approximately 1 hour and 50 minutes to spin out 2408 containers. On average each container requires 2.79 s to start up a web server.

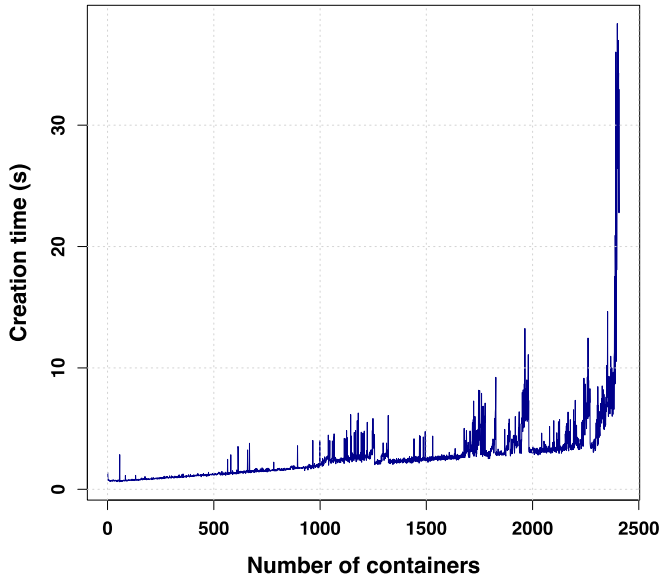


Figure 21: Measuring the creation time while scaling up the number of containers

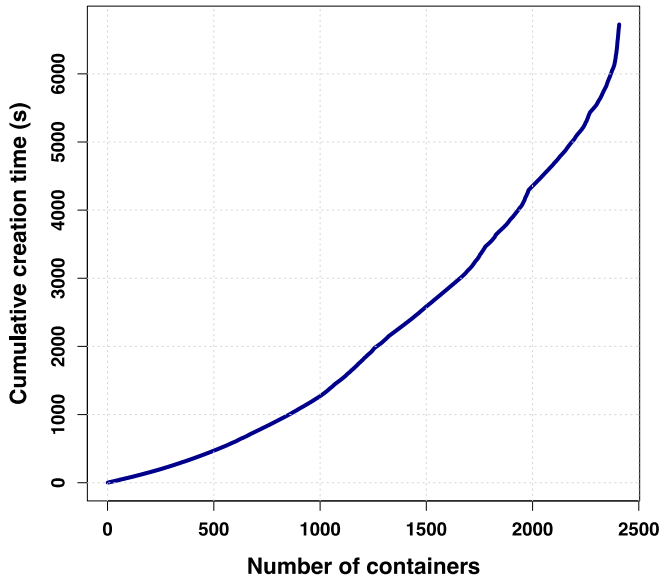


Figure 22: CDF of creation time while scaling up the number of containers

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

Validate the latency of web accessing with exhaustive resources

In order to investigate the feasibility of using lightweight containers as a base technology for the service migration, we aim to evaluate the scalability of each containers running on a single Raspberry PI while serving a large number of requests. The goal is to bring the CPU to its exhaustion point where its response to requests is of the order of hundreds of milliseconds or even seconds. We deploy a minimal static web server using httpd Docker base image using a similar configuration as the experiments in the previous section. The benchmarking scenario represents the UMOBILE operation where users can access the services provided by the UMOBILE hotspots through a wireless interface. Using the Ab - Apache HTTP server benchmarking tool[3], we conduct stress tests on the UMOBILE hotspot while scaling the number of concurrent users from 10 to 250. The total number of requests was set as 10000 transactions per experiment. For instance, in case of 10 users, 1000 requests were sent by each user.

Figure 23 illustrates the CDF of response time from the web container running on the Pi while varying the number of simultaneous users accessing the service. As shown in Figure 23, a single deployed Docker container is capable of serving a large number of concurrent users. As expected, the average response time increases when the number of con- current users is scaled up. Figure 24 and Figure 25 plot the CPU utilization and CPU load of the Pi using *sysstat* tools. The CPU utilization increases almost 20% when the number of users increases from 10 to 250. This increases the response time for the processes. Even though the utilization has not reached the capacity of CPU, processes still run slower as the CPU load increases. Figure 25 shows the average CPU load of the Pi (sampled in one min intervals). As the arrival rate of user requests increases, the amount of computational work need to process also increases. This has impact on the response of time (Figure 23).

[3] https://httpd.apache.org/docs/current/programs/ab.html

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

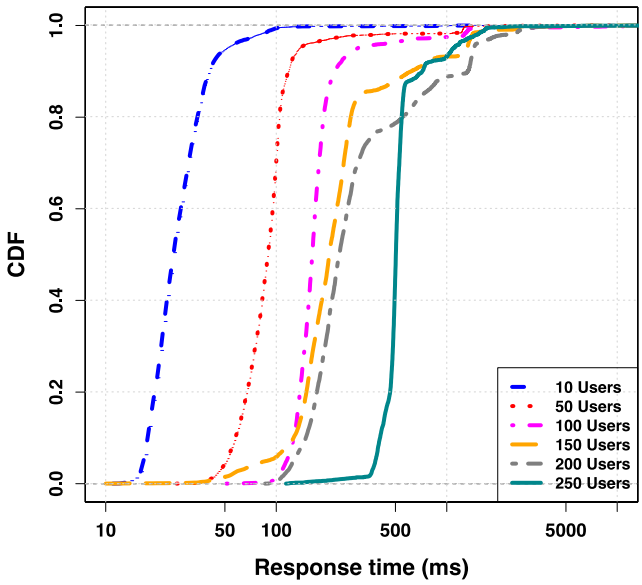


Figure 23: CDF response time with multiple users requests

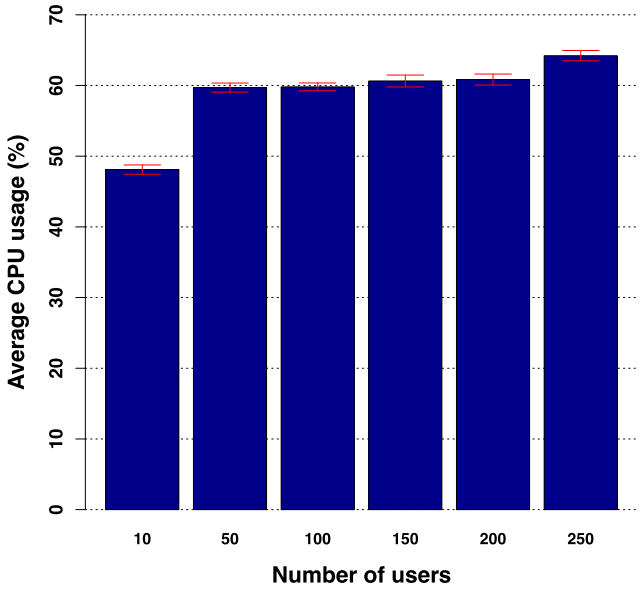


Figure 24: Average CPU usage with multiple users' requests

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

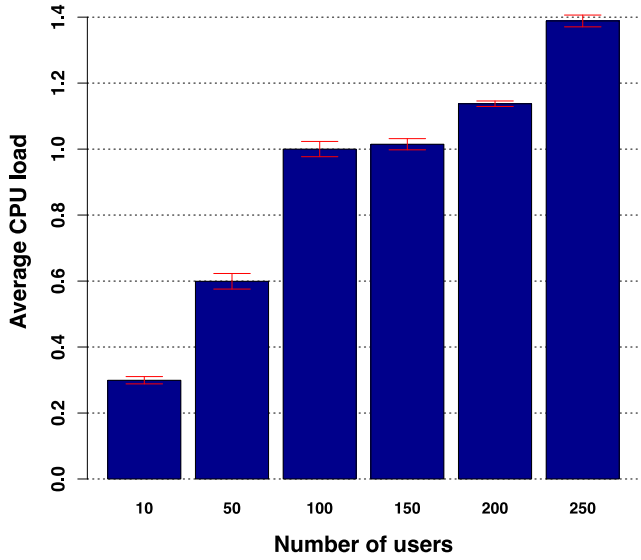


Figure 25: Average CPU load with multiple users' requests

Validating the monitoring system with pull communication

Now, we deploy monitors (Python or shell scripts) in strategic locations to collect metrics about the current status of the Hotspots and their network links and report their observances to the decision engine. The goal of this validation mainly presents the ability to retrieve information from each particular hotspot in UMOBILE network. In our scenario, the monitoring engine (collocated with service manager) sends an Interest packet with a compound name that contains two parts. The first part (prefix) is the named entity. In our case, it is */umobile/service_monitoring/.* The second part has the reserved name component ID which */hs1*. Such that the full name for the monitoring Interest in our case is */umobile/service_monitoring/hs1*.

Next we show an Interest message sent from the monitoring engine.

```
Sending Interest message: /umobile/service_monitoring/hs1
Sent Interest for umobile/service_monitoring/hs1
```

Following we show a log screen of hs1 when it replied monitoring Interest with monitoring Data.

```
Node-name:  hs1
Registering  prefix : /umobile/service_monitoring/hs1
Receive  Interest  message  name:  /umobile/service_monitoring/hs1
Number  of user = 1
Replied  DATA  to: /umobile/service_monitoring/hs1
```

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

Finally, we present a log screen of monitoring engine when it received the monitoring Data from hs1.

```
Receive monitoring data
Receive Data from SEG_1
data payload: 15.0/0.03

{'loadAvg': '0.03\n', 'memUsage': '15.0'}
Memory Usage: 15.0
Load Average: 0.03
```

The actual deployment of the service over the NDN-based UMOBILE Domain

The focus here is the validation of the Push-based model. Such model is central since the service platform uses it to move a service image from its original location (for example, the Service Manager) to the selected Hotspot. After receiving the monitoring Data, the decision engine, located inside manager, will decide to migrate the service to the AP with the lowest CPU load and memory usage. The service manager will apply Push-Publish data dissemination model (details in D3.1) to migrate the service to the selected hotspot.

Next we present the push message sent from service manager to the UMOBILE hotspot 1, i.e., a log screen of monitoring engine when it received the monitoring Data from hs1.

```
Sent Push Interest for /umobile/service_migration/push/SEG_1/rpi-nano-
httpd.tar
```

The first part (prefix) is the named entity. In this case, it is */umobile/service_migration/push.* The second part has the name of selected UMOBILE hotspot with the service name which is /SEG_1/rpi-nano-httpd.tar.

Following we show the log screen of the UMOBILE hotspot (SEG_1) receiving the push Interest message from the service manager. Consequently, the UMOBILE hotspot (SEG_1) sends subsequent Interest message with the service name mentioned in the previous push Interest message.

```
Receive Interest message name: /umobile/service_migration/push/SEG_1/rpi-nano-httpd.tar
Sending Interest message:/sm/service_migration/rpi-nano-httpd.tar
```

With the benefit of ICN, the UMOBILE hotspot will be able to fetch the service from any node that has the matching service in the cache. For simplicity, this validation performs the experiment with only two nodes that are the UMOBILE hotspot and Service manager. Therefore, the Interest message requesting the service is directly delivered to the service manager as presented in the log screen shown below. Consequently, the service manager

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

replies to the UMOBILE hotspot with the Data message containing the first chunk of data (e.g., */rpi-nano-httpd.tar/%00%00*). The UMOBILE hotspot automatically increases the data chunk id and sends the Interest message back to the service manager. Once the UMOBILE hotspot receives the first data chunk, it automatically increases the chunk id and sends back to the service manager. As shown in the log screen below, the service manager sends the second chunk of data *(e.g., /rpi-nano-httpd.tar/%00%01*) after receiving Interest message with increment chunk ID.

Following we present the log screen of service manager when it receives the Interest message requesting the service:

```
Receive Interest message: /umobile/service_migration/rpi-nano-httpd.tar , sending back DATA message
/pi/SM_NDN/SC_repository/rpi-nano-httpd.tar

Replied to Interest name: /umobile/service_migration/rpi-nano-httpd.tar/%00%00
Replied with Data name: /umobile/service_migration/rpi-nano-httpd.tar/%00%00

Receive Interest message: /umobile/service_migration/rpi-nano-httpd.tar/%00%01, sending back DATA message
/Users/adisorn/PycharmProjects/SM_NDN/SC_repository/rpi-nano-httpd.tar
Replied to Interest name: /sm/service_migration/rpi-nano-httpd.tar/%00%01
```

4.6.5 Use case mapping

Emergency scenario and civil protection

4.7 Smart routing and forwarding improved by the use of PerSense Mobile Light Information

4.7.1 Description

Current opportunistic networking algorithms are based on the frequency of encounters and the duration of encounters. It is our understanding that the usage of context information about users' roaming routines and users' behaviour such as visited networks, as well as affinity networks will improve the direct communication among devices, as well as the potential usage of open WiFi hotspots.

The demo will show how using data captured via PerSense Mobile Light and capturing its tracking information can evolve and complement smart routing in NDN-based opportunistic networking environments, namely environments supported by NDN-Opp. In particular, we will focus on the social routine scenario and we will define two showcases: a) one where messages are routed and forwarded towards a specific destination node. In this showcase, we'll make use of the Oi! application that runs on top of NDN-Opp; b) A second showcase will be defined where users can gather access to data by posting they interests. Posting of interests and data transmission will be handled by NDN-Opp in an opportunistic networking setting. This means that data will be forwarded

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

to interested devices by exploiting direct communication about them. In any of these two showcases, data exchange will be done by making use of hybrid routing metrics computed based on the contextual information gathered by PerSense Mobile Light tool. Data shall be transmitted via opportunistic communication provided by NDN-Opp, but based on a more complete set of users' context information provided by PerSense Mobile Light.

4.7.2 Demo Key Features

The above mentioned showcases will be developed based on a set of end-user devices implementing NDN-Opp (being developed by COPELABS): NDN-Opp will be able to run a smart routing algorithm to decide about the best neighbours to carry data that is intended to a specific node or to a set of nodes. Such smart routing will rely on metrics that are computed based on indicators provided by the UMOBILE contextual manager, which includes the PerSense Mobile Light module developed by Senception. Tecnalia will work on the definition of a hybrid mechanism that shall take into consideration the data captured by PerSense Mobile Light, and that shall apply it into alternative social-aware routing, thus allowing data to be transmitted via other than shortest path routing (e.g. by considering the wireless roaming context of individual users and providing matches that can assist in a "better" choice of successors), and in a way more suitable to the environments served by UMOBILE, e.g. support for intermittent Internet access. This aspect shall be worked jointly with COPELABS (social-aware routing), who shall also assist the integration of opportunistic communication via the NDN-Opp framework, based on WiFi and WiFi Direct, into NFD.

4.7.3 Hardware and Software Requirements & Experimental Settings

- Hardware and software requirements:
  - Hardware
    - 4 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 and PerSense Mobile Light
    - Oi! and Now@ applications (which run on top of NDN-Opp)
- Experimental Settings

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

- Creation of different settings (at least involving 20 smartphones and 3 different clusters)

4.7.4 Validation & KPIs

The validation will be performed using a series of tests emulating a real environment. Simulation tools will be as complimentary mean to the real environment tests, especially to validate the smart routing solution. There are two options being considered. The first one is the use the opportunistic networks simulator ONE. The second option is to build our own simulator better adapted for our particular purpose, in order to research and validate new smart routing parameters apart from the duration, parameter that was already identified.

The performance of the UMOBILE routing functionality will be mainly validated based on a real test-bed encompassing a set of personal mobile devices (smartphones), which will, in a first set of tests, communicate directly in a local network, by exploiting only existing WiFi direct connections.

In a second validation set, the same set of smartphones, deployed in a local wireless network (e.g. COPELABS facilities) will be able to exchange data, not only with local devises, as in the initial validation set, but also with remote personal mobile devices, located in another facility (e.g. UMOBILE testbed). The communication between COPELABS facilities and the UMOBILE testbed may be supported by the NDN global testbed (https://named-data.net/ndn-testbed).

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

- **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

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

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*.

While the previous evaluation sets aim to validate the UMOBILE routing functionality provided by NDN-Opp based on the contextual information provided by the PerSense Mobile Light, a further validation set may be used to evaluate the performance of the proposed routing mechanism. In this case the validation may be performed using a series of tests emulating a real environment. Simulation tools will be used as complimentary mean to the real environment tests, especially to validate the smart routing solution. There are two options being considered. The first one is the use the opportunistic networks simulator *One*. The second option is to build our own simulator better adapted for our particular purpose, in order to research and validate new smart routing parameters apart from the duration, parameter that was already identified.

4.7.5 Use case mapping

The validation of the UMOBILE end-user device developed based on NDN-Opp and the Contextual Manager (including the PerSense Mobile Light) 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

5. 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.

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 have explained the simulation methodology and the tools that are being used to evaluate the different parts of the solution through simulation. We are working on simulation right now and results will be provided in the next version of this deliverable, D5.2.

Finally, in Section 4 we have assessed 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. Therefore, we link each of the components with its real application. Further results of these validations will be provided in the next version of this deliverable, D5.2.

This deliverable is a first version that provides a guideline and a methodology for both the validation and the simulation. Both of them are a work in progress right now and they are being executed following what has been defined in this deliverable. D5.2, planned for M36, will contain the conclusions of this validation and simulation works.

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

6. References

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

[2] UMOBILE Project, "D.3.3 - UMOBILE ICN layer abstraction initial specification", February 2016

[3] UMOBILE Project, "D.4.1 – Flowlet Congestion Control (1)", January 2016

[4] UMOBILE Project, "D.2.2 - System and Network Requirements Specification", March 2016

[5] Christopher A. Wood , "Multipurpose IP/NDN Gateway and Bridge for Heterogeneous Network Interoperability", http://www.ics.uci.edu/~cs237/CS237preslide/woodc1_ndn_gateway.pdf, last accessed 21/01/2017

[6] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A survey of information centric 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 information centric 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] N. Spring, R. Mahajan, D. Wetherall and T. Anderson "Measuring ISP Topologies with Rocketfuel," IEEE/ACM Transactions on Networking (TON), 2004.

[13] G. Dan and N. Carlsson, "Power-law revisited: Large scale measurement study of p2p content popularity," IPTPS, 2010.

[14] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," ACM CoNEXT, 2009.

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

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

[16] F. Ekman, A. Keraenen, J. Karvo, and J. Ott, "Working day movement model," in ACM SIGMOBILE MobilityModels, 2008.

[17] Q. Inc, "Trepn Profiler - Qualcomm Developer Network," Feb. 2014. [Online]. Available: https://developer.qualcomm.com/software/trepn-power-profiler, last accessed 21/01/2017

[18] D. Camps-Mur, X. Perez-Costa, and S. Sallent-Ribes, "Designing energy efficient access points with WiFi direct," Computer Networks, vol. 55, pp. 2838 – 2855, 2011.

[19] "NS-3, A Discrete Event Simulator." http://www.nsnam.org.

[20] "Kickass for NS-3 - v. 1.0.", http://users.eecs.northwestern.edu/~mef294/projects/kickass/code/README-kickass-ns3, last accessed 21/01/2017

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

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

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