Action full title:

Universal, mobile-centric and opportunistic communications architecture

Action acronym:

UMOBILE

Deliverable:

D3.1. “UMOBILE architecture report (1)”

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
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, Democritus University of Thrace

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

Deliverable Information:

Deliverable Number-title	D3.1 UMOBILE architecture report (1)
WP Number	WP3
WP Leader	Sotiris Diamantopoulos
Task Leader	Sotiris Diamantopoulos
Authors	**COPELABS**: Paulo Mendes, Waldir Moreira **DUTH**: Sotiris Diamantopoulos, Christos-Alexandros Sarros, Dimitris Vardalis, Ioannis Komnios, Vassilis Tsaoussidis **UCL:** Ioannis Psaras, Sergi Rene **UCAM:** Adisorn Lertsinsrubtavee, Carlos Molina-Jimenez, Arjuna Sathiaseelan **COPELABS:** Paulo Mendes, Waldir Moreira, Luis Amaral Lopes **TECNALIA:** Iñigo Sedano, Susana Sanchez Perez **SENCEPTION**: Rute Sofia
Contact	diamantopoulos.sotiris (at) gmail.com , csarros (at) ee.duth.gr
Due date	17/11/2016
Actual date of submission	17/11/2016

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
1.0	31/05/16	Initial definition of the architecture
1.1	11/11/2016	Revised definition of the architecture – 1st draft to the consortium
1.2	16/11/2016	Revised definition of the architecture – 2nd draft to the consortium
1.3	17/11/2016	Revised definition of the architecture – Final version submitted to the EC

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

Executive Summary

Background: This report is written in the framework of WP3 "System and node architecture development" WP3" of the UMOBILE project. The deliverable aims to describe work on the UMOBILE core architecture and is accompanied by the implementation of the architecture so far and the documentation on the code.

Objectives: The core activity of WP3 is the design and implementation of the UMOBILE platform. Departing from the existing properties of DTN and ICN, we establish an architectural framework that extends connectivity options by being delay-tolerant and by exposing a common information-centric abstraction to applications.

UMOBILE aims 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 and free user-to-user networking.

The goal of this document is to provide a detailed description, along with a manual when applicable, of the new features, mechanisms and applications that have been developed so far as part of the UMOBILE architecture. Implementation code so far is also provided. The basis for UMOBILE platform is the Named Data Networking (NDN) architecture, one of the most promising ICN implementations, and UMOBILE features are being built in line with NDN. The result is a novel architecture that allows for new services and applications.

In the rest of the document, we present our vision and provide the full architectural picture, as well as the specific modules and their interconnections in the rest of the document.

NOTE (to be removed from the public version):
D3.1 provides the detailed view of the UMOBILE architecture, while D3.3 is a high-level description. Therefore, it is suggested that the reader/reviewer starts from D3.3 first, in order to get the high-level overview of the architecture and then go into the details in D3.1.

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 Definitions

Term	Meaning
AP	An access point (AP) is a networking hardware device that allows a WiFi compliant device to connect to a wired network.
Application	Computer software design to perform a single or several specific tasks, e.g. a calendar and map services.
BER	In digital transmission, the number of bit errors is the number of received bits of a data stream over a communication channel that have been altered due to noise, interference, distortion or bit synchronization errors. The bit error rate (BER) is the number of bit errors per unit time.
BP	Bundle Protocol (BP) defines the bundle as the core unit of the DTN architecture; a bundle is a series of data blocks that is routed in a store-and-forward manner between nodes over various transport networks.
BT	Bluetooth is a wireless technology standard for exchanging data over short distances from fixed and mobile devices, and building personal area networks (PANs).
CIT	Carried Interest Table is responsible for keeping up-to-date information concerning the data interests of the current node along with its social weights towards other nodes with whom it socially interacts.
CM	The Content Manager (CM) module in Oi! is responsible to manage all the messages to be sent, as well as received messages.
Content	Content refers to a piece of digital information that is disseminated and consumed by the end user equipment.
CS	A temporary cache of Data packets the router has received. Caching NDN Data packet helps satisfy future Interests for the same data faster. Various replacement strategies are implemented for the content store.
Data	Data is raw. Data is numbers that have no interpretation.
Data packet	In NDN, once the Interest reaches a node that has the requested data, the node will return a Data packet that contains both the name and the content, together with a signature by the producer's key which binds the two. This Data packet follows in reverse the path taken by the Interest to get back to the requesting consumer.

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

DM	Decision Maker is responsible for deciding whether replication should occur based on the level of social interaction towards specific interests, based on the SCORP algorithm.
DTN	Delay Tolerant Networking (DTN) supports interoperability of other networks by accommodating long disruptions and delays between and within those networks. DTN operates in a store-and-forward fashion where intermediate node can temporarily keep the messages and opportunistically forward them to the next hop. This inherently deals with temporary disruptions and allows connecting nodes that would otherwise be disconnected in space at any point in time by exploiting time-space paths.
EC	European Commission
E2E	In networks designed according to the end-to-end (E2E) principle, application-specific features reside in the communicating end nodes of the network, rather than in intermediary nodes, such as gateways, that exist to establish the network.
FIB	A routing table which maps name components to interfaces. The FIB itself is populated by a name-prefix based routing protocol, and can have multiple output interfaces for each prefix.
FN	Forwarding Node is responsible for routing requests for services towards the available copies in the service migration module.
Gateway	Gateway typically means an equipment installed at the edge of a network. It connects the local network to larger network or Internet. In addition, gateway also has the capability to store services and contents in its cache to subsequently provide localized access.
IBR-DTN	IBR-DTN is a framework for DTN applications; its module-based architecture with miscellaneous interfaces makes it possible to change functionalities like routing or bundle storage just by inheriting a specific class.
ICN	Information-Centric Networking (ICN) supports efficient delivery of both content and services by identifying information by name rather than the actual location. This decoupling of the information from its actual location breaks the need for end to end connectivity thus enabling much wider flexibility for efficient content and service retrieval. ICN also inherently supports caching thus enabling much better localised communications.
Information	Information is about understanding what the data is telling us. It provides an understanding about what is happening to users so we can make it easier for them to get the content they need when they need it.

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

Interest	A parameter capable of providing a measure (cost) of the "attention" of a user towards a specific content in a specific time instant. Users can cooperate and share their interests.
Interest packet	In NDN, a consumer puts the name of a desired piece of data into an Interest packet and sends it to the network. Routers use this name to forward the Interest toward the data producer(s).
NACK	A negative-acknowledge character (NAK or NACK) is a transmission control character sent by a station as a negative response to the station with which the connection has been set up.
NDN	Named Data Networking (NDN) is a Future Internet architecture that aims to transition today's host-centric network architecture into a data-centric network architecture. In particular, users will no longer need to retrieve data from a specific physical location; instead, users will be able to search for content, independent of the location where the content is stored.
NDN-CXX	NDN-CXX library (NDN C++ library with eXperimental eXtensions) provides the various common services shared between different NDF module.
NFD	The NDN Forwarding *Daemon*
Node	A wireless or wired capable device.
OPEX	An operating expense (OPEX) is an ongoing cost for running a product, business, or system. Its counterpart, a capital expenditure (CAPEX), is the cost of developing or providing non-consumable parts for the product or system.
OS	An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. The operating system is a component of the system software in a computer system. Application programs usually require an operating system to function.
PerSense	PerSense is an open-source sensing platform that senses the environment of a user and provides relevant services.
PIT	A table that stores all the Interests that a router has forwarded but not satisfied yet. Each PIT entry records the data name carried in the Interest, its incoming and outgoing interface.
RT	RT is the routing module in Oi! App

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

RTT	Round-trip time (RTT) is the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgment of that signal to be received. This time delay therefore consists of the propagation times between the two points of a signal.
SC	Service Controller manages the mapping of publishers of services and subscribers of services in the service migration module.
SEG	Service Execution Gateway is the point of attachment for clients in the service migration module.
Service	Service refers to a computational operation or application running on the network which can fulfil an end user's request. The services can be hosted and computed in some specific nodes such as servers or gateways. Specifically, services are normally provided for remuneration, at a distance, by electronic means and at the individual request of a recipient of services. For the purposes of this definition; "at a distance" means that the service is provided without the parties being simultaneously present; "by electronic means" means that the service is sent initially and received at its destination by means of electronic equipment for the processing (including digital compression) and storage of data, and entirely transmitted, conveyed and received by wire, by radio, by optical means or by other electromagnetic means; "at the individual request of a recipient of services" means that the service is provided through the transmission of data on individual request. Refer to D2.2 for further details.
SLA	Service-level agreement, a contractual agreement on the level of service to be provided by a service provider to a customer.
Social trust	Trust which builds upon associations of nodes is based on the notion of shared interests; individual or collective expression of interests; affinities between end users.

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

1. Introduction

The main objective of UMOBILE is to develop a mobile-centric, service oriented architecture that efficiently delivers content and services to end-users. By efficiently we mean that content/services are reliably available in spite of impairments of the communication infrastructure, and with the expected quality of service. UMOBILE decouples services from their origin locations, shifting the host-centric paradigm to a new paradigm, one that incorporates aspects from both information-centric and opportunistic networking with the ultimate purpose of delivering an architecture focused in: i) improving aspects of the existing infrastructure (e.g., keeping traffic local to lower delays and OPEX); ii) improving the social routine of Internet users via technology-mediated approaches; iii) extending the reach of services to areas with little or no infrastructure (e.g., remote areas, emergency situations).

UMOBILE aims to push network services (e.g., mobility management, intermittent connectivity support) and user services (e.g., pervasive content management) as close as possible to the end-users. By pushing such services closer to the users, we can optimize, in a scalable way, aspects such as bandwidth utilization and resource management. We can also improve the service availability in challenged network environments. For example, users in some areas may suffer from intermittent and unstable Internet connectivity while they are trying to access Internet services.

To achieve ubiquitous, local and edge-based services, the proposed UMOBILE architecture combines two emerging architecture and connectivity approaches: Information Centric Networking (ICN) and Delay Tolerant Networking (DTN). The aim is to build an architecture that defines a new service abstraction that brings together both information centric as well as delay tolerant networking principles into one single abstraction.

In this document, we describe the UMOBILE architecture in detail. We place special focus on the new modules developed in order to:

- Achieve the integration of ICN and DTN, to enable delay-tolerant and opportunistic communications in an information-centric framework.
- Allow for new services at the edge of the network.
- Assist in usage contextualisation to develop new types of applications.
- Consolidate the architecture with novel congestion control, QoS and routing mechanisms.

We analyse the individual components in the following sections. Our goal is to integrate all modules into a unified UMOBILE platform that will provide the aforementioned functionality.

As detailed at D3.3, the high-level description of the UMOBILE architecture, the proposed UMOBILE architecture consists of two main parts, one of elements in the fixed part of the network (e.g., routers, gateways, access points) and one for mobile devices (focusing mainly on Android OS at the moment).

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

A prototype of the platform will comprise:

1. The UMOBILE Access-Point/Router

The UMOBILE Access-Points/Routers are devices deployed in the fixed part of the network or onboard mobile nodes (e.g., UAVs), enabling a multitude of network-related operations. In particular, these devices employ the UMOBILE platform to cache content, execute services, act as proxies to the Internet, allow for granular QoS and QoE application interfaces, ensure data delivery even under adverse network conditions, etc.

2. The UMOBILE Smartphone *daemon*

This is an application, running as a *daemon* on end-user devices such as smartphones, also enabling multiple network-related operations. More specifically, this *daemon* employs the UMOBILE platform to integrate modules to opportunistically transfer between nodes based on social metrics, an application sharing framework, a module to infer connectivity patterns, delay-tolerant networking to increase reliability, along with smart routing and forwarding engines. Applications can exploit the UMOBILE Smartphone *daemon* to provide novel communication operations.

3. The UMOBILE Service

The UMOBILE Service exploits both the UMOBILE Access Points and Smartphone *daemons* to provide interested parties with an innovative communication paradigm. The UMOBILE Service can be employed, among others, to support communications in emergency situations, and to improve the daily routine of end-users. It can be employed in different scenarios.

Apart from describing all UMOBILE architectural components, the present deliverable also includes:

- Code on the integration of NDN and IBR-DTN (available in zip file, documentation is included);
- Code on the migration of a simple Web server;
- Code of the Oi! Instant Messaging application (available in GitHub);
- Code of the SOCIO routing framework (available in GitHub);
- Guidelines to download and test the PerSense Mobile Light application, and
- Technical report on Oi! and SOCIO (available in zip file).

The document is organised as follows:

- Section 2 provides an insight to the vision of UMOBILE and highlights how each UMOBILE feature contributes to this vision,
- Section 3 provides an introduction to Named Data Networking, as the starting point of the UMOBILE architecture.
- Section 4 presents the full architectural picture, along with the new services, mechanisms and apps developed as part 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

- Section 5 concludes the document.

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

2. UMOBILE Vision

In D3.3, we described the UMOBILE architecture and highlighted the key services that the proposed architecture will provide. Figure 1 (extracted from D3.3) provides an overview of the UMOBILE platform, in terms of actors and connectivity options involved.

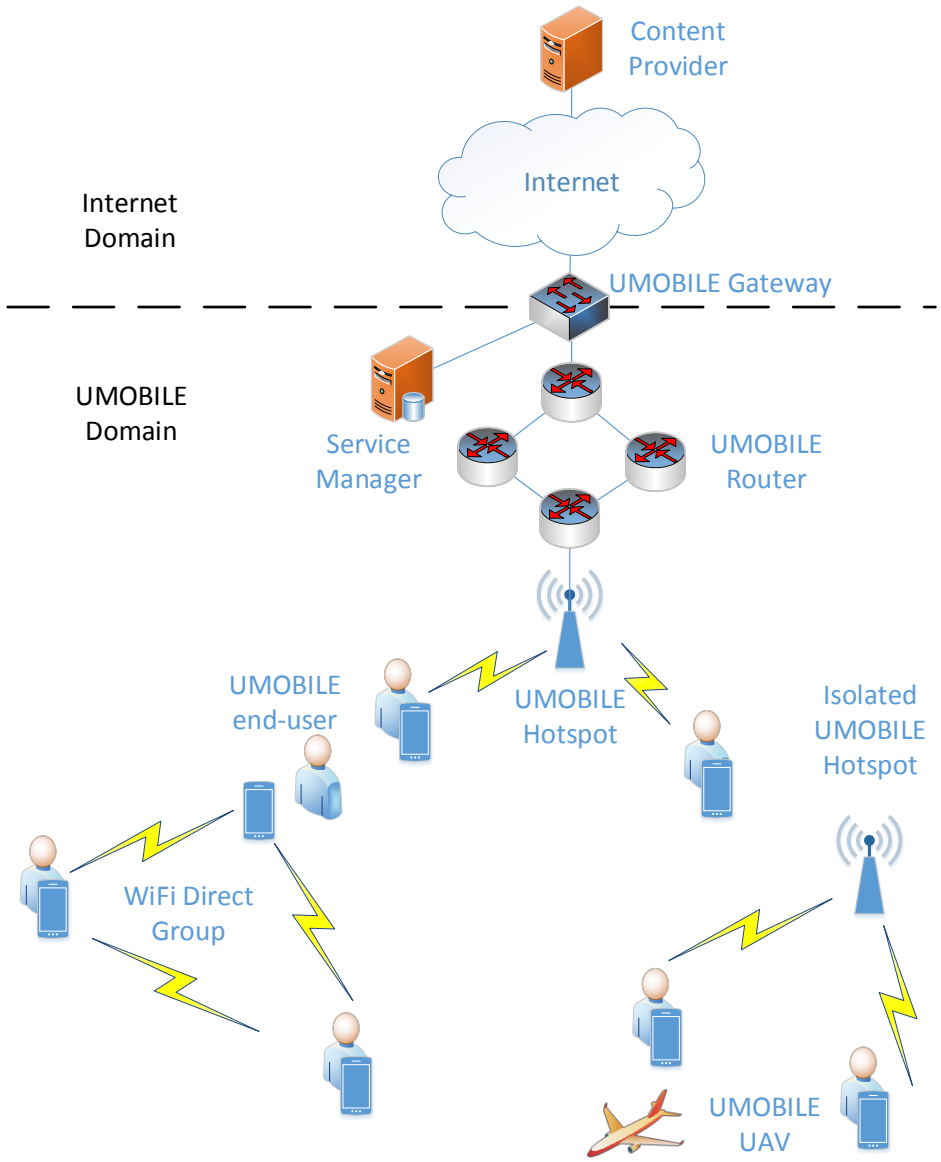

Figure 1: High-level design of the UMOBILE architecture

The actors involved in the UMOBILE architecture include:

- UMOBILE-enabled mobile devices (i.e., smartphone, tablet), used to send and receive participatory data (e.g. photos, short messages) as well as opportunistic data (e.g. atmospheric pressure, temperature, noise, roaming patterns).
- UMOBILE-enabled hotspots able to collect and relay relevant information (e.g., alert messages, instructions from emergency authorities), host some instantiated services or

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

store collected data, check its validity and perform computational functions to increase the value of the information to the civil authorities.
- UMOBILE-enabled UAV devices or other vehicles able to collect and relay relevant information and connect two isolated areas.
- UMOBILE-enabled gateways/proxies that provide interconnectivity between UMOBILE domain and the Internet domain.
- UMOBILE Routers that use UMOBILE protocols to support Service migration and the QoS mechanisms implemented in this project.

The aim of the UMOBILE project is to provide an architecture that merges information-centric networking with delay-tolerant networking in order to efficiently operate in different network situations, reaching disconnected environments and users and providing new types of services. These services can be exploited from a variety of applications (examples of which include opportunistic chat and news apps). Having already described in detail the envisioned UMOBILE services in D3.3, in this deliverable we focus on the specific enhancements and modifications of the NDN platform that are required to provide these new services.

In particular, we build the UMOBILE architecture using the NDN architecture as our starting point and:

- Build a new delay-tolerant face in the NDN architecture. This DTN face is used in opportunistic and disruptive scenarios and delivers packets to a Bundle Protocol implementation (IBR-DTN[1]), which is then responsible for their delay-tolerant transmission;
- Develop a novel framework (NDN-Opp) that natively enables opportunistic communications over NDN and supports social-aware routing. In this context, a new WiFi Direct face is introduced and two new applications are created: a chat app and a news app;
- Provide two new types of services that can be used in a variety of scenarios: a service migration platform that can move services closer to the edge of the network where they can be accessed significantly faster, even in cases when the remote server is not available and a push service for cases when the user needs to push data to the network (e.g. in cases of emergency);
- Propose a new application-centric naming framework (KEBAPP) [9], where applications share common name-spaces and further support the use of a keywords;
- Employ novel mechanisms to cope with extreme situations (e.g. disaster cases): an opportunistic off-path content discovery mechanism that exploits cached content in both in-network caches and end-users' devices to facilitate content retrieval even when the original server is not available, as well as a name-based replication priorities scheme (NREP)[4] that favours spreading of the most important messages when normal communications are disrupted and traffic is increased.

[1] https://www.ibr.cs.tu-bs.de/projects/ibr-dtn/

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

- Introduce a new congestion control algorithm (INRPP)[15] which pools bandwidth and in-network cache resources in a novel congestion control framework to reach global fairness and local stability.
- Develop a new smart routing framework that also takes into consideration user context and behaviour.
- Introduce the contextual manager which is based on the PerSense Mobile Light tool, as a UMOBILE module that captures data from the user environment (network mining) and that shall have as output routing metrics as well as user/usage recommendations suitable for e.g. applications or service management.

An overview of the different components and their contributions to the architecture is given in the image below:

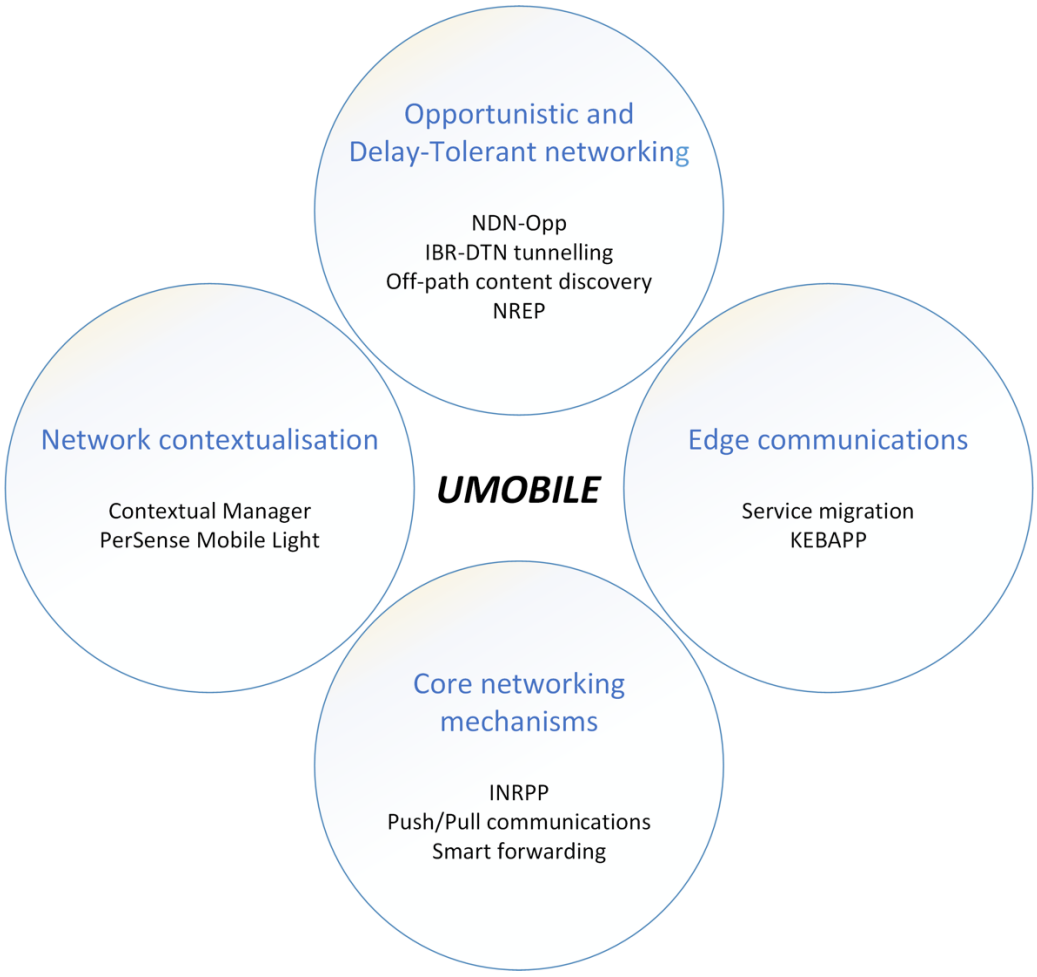

Figure 2: UMOBILE architecture conceptual overview

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

3. UMOBILE starting points

The aim of UMOBILE is to build an architecture that defines a new service abstraction, bringing together information-centric, delay-tolerant, and opportunistic communications principles into one single abstraction. To achieve that, UMOBILE uses Named Data Networking (NDN) - one of the most promising ICN implementations - as a starting point. We modify and enhance the NDN architecture, in order to support delay-tolerant and opportunistic communications naturally, building an integrated platform that facilitates communications in disruptive environments at the edge of the network.

In this Section, we provide a brief description of NDN, which constitutes the basis for the UMOBILE architecture.

Named Data Networking (NDN)

Named Data Networking (NDN)[2] is a Future Internet architecture that aims to transition today's host-centric network architecture into a data-centric network architecture. In particular, users will no longer need to retrieve data from a specific physical location; instead, users will be able to search for content, independent of the location where the content is stored. NDN changes the semantics of network service from delivering the packet to a given destination address to fetching data identified by a given name; NDN is a part of the broader information-centric networking approaches. In ICN approaches, data becomes independent from location, application, storage, and means of transportation, enabling in-network caching and replication.

Our initial approach was to introduce a "content layer" that intercepts communication, produces unique location-independent names for requested content and stores the latter within the network according to sophisticated caching policies. This way, the first stage of communication between the user and the content source, i.e., the content resolution stage, follows the approach of the current Internet and uses search engines, with the URL containing the name of the primary content server, while later stages of communication are based on the location-independent names for requested content.

However, in the UMOBILE project we focus on the mobile part of the network, where Internet nodes are mobile and connectivity can be disrupted, e.g., in an emergency scenario where the network might get fragmented and communication takes place in an ad hoc manner between mobile devices. In this scenario, we have to overcome the difficulties of a host-centric and connection-oriented communication paradigm. That said, we have come to realise that our initial plans for operating on top of an extra content-layer would not provide huge benefits, but would rather recycle old problems with regard to mobility support.

Thus, we decided to rely on pure ICN features to develop our UMOBILE architecture. Such features include flexibility, host multihoming, content name consistency and resiliency and

[2] https://named-data.net

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

provide a solid base to build an architecture for mobile, infrastructureless or delay-tolerant networks.

Among all ICN architectures, we decided to use NDN as the baseline. NDN is the most promising ICN architecture with more participants in the ICN community, and with more active projects implementation-wise. We think NDN is a perfect candidate as a starting point for our UMOBILE architecture.

To provide compatibility with existing mobile user devices, we devise the UMOBILE architecture 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. Also, in order to provide connectivity between any opportunistic UMOBILE device and the fixed network (IP network), we devise a UMOBILE Proxy/Gateway able to translate interest packets to HTTP requests and vice versa.

Communication in NDN is driven by receivers i.e., data consumers, through the exchange of two types of packets: Interest and Data. Both types of packets carry a name that identifies a piece of data that can be transmitted in one Data packet.

- **Interest**: A consumer puts the name of a desired piece of data into an Interest packet and sends it to the network. Routers use this name to forward the Interest toward the data producer(s).
- **Data**: Once the Interest reaches a node that has the requested data, the node will return a Data packet that contains both the name and the content, together with a signature by the producer's key which binds the two. This Data packet follows in reverse the path taken by the Interest to get back to the requesting consumer.

The core of the NDN architecture lies in a network forwarder, the NDN Forwarding *Daemon* (NFD). NFD is a modular and extensible forwarder that implements the forwarding of both Interest and Data packets. To achieve that, it abstracts lower-level network transport mechanisms into NDN Faces that provide the necessary abstraction on top of various lower level transport mechanisms.

Several data structures are used to support forwarding of NDN Interest and Data, the most important being:

- **Content Store (CS)**: A temporary cache of Data packets the router has received. Caching NDN Data packets helps satisfy future Interests for the same data faster. Various replacement strategies are implemented for the content store.
- **Pending Interest Table (PIT)**: A table that stores all the Interests that a router has forwarded but not yet satisfied. Each PIT entry records the data name carried in the Interest, its incoming and outgoing interface.
- **Forwarding Information Base (FIB)**: A routing table which maps name components to interfaces. The FIB itself is populated by a name-prefix based routing protocol, and can have multiple output interfaces for each prefix.

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

NDN-CXX library (NDN C++ library with eXperimental eXtensions) provides the various common services shared between different NDF module. On top of NDN-CXX, NDN supports:

- NDN Common Client libraries that provide APIs for Python, C++, Java and JavaScript,
- A routing module that supports conventional routing algorithms such as link state and distance vector adapted to route on name prefixes and
- A repository for persistent storage.

Figure 3 depicts the high-level design of NDN architecture.

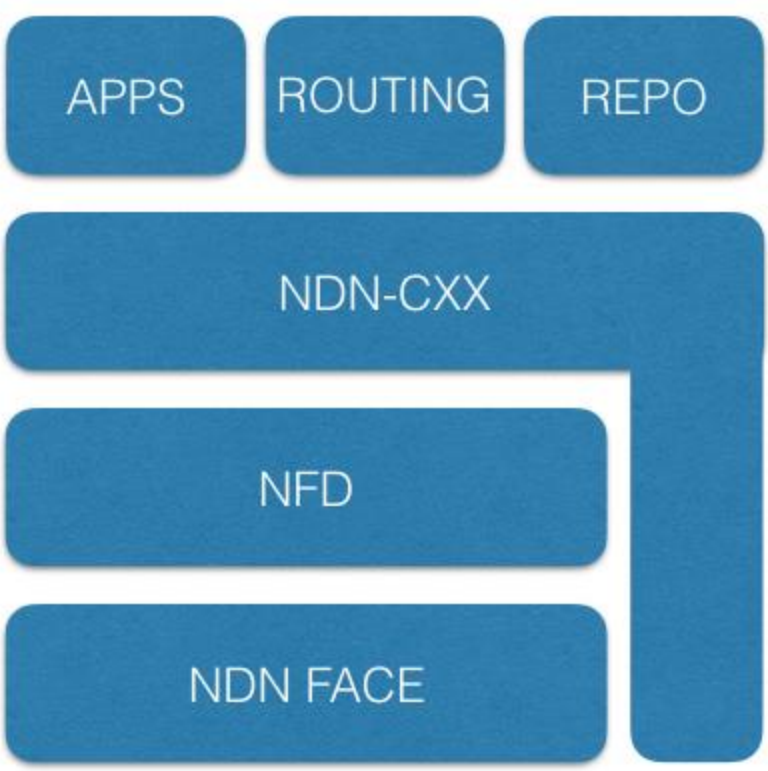

Figure 3: Conceptual depiction of Named Data Networking architecture

More information on NDN architecture can be found on Named Data Networking website: http://named-data.net.

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

4. UMOBILE architecture

The UMOBILE architecture builds on the NDN platform. We have modified and expanded the original NDN platform to enable new services while focusing on the edge of the network, especially on opportunistic mobile communications. This way all communication opportunities are exploited, achieving reliability and resilience even against extremely challenging network conditions where availability of a central server is not to be taken for granted.

A new class of service is introduced and QoE for the end-user is improved by service localisation as well as via user, usage and network contextualisation. In addition to the aforementioned benefits, contextualization assists in the development of new strategies, e.g. "smart" forwarding strategies which shall consider not only the network context, as well as the user networking context to improve the network operation.

Our focus on the mobile part of the network does not mean that we do not consider the fixed – typically wired – part. On the contrary, we exploit the properties of Information-Centric and Delay-Tolerant Networks to design novel congestion control mechanisms, QoS schemes and make relevant advancements to improve core network operation.

Our starting point is – as already noted – the original NDN architecture. Below, we provide the modules present on the original NDN platform along with their interrelations.

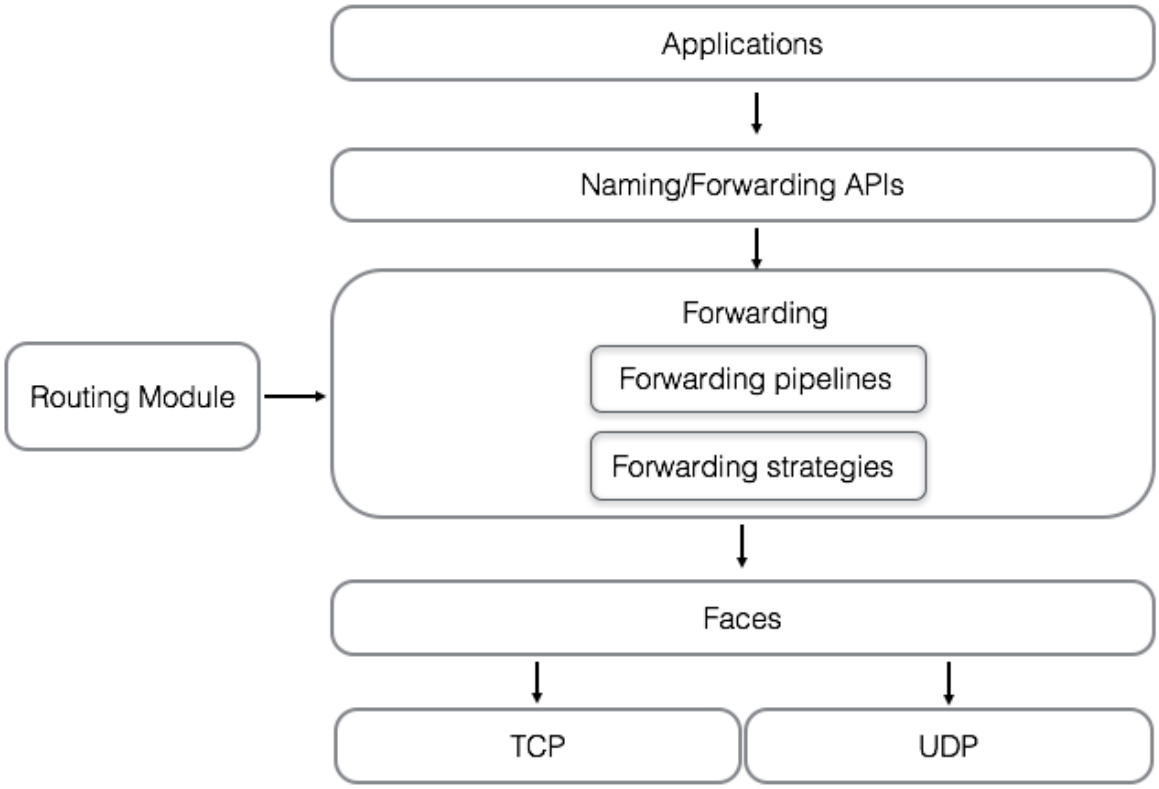

Figure 4: NDN architectural modules

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

Software-wise, the UMOBILE platform extends NDN providing support for decentralised services and opportunistic communications. The full architectural diagram of the UMOBILE platform is provided in Fig. 5. Green boxes depict new components implemented in the framework of the UMOBILE project, whereas yellow ones depict modified components of the original NDN platform.

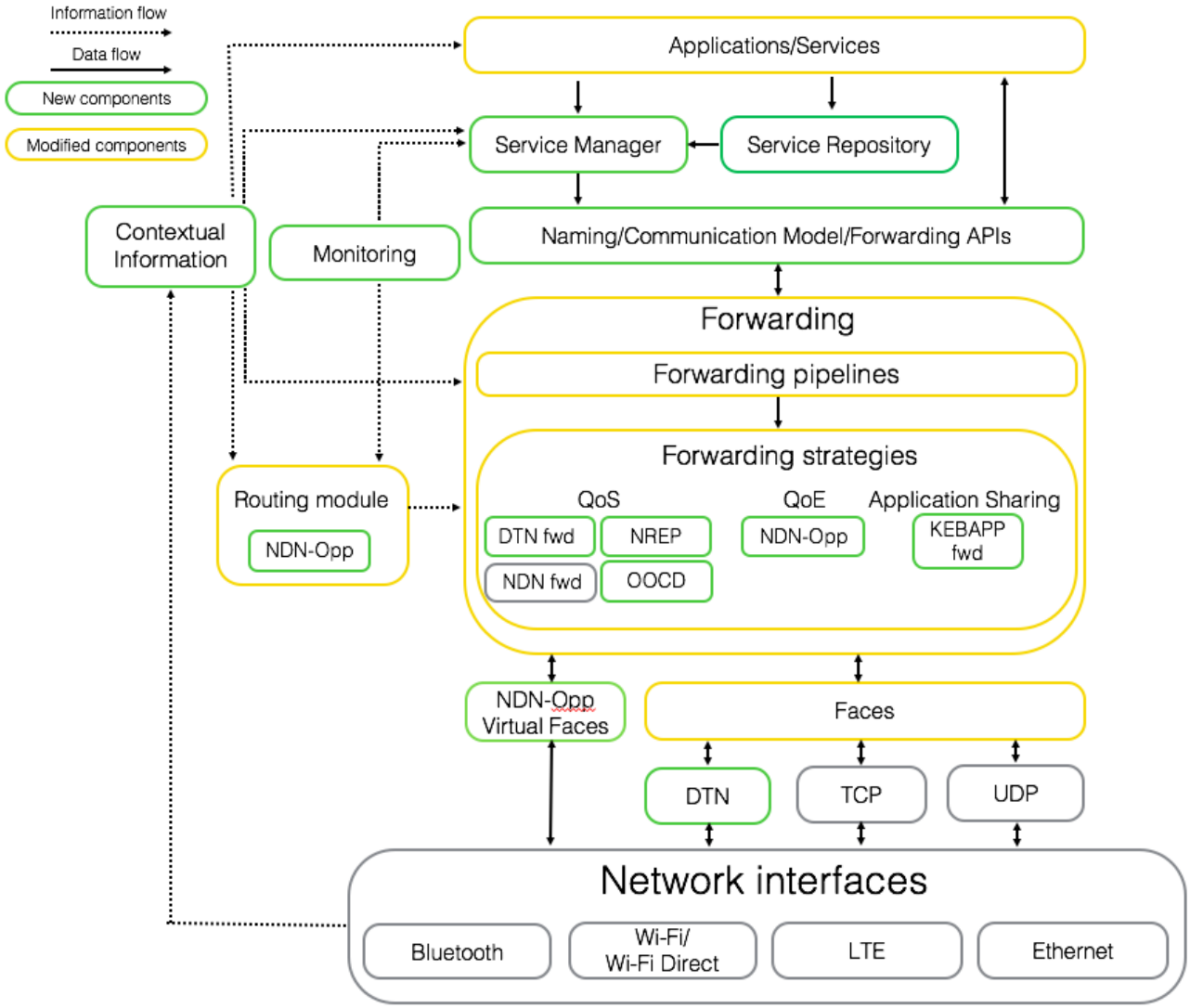

Figure 5 : UMOBILE architectural modules

New and updated software modules include:

a) New forwarding mechanisms

- DTN forwarding engine

  We have implemented and integrated a DTN interface into the UMOBILE platform to enable the forwarding of packets through DTN tunnelling. DTN forwarding provides

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

enhanced resilience in less than ideal networking conditions. It can also be employed as a QoS class.

- NDN-Opp

  NDN-Opp is enabling opportunistic communications in NDN. NDN-Opp provides an updated version of the NDN Android implementation. It is based on social connections between nodes, being able to forward packets in cases where typical NDN would fail. For this proposes NDN-Opp makes use of the existing Best Route forwarding strategy.

- Name-based Replication Priorities (NREP)

  NREP prioritises replication of packets based on their naming characteristics.

- Off-Path Content Discovery

  Off-Path Content Discovery extends typical NDN forwarding providing alternative content sources.

- Keyword-Based Application Sharing (KEBAPP) forwarding engine

  We are developing a forwarding strategy to enable users to receive application-related data locally shared by other users.

These forwarding strategies are developed in the framework of T3.3. NREP is developed in the framework of T4.3.

b) New northbound APIs

- Contextualization – PerSense Mobile Light

  PerSense Mobile Light module provides applications with contextual and behavioral information, enabling the deployment of a new type of services, such as daily routine improvement.

- Push services

  UMOBILE extends the NDN communication model by supporting push services. Based on this type of service, applications now are able to produce and forward content without the need to receive an Interest packet first. This functionality can be particularly useful in the implementation of emergency applications.

- KEBAPP

  KEBAPP is an application sharing framework that enables the discovery of locally available application data, without the need to contact a centralized service.

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

c) A multi-plane QoS mechanism

A multi-planar QoS mechanism is being developed in the framework of the UMOBILE project. At the application level, the service migration framework is employed to move services closer to the end-users when necessary. Complimentary, at the network level, INRPP takes advantage of in-network caching and multiple path availability to mitigate congestion to observe QoS requirements.

d) A smart forwarding module

The current implementation of the UMOBILE architecture for the smartphone *daemon* makes use of the NDN-Opp routing engine to decide about which wireless neighbours should get a packet copy. The first algorithm used by NDN-Opp is the Time-Evolving Contact Duration approach [1] used by the dLife protocol [2,3]. In the next version the NDN-Opp routing engine will integrate a novel routing scheme able to exploit the context information made available by the Contextual Management module.

Moreover, a smart routing/forwarding module, currently under design in the framework of T3.3 along with the forwarding strategies mentioned above, is selecting the optimal forwarding strategy, based on contextual information and application requirements.

In the following sub-sections, we describe the different functionalities provided by the UMOBILE platform. Special focus is being placed on the advancements introduced within the scope of the project.

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

4.1 Forwarding

Forwarding in UMOBILE follows typical NDN forwarding, by enhancing it with new forwarding strategies and modifications that allow for delay/disruption tolerant and opportunistic communications. We aim to provide support for challenging network scenarios with intermittent connectivity, by enabling D2D communications over WiFi Direct and Bluetooth links, or even remote WiFi hotspots.

We note that forwarding plays a special role in the UMOBILE architecture. In traditional IP routing, routers exchange updates and select the best routes to construct the forwarding table (FIB) but the forwarding plane by itself strictly follows the FIB. This results in a "smart routing/dumb forwarding" approach that places the responsibility for data delivery on the routing.

In UMOBILE, and following the NDN architecture, this relationship is changed. While routing serves the same purpose (to compute routing tables to be used in forwarding NDN's Interest packets), the forwarding plane is now stateful: the routers keep state of the pending Interests to guide Data packets back to requesting consumers. This allows for an intelligent forwarding plane, as by recording pending Interests and observing Data packets coming back, each NDN router can measure packet delivery performance and utilise multiple alternative paths to improve performance. As a result, much of the routing functionality is offloaded to forwarding Interest packets and the routing plane only needs to disseminate long-term changes in topology, without having to deal with short-term churns.

Forwarding strategies are the decision makers in the forwarding *daemon*, deciding whether, when and where to forward the Interests. A per-namespace selection of the appropriate strategy is available, to fulfil the needs for different applications that utilise different naming schemes. By utilising different naming schemes we can invoke different strategies and provide support for multiple applications.

Below we detail the modules included in the UMOBILE architecture that are related to the forwarding plane. Our major focus is on extending the reach of the communication paradigm, by enabling and facilitating delay-tolerant and opportunistic communications that are currently poorly supported by the NDN paradigm.

4.1.1. DTN tunnelling

DTN is an emerging technology to support a new era in internetworking and interoperable communications, either on Earth, or in Space. Like IP, DTN operates on top of existing network architectures, creating a DTN overlay. In particular, DTN extends internetworking in the 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:

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

the model is also termed store-carry-forward. This inherently deals with temporary disconnections or disruptions and allows the connection of nodes that would else be disconnected in space at any point in time.

Key design assumptions of Internet protocols such as short round-trip time (RTT), absence of disruptions and continuous E2E path availability are challenged by such a concept.

The core of the DTN architecture lies in the Bundle Protocol (BP). BP defines the bundle as the core unit of the DTN architecture; a bundle is a series of data blocks that is routed in a store-and-forward manner between nodes over various transport networks. In order to improve the efficiency of transfers, the DTN architecture supports several features, such as fragmentation and custody transfer. DTN fragmentation and reassembly ensures that contact volumes are fully utilised, avoiding retransmission of partially-forwarded bundles. Bundle fragmentation can be performed either proactively at the sender or reactively at an intermediate node, if the reduction of the size of a bundle is required to forward it. Reassembly can be performed either at the destination or at some other node on the route to the destination. Custody transfer is another important feature of the DTN architecture. In essence, custody transfer moves the responsibility for reliable delivery of a bundle among different DTN nodes in the network. Whenever a node accepts the reliable delivery responsibility of a bundle, it is called the "custodian" of this bundle.

As far as the existing DTN reference implementations are concerned, several implementations have been developed during the last years, each targeting different applicability scenarios. 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 change functionalities like routing or bundle storage just by inheriting a specific class.

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: it enables a whole set of new communication capabilities when necessary. In particular, DTN enhances network operation by providing:

i) A new service class, namely less-than-best-effort. This shall support services with non-stringent QoS requirements.
ii) Reliability to services in congested environments (e.g. in emergency cases)
iii) A means of congestion control, by reactively offloading traffic to DTN when congestion is detected.
iv) A means of congestion avoidance, by proactively scheduling and/or shaping traffic to reach its destination only when links are underutilized.

Architecturally, the DTN face is the module interconnecting the UMOBILE platform with the IBR-DTN *daemon*, as depicted in Fig 6.

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

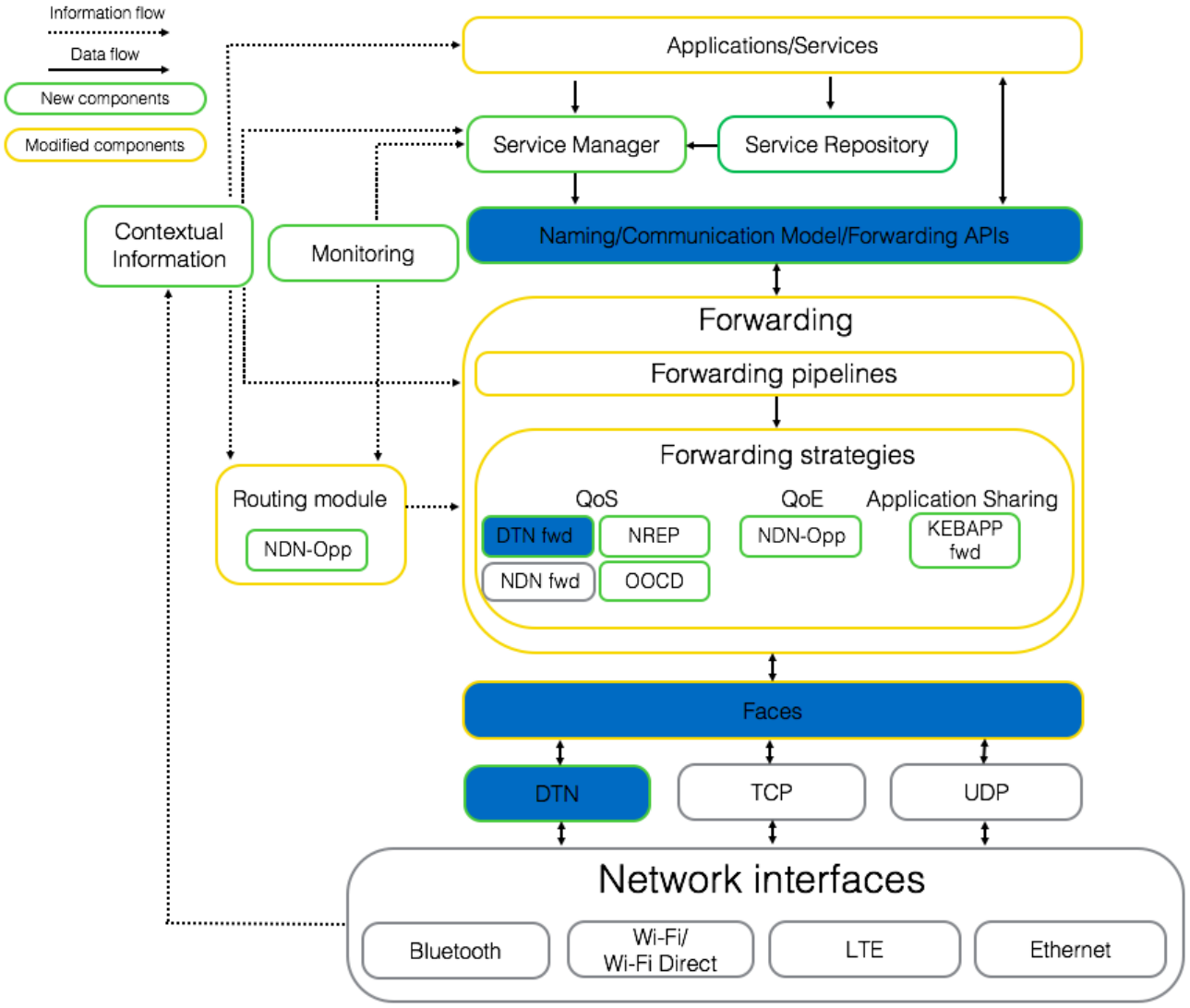

Figure 6. NDN-DTN tunnelling

Focusing on the DTN face module, its internal structure is depicted in Fig. 7.

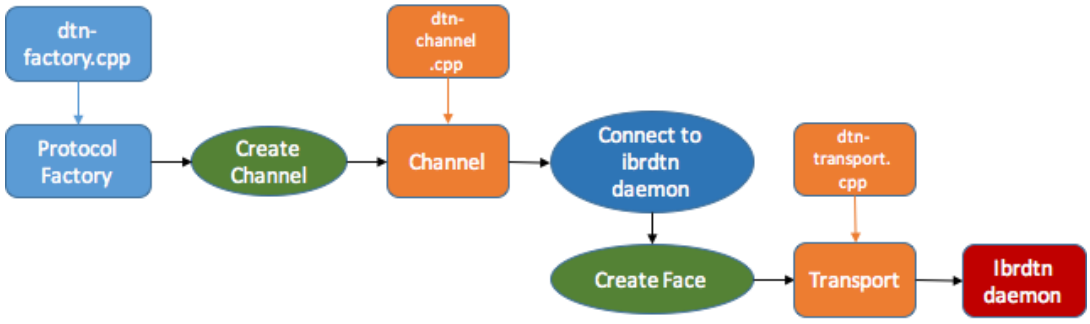

Figure 7. Internal structure of the DTN face

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

The DTN face is employed in the forwarding plane as follows:

- In the non-infrastructure part of the network, it will be part of other forwarding strategies, so that a node can automatically opt for a DTN tunnel to opportunistically reach a more "fixed" node (e.g. an access point in the vicinity) instead of relying on typical NDN over TCP/UDP forwarding, when it senses via the routing module that there is too much network volatility around. This functionality is depicted in Fig. 8

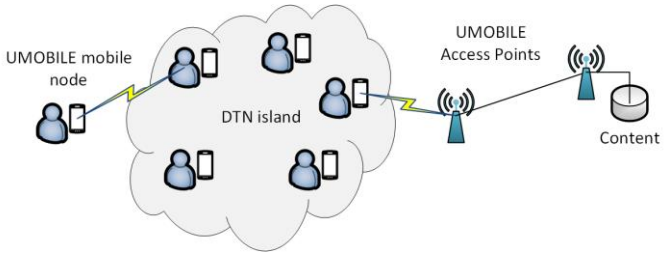

Figure 8. DTN forwarding in volatile networking environments

- It will constitute a separate forwarding strategy, which will be employed a priori by applications when network delays are to be expected (e.g. there is content delivery through UAVs), in an opportunistic or infrastructure network context. This functionality is depicted in Fig. 9

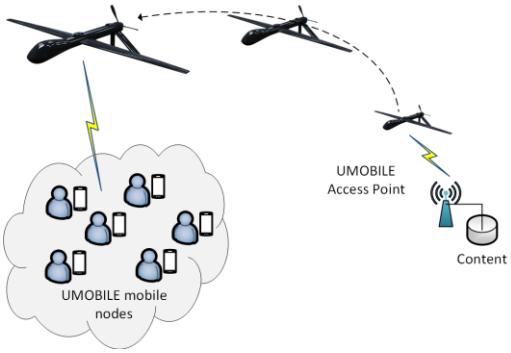

Figure 9. DTN forwarding accommodating delays inflicted by a UAV flying pattern

- It will be a part of a QoS scheme supporting a) services with non-stringent QoS requirements (lower-than-best-effort QoS class), as depicted in Fig. 10, or b) services that prioritise reliability in congested environments (e.g. in emergency cases), depicted in Fig. 11.

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

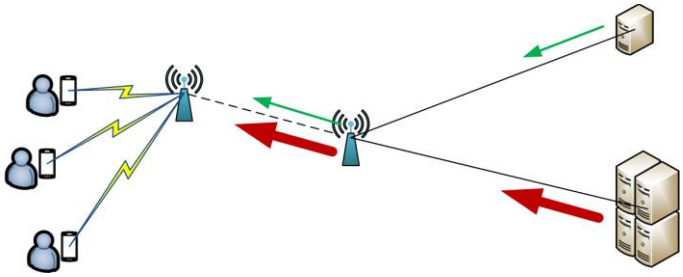

Figure 10. DTN forwarding providing a lower QoS class

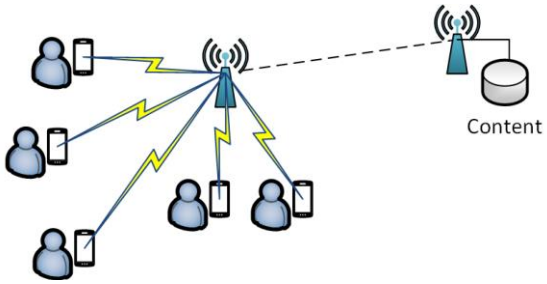

Figure 11. DTN forwarding increasing reliability in congested environments

The DTN forwarding engine is employed either proactively by the application via the UMOBILE naming API, or reactively, upon reception of congestion-related messages. In particular, an application can use specific hashtags that notify the smart forwarding module that a DTN-related strategy should be selected for the forwarding of the former's data. This activation method is depicted in Fig. 12. It should be noted unnecessary modules have been removed from the figure, to improve intelligibility.

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

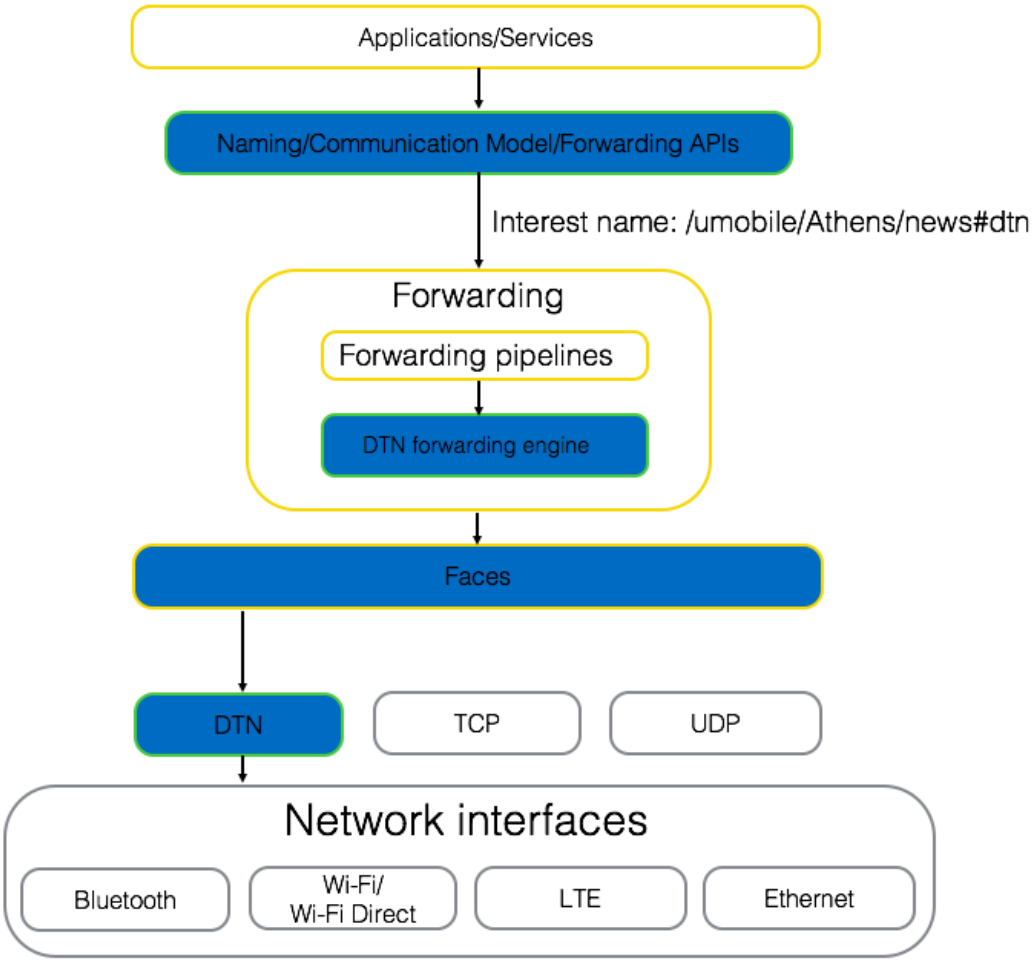

Figure 12. Activation of the DTN forwarding engine based on naming

The reactive DTN forwarding activation method is activated by NACK messages denoting congestion. We should note here that, again, the application shall enable DTN forwarding via the naming. The difference with the previous case is that now DTN forwarding is not enabled by default, by only upon reception of NACK messages. This activation method is provided in Fig. 13.

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

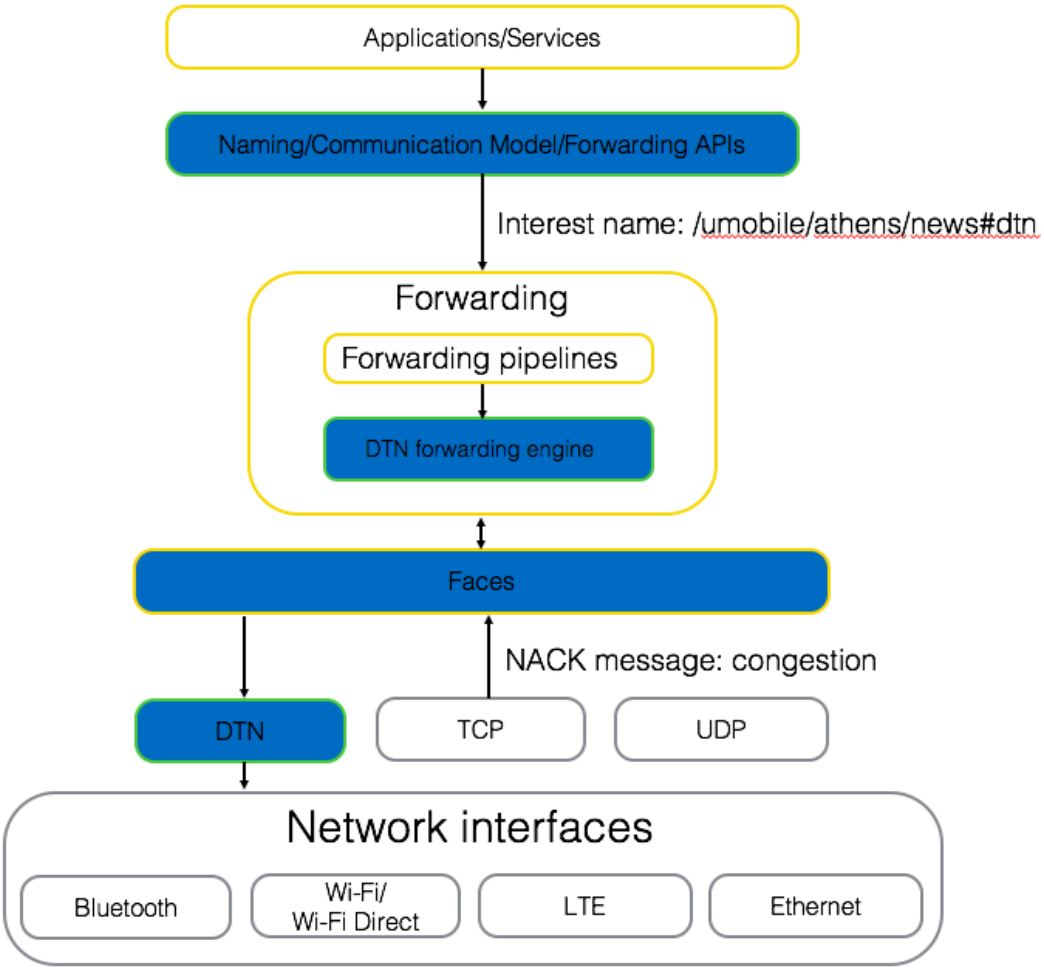

Figure 13. DTN forwarding activation upon reception of congestion-related NACK message

In terms of system requirements, the DTN integration into the UMOBILE platform covers the following requirements:

- R2: UMOBILE systems MUST be able to exchange data by exploiting every communication opportunity through WiFi (structured, direct), 3G and Bluetooth, among UMOBILE systems, operating even in situations with intermittent Internet connectivity)
- R13: UMOBILE systems MUST be able to provide the services to the end users when there is no Internet connectivity)

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

4.1.2. NREP: Name-based Replication Priorities

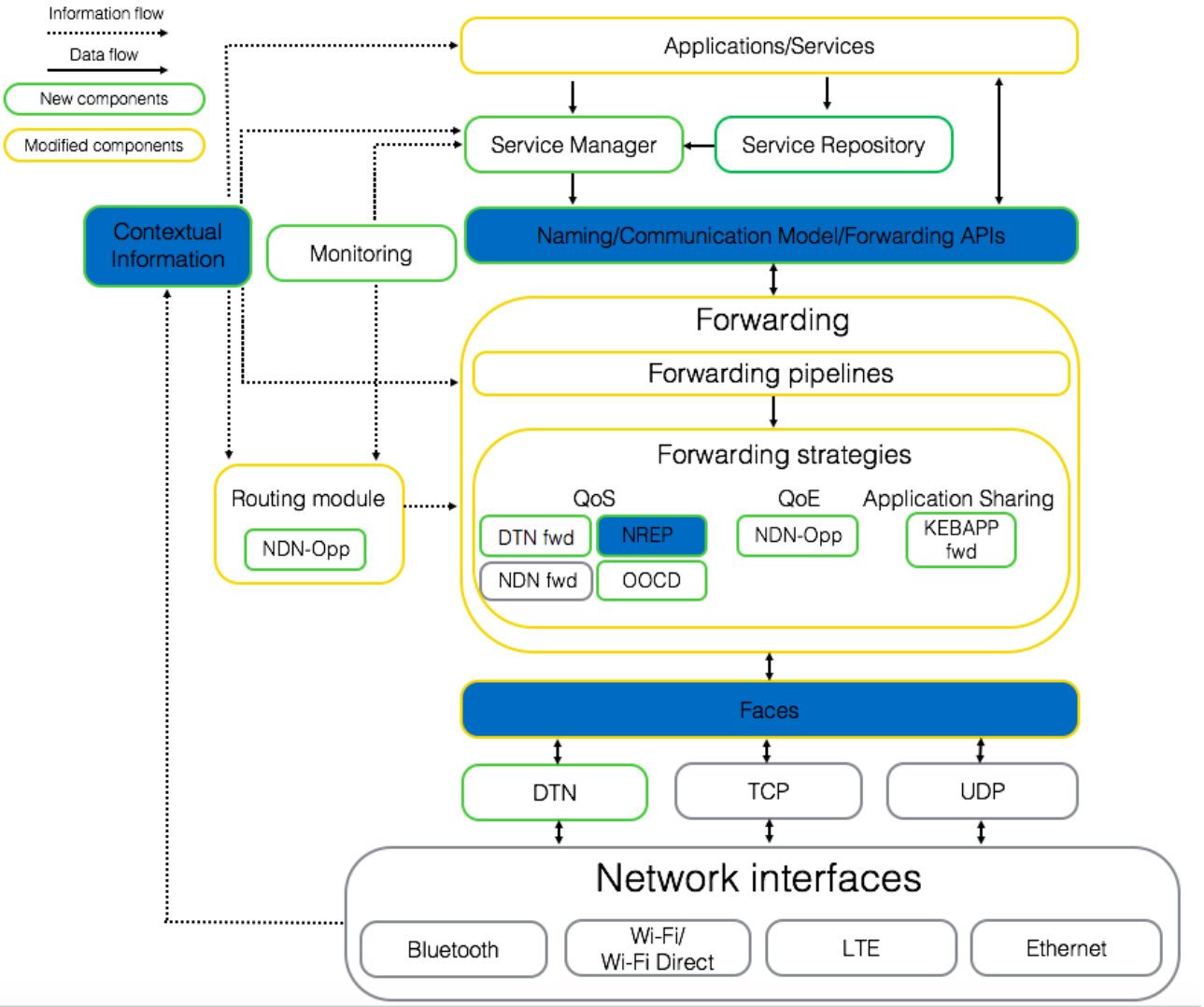

Figure 14. NREP modules

NREP has first been devised aimed at providing the best solution for emergency services transmissions using ICN opportunistic communications. These services will be supported by using replication, optimised by prioritisation rules, integrated within the information message's name to favour spreading of the most important messages. For example, in case of an emergency in a disaster area, we consider messages from first responders as more important than messages between friends. We focus on cases where the mobile network infrastructure is not available and therefore messages have to be stored, carried and forwarded by mobile devices.

In this task, we attempt to leverage the benefits of ICN in the aftermaths of a disaster, where infrastructureless delay tolerant communication becomes essential in order to deal with fragmented networks and the increase in traffic demand. Previous work done by UCL about NREP [4] use content names as the primary means for routing. However, unlike conventional ICN that is primarily designed to support name-based routing in an infrastructure-based

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

environment, NREP is designed to operate in an infrastructureless environment and focuses on name-based replication, rather than routing.

We argue for the need of a name-based forwarding/replication scheme, wherein intermediate nodes use a name associated with each message to make decisions such as whether to replicate and if so, according to what priority, or otherwise, store(-and-carry) and for how long storage should be allocated. Moreover, we discuss the need to expose other parameters such as priority, time-to-live and geographical constraints in the name or as attributes of the name.

This is done in order to help increase the efficiency of intermediate nodes to make decisions on storage and replication. However, since NREP is aimed to be developed over the UMOBILE platform and using WiFi Direct communications for Android devices, when we need to broadcast a message within a certain area, it is necessary to first establish a WiFi Direct group and broadcast the emergency message in this group, being this group formed by 2 peers or more. This connection period takes some time and adds an important latency to the communications, being up to 10 seconds in certain situations [5]. This latency cannot be omitted in mobile scenarios where contacts are short in duration time, since during that time the number of contacts around can change, losing contact with some of the users that were around. We need to prioritise those contacts with users that will have more affinity to the recipient of the message than others users.

To this end, we believe that we should take into account not only priorities between different content, but also we should have to prioritise WiFi Direct group establishments between those peers that are more suitable for their mobility history. This task considers NREP as basis as [4] to include social encounters in the connectivity manager aimed at prioritise connections with the best users to send emergency information.

Then, the task shall integrate other measures of affinity and develop NREP extensions for priorities, derived from contextual data. Examples of such measure can be aspects such as surrounding density (affinity network); priorities for the content (urgent, not so urgent); contextual input from the device (e.g., low battery). Such measures are derived from the sociability forecasting module (task 4.2, PerSense Mobile Light v2.0). PerSense Mobile Light v2.0 provides contextual data concerning roaming habits by using WiFi, data concerning social interaction, by using the WiFi and Bluetooth interfaces, and about users' behaviour, e.g., by using data concerning device usage. Such "smart" captured data (in contrast to raw data) is then made available to several UMOBILE modules, and in the case of the forwarding strategies, to the connectivity manager module.

4.1.2.1. Priorities and Namespace

In the following table, we specify an example of different priorities with its related attributes that can be defined in the NREP system and used to give priority to emergency services over other services. We think we can use this set of name prefixes as an example of different applications that can be used in the UMOBILE platform. As mentioned, these will be extended in the context of 4.3.

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

Table 1: NREP priorities example

Name-prefix	Priority	Time-to-live	Space	Authorizaiton	Recipient	Notes
SOS	High	Short	Closeby	All	First-responders	To use to ask for help
Government	High	Indefinite	All	Officials	All	To inform all of food-shelter, danger
First-responders	High	Indefinite	Depends	First-responders	All	To inform all of rescue-teams arrival
Warning	Medium	Indefinite	All	All	First-responders	First-responders verify and publish to all
Police	High	Depends	Depends	Police	Police members	To chat among themselves
Safe	Medium	Short	All	All	Public / Family	To inform others they are safe
Chat	Low	Short	All	All	Public	To chat among each other

4.1.2.2. Modules to be modified

- Naming scheme: The naming scheme needs to represent the content prioritisation depending on how critical is the application/service
- Content store: The caching policy should implement the prioritisation policies in order to replace first the content that is not important in terms of critical level.
- Application/services: The application services should use the specific naming scheme in order to take benefit of the proposed scheme.
- Forwarding engine: The forwarding engine should be modified in order to prioritise the critical content exchange when contacts between users occur.
- Face manager: The face manager should be modified to take into account the contextual information to manage and prioritise connection establishment with those users socially closer to the recipient of the content or those users that will be able to better spread the content.

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

- Contextual information: We will need to design interfaces between the contextual information module (PerSense Mobile Light) and the NREP modules in order to be able to use the contextual information to improve the priority information forwarding.

The NREP forwarding mechanism will be specified and developed within the Task 4.3 "Name-Based Replication Priorities" of the WP4. Deliverable D4.3 that is going to be submitted in M24 will provide all the details of NREP.

4.1.2.3. Network Requirements

The NREP approach fulfils the following network requirements:

- UMOBILE systems MUST be able to ensure data reliability and availability (e.g. taking into account data usefulness - time to live; manage duplicated pieces of information) among a set of distributed surrogates.

NREP will forward data to opportunistic nodes improving the availability of the content.

- UMOBILE systems MUST prioritise data to be exchanged, for instance giving high priority to emergency and civil protection information.

NREP will prioritise emergency data over other services less critical.

- UMOBILE systems MUST be able to deliver information within geographic regions and time frames that are relevant to different types of data.

NREP will forward content taking into account some policies like time-to-live, geographic region, priority or recipient.

- UMOBILE systems MUST be able to provide the services to the end users when there is no Internet connectivity.

NREP will provide critical information to users taking advantage of opportunistic communications even when there is no Internet connectivity.

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

4.1.3. Opportunistic Off-path Content Discovery

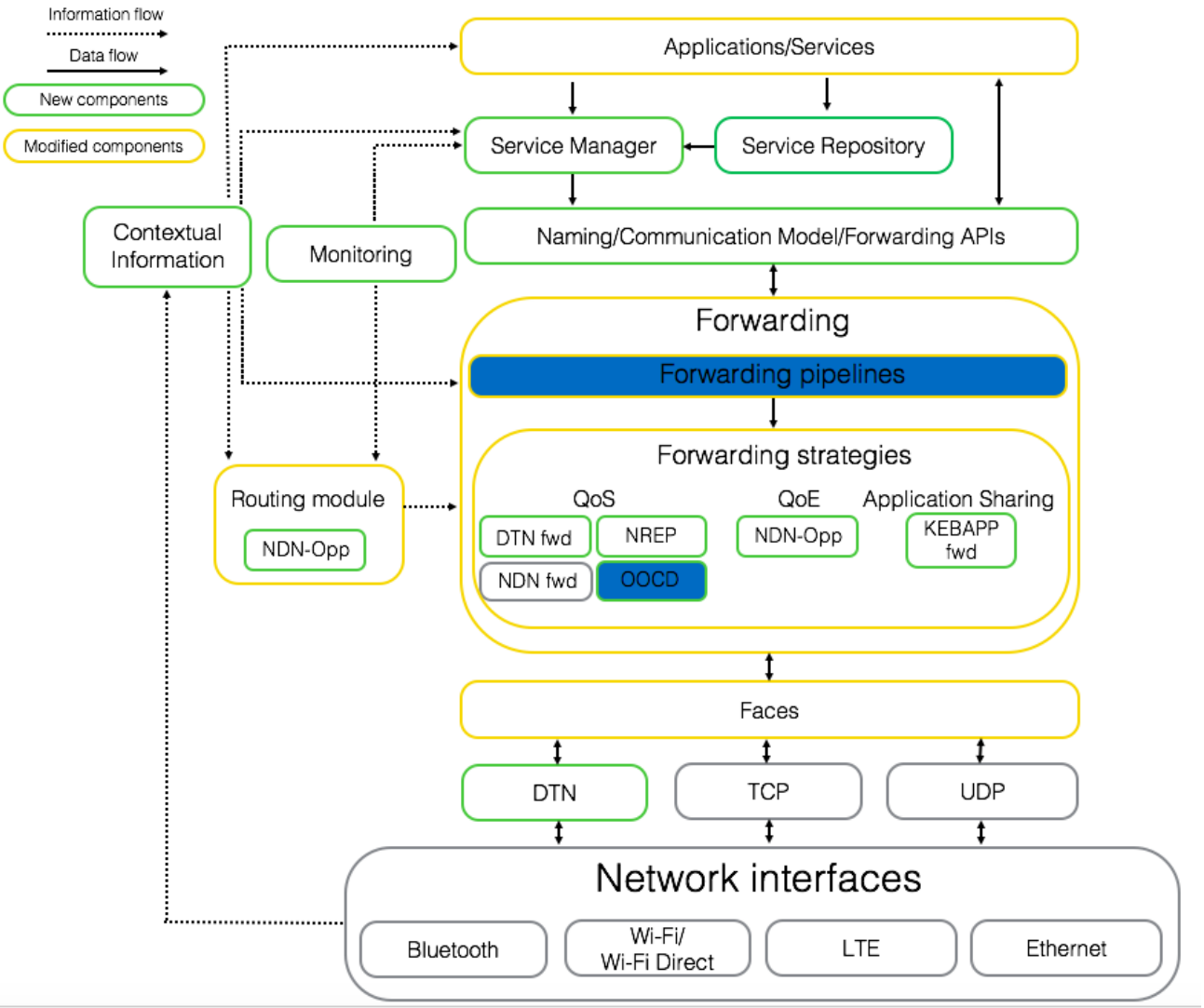

Figure 15. Opportunistic off-path content discovery modules

Data communications in fragmented networks, as can be UMOBILE networks, should not only relay on network-centric resilience schemes, as is traditionally the case, but should also take advantage of network management techniques that focus on information centric resilience. Such an information resilience scheme is useful in networks set after a disaster scenario, where the network infrastructure is only partially available (in either a temporal or spatial manner) and the reachability of the content origin is not guaranteed.

One of the main technical challenges according to the IETF ICNRG working group [6-7], regarding the usage of ICN in disaster scenarios, is to exploit network management techniques in order to enable the usage of functional parts of the infrastructure, even when these are disconnected from the rest of the network. As in [6], we also assume that parts of the network infrastructure are functional after a disaster has taken place and it is desirable to be able to continue using such components for communication as much as possible. Unfortunately, this is

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

especially difficult for today's mobile and fixed networks which are dependent on a centralised infrastructure, mandating connectivity to central entities for communication.

The desired functionality of an ICN set after a disruptive scenario is to allow the delivery of messages between relatives and friends, enable the spreading of crucial information to citizens and enable crucial content from legal authorities to reach all users in time. In such a deployment, where the content origin is not always reachable, a user can take advantage of similar interests issued by neighbouring users and their cached content in order to retrieve the requested data.

In this context, we build on the ICN paradigm and enhance the NDN architecture with extra components in order to make it resilient to network disruptions [8]. In this NDN extension, we introduce an extra Interest management routing table, which we call the "Satisfied Interest Table" (SIT) and which points to the direction of already satisfied Interests. This way, upon failure of links/nodes towards the content origin, the SIT table is redirecting Interests towards caches and end users that have recently received the requested content. We believe that network management for disruptive environments should take advantage of information-centricity, instead of focusing only on protocol-specific path recovery routing.

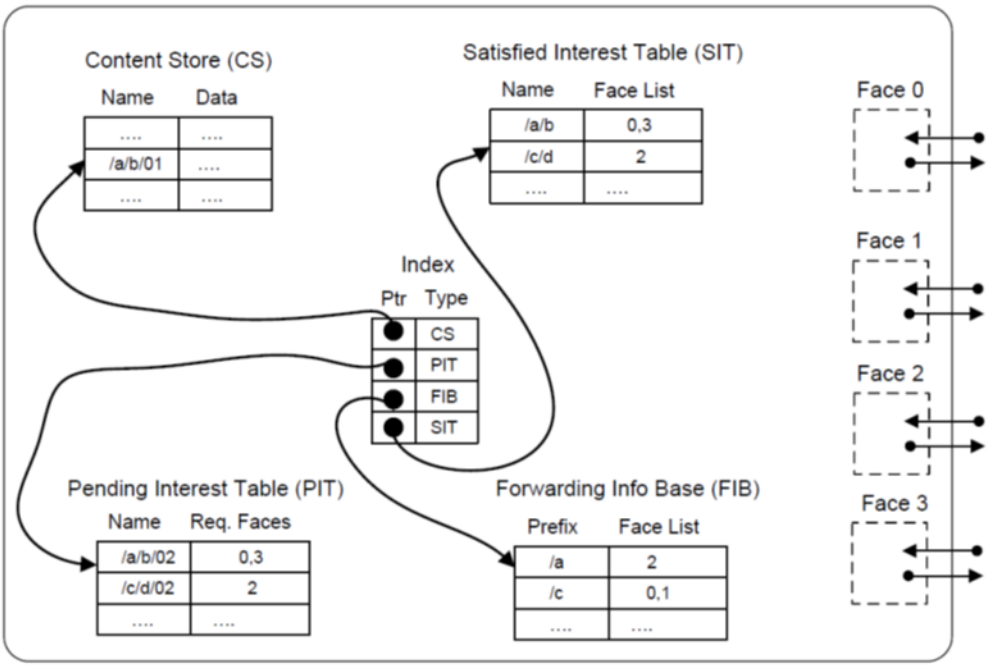

Figure 16. NDN design with the new Satisfied Interest Table (SIT)

In Figure 16, we can observe the main components of this NDN extension. As a new component of the NDN architecture, we have the aforementioned SIT table. SIT keeps track of the Data packets that are heading towards users. In the event that Interest packets cannot reach the content origin following the FIB entries, they can be forwarded based on the SIT entries towards users that issued similar interests in the past. SIT entries like PIT entries also allow for a list of outgoing faces, supporting multiple sources of data, which can be queried in parallel or sequentially depending on the chosen forwarding mechanism. Unlike a PIT entry, a SIT entry is triggered by a returning Data packet, and similarly to a PIT entry, SIT entries comprise a trail of

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

"bread crumbs" for a matching Interest packet that leads back to users with similar satisfied interests. Here, in order to improve scalability in disruptive networks, we assume that the newly introduced SIT entries are compiled in a file/object/content item basis instead of chunk/packet IDs to speed up the opportunistic retrieval mechanism. A node removes a SIT entry associated with a (directly attached) user in the event of a disconnection or when the entry expires (e.g., according to a time-to-live parameter).

We also introduce an Interest Destination flag (IDF) bit to the Interest packet in order to distinguish whether the packet is heading towards the content origin (IDF is set to zero), following a FIB entry, or is heading towards users with similar satisfied interests (IDF is set to one). In the second case (IDF is set to one), the Interest packet follows matching entries in the SIT of each passing-by router. We also introduce a Scoped-Flooding Counter (SFC) to further enhance the information retrieval capabilities of the proposed resilience scheme in some special cases that we describe in the next section. We assume here that the FIB entries of all the chunks of an item will be removed simultaneously from a router upon the "disappearance" of the content origin. This means that an Interest packet will have to visit only one network router (the one that the user is attached to) until the IDF is set to one, upon the fragmentation of the network.

4.1.3.1 Modules to be modified

- Routing module: The routing module should be modified in order to provide routing information not only for the shortest path, but also for the one-hop detour path to the content.
- Forwarding strategy: The forwarding strategy should be modified in order to forward content using the three modes of operation defined by INRPP.
- PIT: PIT tables should be modified in order to follow the "breadcrumbs" through the detour path as well.
- FIB: FIB tables should provide information for the detour paths as well.

4.1.3.2 Network requirements

- *UMOBILE systems MUST be able to provide the services to the end users when there is no Internet connectivity.*

Opportunistic Off-path Content Discovery is aimed at enabling the usage of functional parts of the network, even when these are disconnected from the rest. This means that the local data in the fragmented network is still available, even when there is no Internet connectivity.

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

4.2. Northbound APIs

4.2.1. Contextualisation (PerSense Mobile Light)

The introduction of new, cooperative technologies and in particular of low-cost wireless access, allowed the regular citizen to profit from the Internet as a commodity. This implies that most of the mobile devices (cellular, wireless) that have taken part on the Internet up until now as plain end-user devices, today can also be seen as networking nodes, having a role on the network operation. The movement that these devices have therefore impacts the underlying connectivity model and as consequence, it also impacts the network operation. Hence, being able to characterize such movement and also to estimate potential individual as well as aggregate movement is a requirement from the perspective of networking science evolution.

This need goes beyond the integration of movement prediction and/or anticipation mechanisms in the network operation e.g., in routing or mobility management. In fact, roaming behavior is becoming more relevant, and today, due to an extensive effort derived from several initiatives as well as from extensive and wide traces collections, it is globally accepted that there is a relation between social behavior and the user's roaming behavior. It is the social behavior that assists in defining future user movement, both from an individual perspective, and from a group perspective.

Hence, being capable of estimating such behavior is therefore relevant to optimize the network operation, be it from a mobility management perspective - e.g. handover optimization -; from a resource management perspective - e.g. performing a more intelligent load-balancing based on potential future moves of specific devices -; from a routing perspective - e.g. making routing more stable by selecting a priori paths that have a chance to be more stable in the presence of node movement; from a service migration perspective.

In the context of UMOBILE, PerSense Mobile Light is a service that has been developed to assist in performing network contextualization. Currently, PerSense Mobile Light captures information concerning a user's affinity network (contacts derived from WiFi Direct and Bluetooth) as well as concerning roaming habits, over time and space (WiFi). In a future version, PerSense Mobile Light shall collect data on user behavior, derived from additional sensors - the sociability forecasting module.

The current development phase (Phase I) was released in May as PerSense Mobile Light v1.0. This service allows data capture that assists in understanding preferences in terms of roaming habits and visited wireless networks. This type of contextualization is relevant, assuming, for instance, that a UMOBILE mobile device may present a roaming preferences model, where the intent is to consider personal preferences in terms of visited wireless networks. Or, it may contain a model that relates with the need to share data opportunistically based on frequency or volume of wireless contacts. These models shall be part of Phase II of PerSense Mobile Light (v2.0, December 2016), and shall assist a user definition, based on specific parameters. The identified user context is then used, together with the collected sensing data, to infer patterns of user behavior. This support is coined as “sociability forecasting” in the context of UMOBILE

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

(refer to D3.3) and shall integrate aspects such as user recommendations and estimation of conditions for social interaction to occur derived, e.g., from shared interests; affinities; wishes.

Tutorial

PerSense Mobile Light v1.0 can currently be used in Android devices. For that purpose, the researcher needs to join the Google Beta Community Pervasive Sensing Experimental Tools Community [3]. After this step, a URL will be available to download the tool. This first service version has as main purpose to assist the academic community in gathering meaningful traces and develop scientific studies, by reusing traces. The service captures wireless footprinting aspects such as geo-location; Access Points crossed and used; visit type and duration. It also captures affinity networks (devices around, via WiFi Direct). This data is dumped daily to a server database (SQLite) provided by partner Senception. The traces are expected to be open via the UMOBILE Lab as well as via Crawdad, after adequate treatment (e.g. MAC obfuscation). The data is also stored on the device per day; per week, and for one month. During night, the data is dumped to the UMOBILE server. The data can be easily exported to usual formats such as cvs, or xls.

The database comprises several tables:

- Roaming tables:
    - 16 tables, 1 for each day of the week; 1 for each week 1 to 5; 1 for the current month.
    - Table entry tupple for each AP: `<Id, bssid, dayoftheweek, state, ssid, attractiveness1, lastgatewayIP1, dateTime, lat, long>`
- Visits table
    - Connected APs, entry tuple: `<Id, ssid, bssid, timeon, timeout, dayoftheweek, hour>`
- Affinity Network tables
    - 16 tables, 1 for each day of the week; 1 for each week 1 to 5; 1 for the month
    - Table entry tupple for each device in the vicinity: `<Id, bssid, dayoftheweek, state, ssid, attractiveness*, lastgatewayIP*, dateTime, lat, long>`

1 - fields to be worked in version 2.0 only.

[3] https://plus.google.com/communities/104874036636715946374

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

The data is stored per device (1 directory per device, and per day). The SQLite database is stored as e.g. 19-04-2016-persenselight.db. This database can then be accessed with a command to edit SQL databases, such as the sqlitebrowser CLI, and the different tables can then be easily exported to a format such as cls or csv.

Architecturally, and as depicted in Fig. 17, PerSense Mobile Light gathers data from Network Interfaces (such as WiFi Direct) and interfaces with Applications providing them with contextual information. It also feeds the Routing module with relevant data.

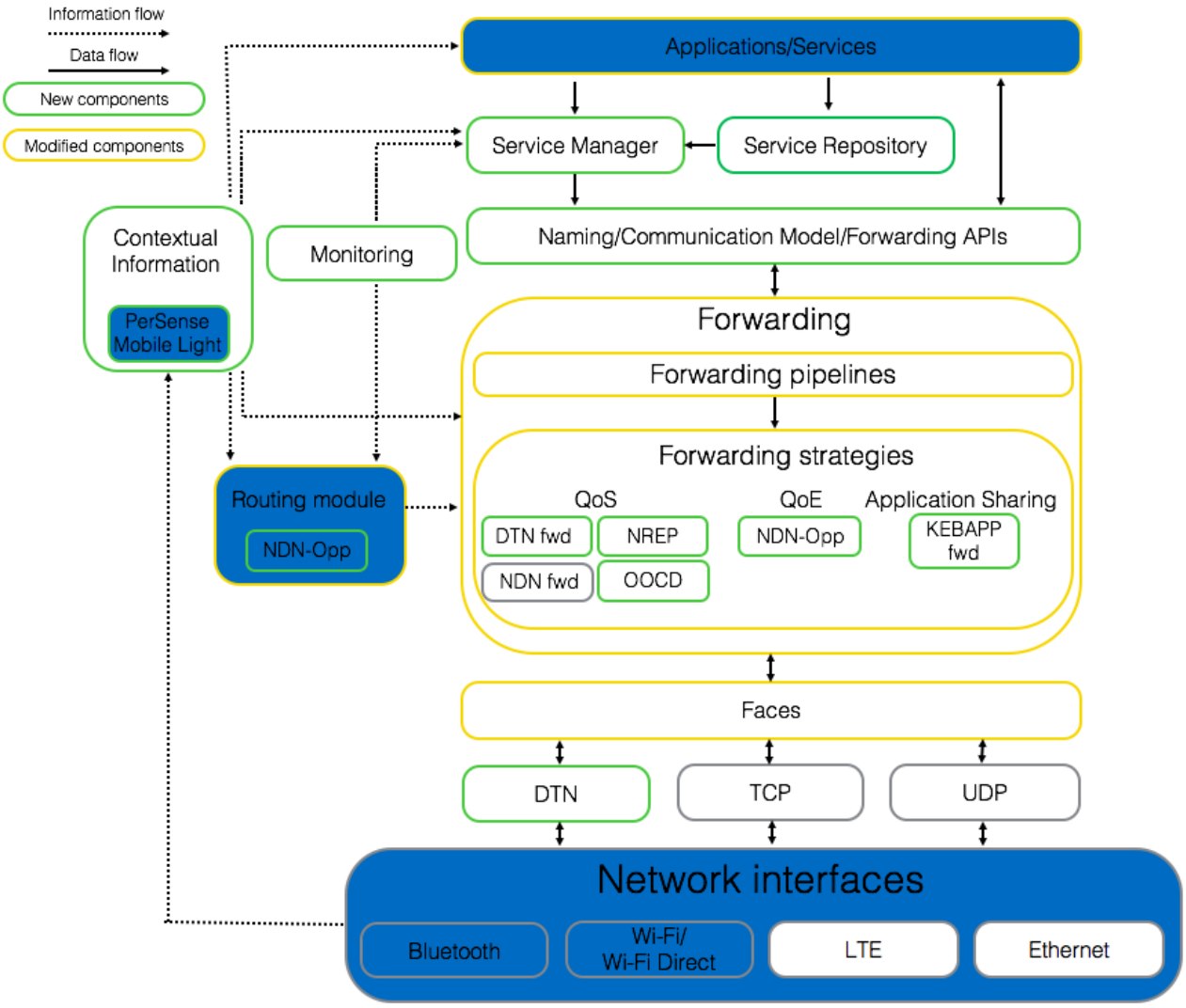

Figure 17. The PerSense Mobile Light as part of the UMOBILE platform

The low-level design of the Contextual Information Module is provided in Fig. 18.

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

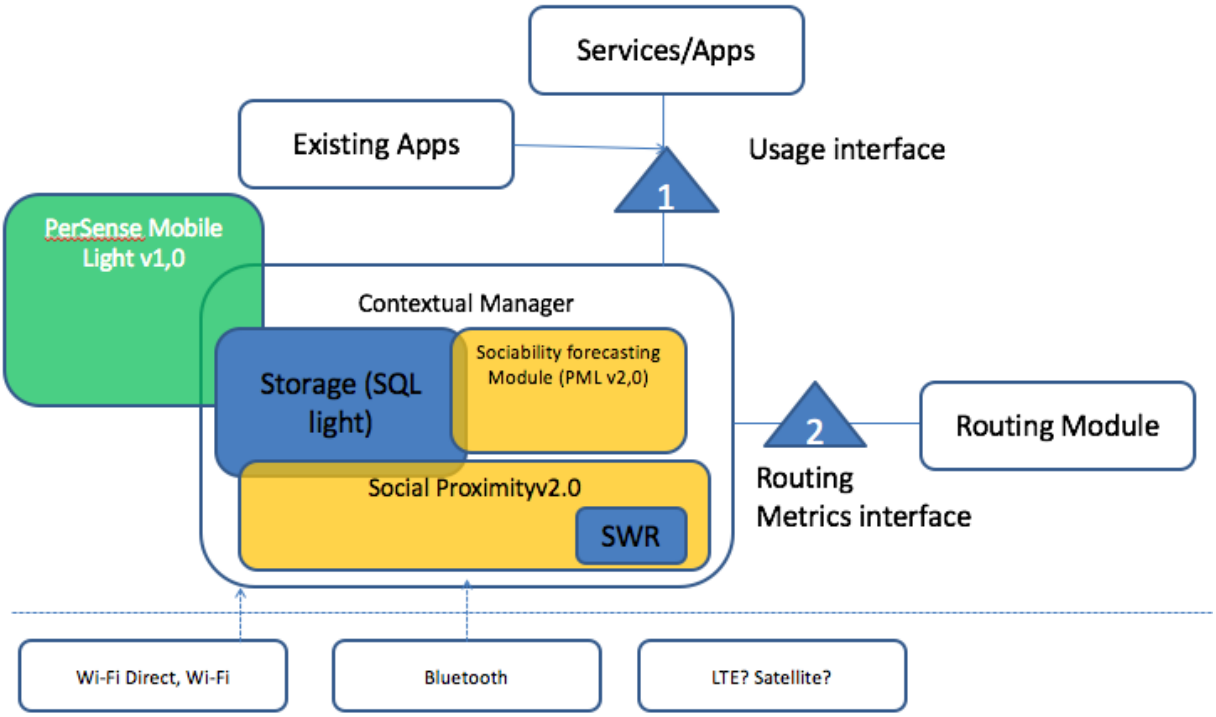

Figure 18. Low-level design of the Contextual Information Module

PerSense Light v1.0 contributes to the following UMOBILE requirements

- R1: assists in understanding social trust circles; how they organize (contextual aspects) and their duration
- R4: keeps all data local; relies on network mining and not on personal raw data capture
- R10: helps in understanding the relation between location and social daily routines (roaming patterns and geographical regions)
- R18: performs seamless sensing of user context
- R20: provides feedback about networking dynamics, based on realistic social routines (roaming patterns)
- R23: allows authorized people to track the routine of registered devices
- R24: does not need an always on Internet access
- R26: allows users to manage their trust circles

Sociability forecasting module in PerSense Light contributes to the following UMOBILE requirements

- R15, R16: assists in inferring user interests by analysing local usage

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

- R18: performs seamless sensing of user context
- R24: does not need an always on Internet access
- R27: assists in matching familiar strangers' interests and assists in facilitating meetings

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

4.2.2. PUSH services

Primitive NDN supports only the pull-based communication model. However, as documented in D3.3, the UMOBILE platform needs to support both push-based and pull-based communication models to be able to serve as an implementation platform for the proposed user scenarios and their specific requirements. To cover the gap, Task 3.1 has implemented the pull-based communication model and integrated it to the UMOBILE platform. We discuss the implementation in the following sections.

In the pull-based model, a consumer (or receiver) initiates the communication. The consumer requests a piece of data that might be available from more than one producer (i.e., any node who has the matched content in the cache, original content source). In contrast, in the push-based model, a producer initiates the communication. The producer pushes a piece of data that might be delivered to one or more consumer who are interested in or subscribed to retrieve the data. A comparison between pull-based and push-based communication is illustrated in Figure 19.

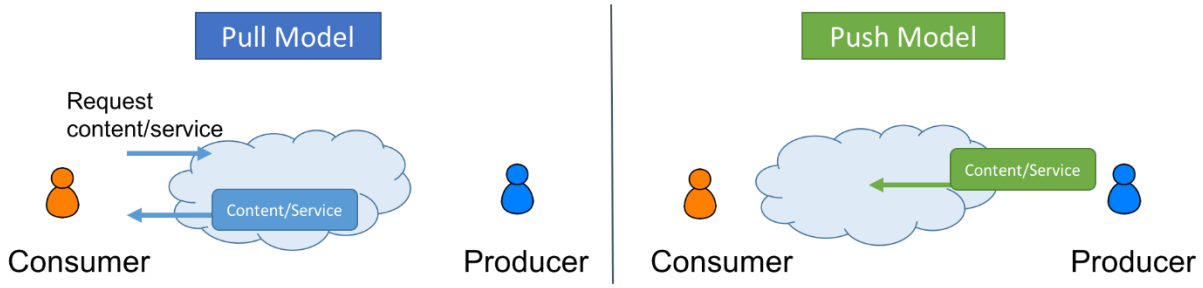

Figure 19: Push-based vs Pull-based communication models

Since the primitive NDN does not support push-based communication, a producer cannot directly send a Data message to the consumer. The reason is that NDN relies on reverse path forwarding where the forwarding Data message follows information in the PIT. However, PIT information is registered only when an Interest message is flowed to the NDN node. In other word, either the consumer or producer must send an Interest message along the path to add a new PIT entry to every intermediate node before sending the Data message. As a result, a Data message can be sent following the chain of the intermediate nodes.

In the implementation of the push-based communication model that we have done for the UMOBILE platform, push-based communication is integrated to the platform and appears as a central component of the architecture. The push-based communication services are meant to help meet the following UMOBILE network requirements as documented in the D2.2.

- R-6: UMOBILE systems MUST have an application programming interface allowing users to publish new data and subscribe/register their interests in data.
- R-8: UMOBILE systems MUST allow geo-location of users to be automatically added to contents generated by emergency applications.

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

- R-9: UMOBILE systems MUST be able to make specific types of messages available to different receivers.

- R-13: UMOBILE systems MUST provide users only with relevant information, i.e., matching user interest.

- R-17: UMOBILE systems SHOULD be energy efficient, aiming to increase their availability and reduce their operational cost.

Figure 20 illustrates our implementation of the push-based services. The functional blocks required to implement the push-based services are shown. The assumption is that the owners of the services/applications provide the communication descriptors to the Service Manager. Consequently, the Service Manager chooses the appropriate communication model (e.g., push-based or pull-based) and passes the specific description to the communication model (R-9).

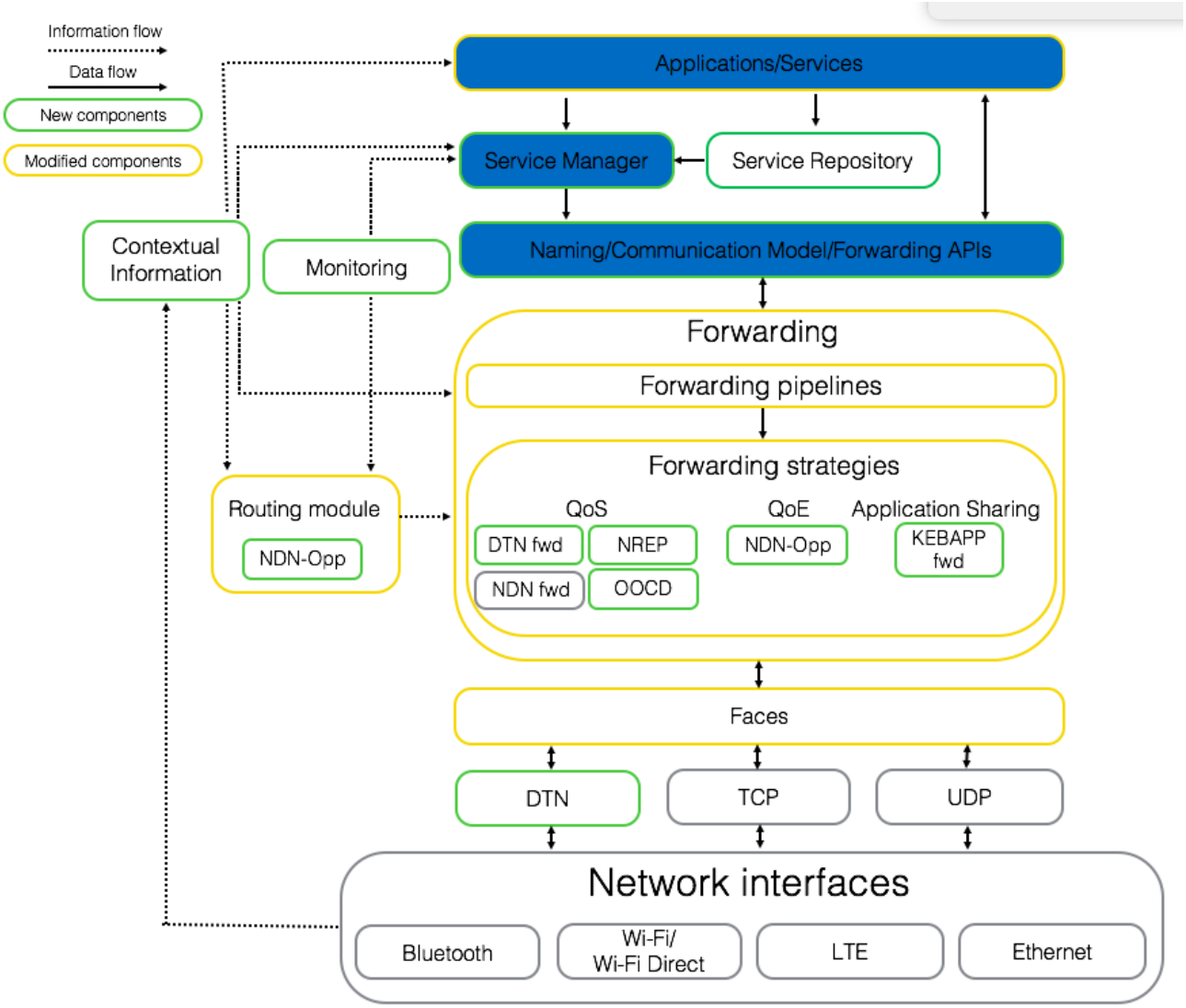

Figure 20. The function blocks for push services

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

Pull-based communication model: The pull-based model is inherently supported by NDN. With this model, clients can request the desired content by issuing Interests with a name prefix. The content providers directly or any other NDN node that has the requested content in its cache can respond by means of forwarding the requested content towards the Interface following information in the PIT. This communication model satisfies the requirement R-13. Figure 21 illustrates the pull-based communication model in the NDN architecture using Interest/Data exchange.

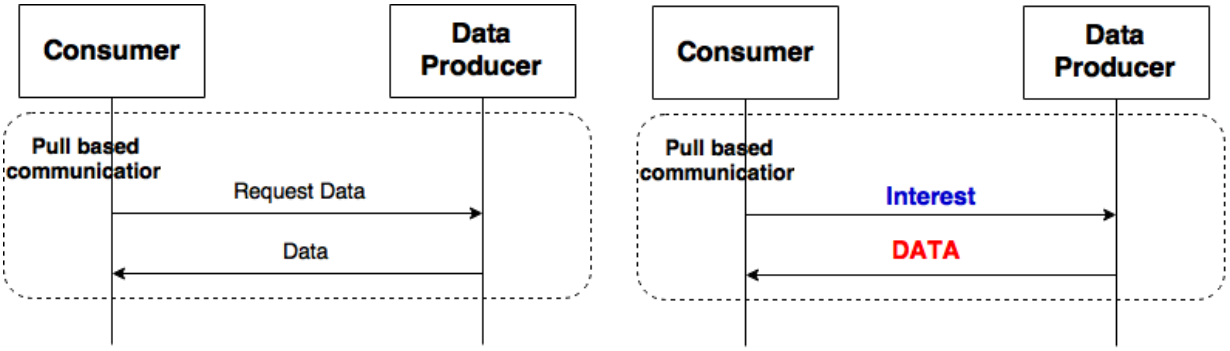

Figure 21. Pull-based communication model

The primitive NDN architecture does not support the push-based communication model. We have implemented it on the basis of its default Interest/Data exchange. Note that we have implemented several functions to enable the UMOBILE platform to support several variations of the push-based communication model: push-Interest poling, push-Interest and push-publish data dissemination.

Push-Interest polling: When the consumer does not know when the data of interest will be generated and available, it can keep polling the producer by means of issuing periodic Interests. When the requested data is available, the data producer can push it in a Data packet. This model can be used in energy efficient mode where a producer can turn off for a short period of time to save energy (support R-17). Once the producer in back to its ON state, it can receive the periodic Interest message and reply back immediately with the matching Data message. Two potential drawbacks of polling are that it can cause network traffic and that the freshness of the received data depends on the polling frequency. However, in some scenarios such as emergency situations where the consumer cannot be associated with any UMOBILE hot spot, this model can be useful to retrieve emergency services or contents. The Interest/Data message exchange of push-Interest polling is illustrated in Figure 22.

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

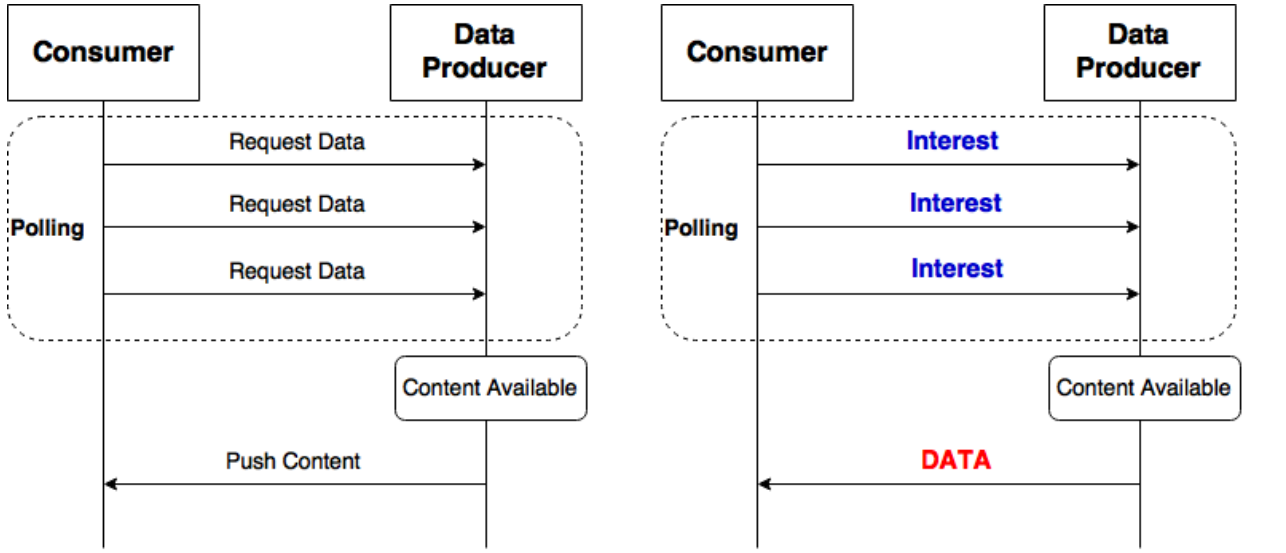

Figure 22. Push-Interest polling communication model

Push-Interest notification: In this model, the content of interest is generated as a text format i.e. as an instance message or a text notification sent by a server. This data can be directly appended to the Interest name. This method is commonly referred as Interest notification. Upon the reception of such an Interest, the receiver can optionally send an Ack Data to confirm Interest reception. This model can support R-8 where the geo-location of the user can be attached to the Interest message. The push-based communication based on the Interest notification model is illustrated in Figure 23.

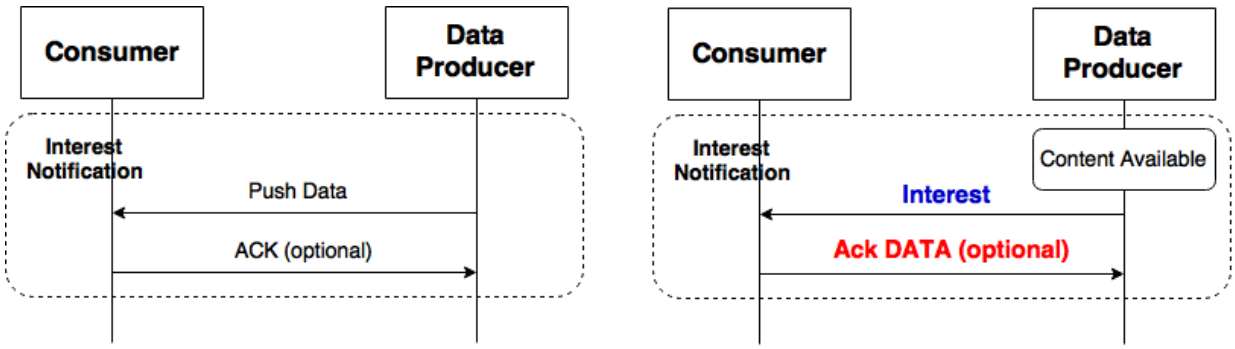

Figure 23. Push-Interest notification communication model

Push-Publish data dissemination: In this model, we follow the publish-subscribe model (R-6) where data producers can publish their contents or services via Interest message to the subscribed consumer which in turn trigger an Interest back from the consumer to fetch data. Figure 24. illustrates the Interest/Data exchange of push-based communication with publish data dissemination under the NDN architecture. The data producer initially sends the publish message to its subscribed consumer. To distinguish the Interest message from other models, a name component, "publish" is added after the content name. Consequently, the consumer receives the publish Interest message, it discards the last component ("publish") and sends the new Interest message with the content name to request the data. In NDN, content is divided into several chunks. Due to self-traffic regulation feature of NDN, N-1 subsequent requests are sent

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

to retrieve all chunks of the requested content. The last component in the name structure is reserved for the incremental chunk ID which is automatically generated regarding previous received data chunk.

In general, in the NDN architecture, the Data message is expected to be sent after receiving an Interest message. However, in the push-publish data dissemination model, the consumer responds to the first published Interest with a new Interest message. In this context, we use the NDN API to control the arrival of Interest messages by filtering names with key word "publish". If the Interest's name contains this keyword, the communication is processed through the publish data dissemination scheme.

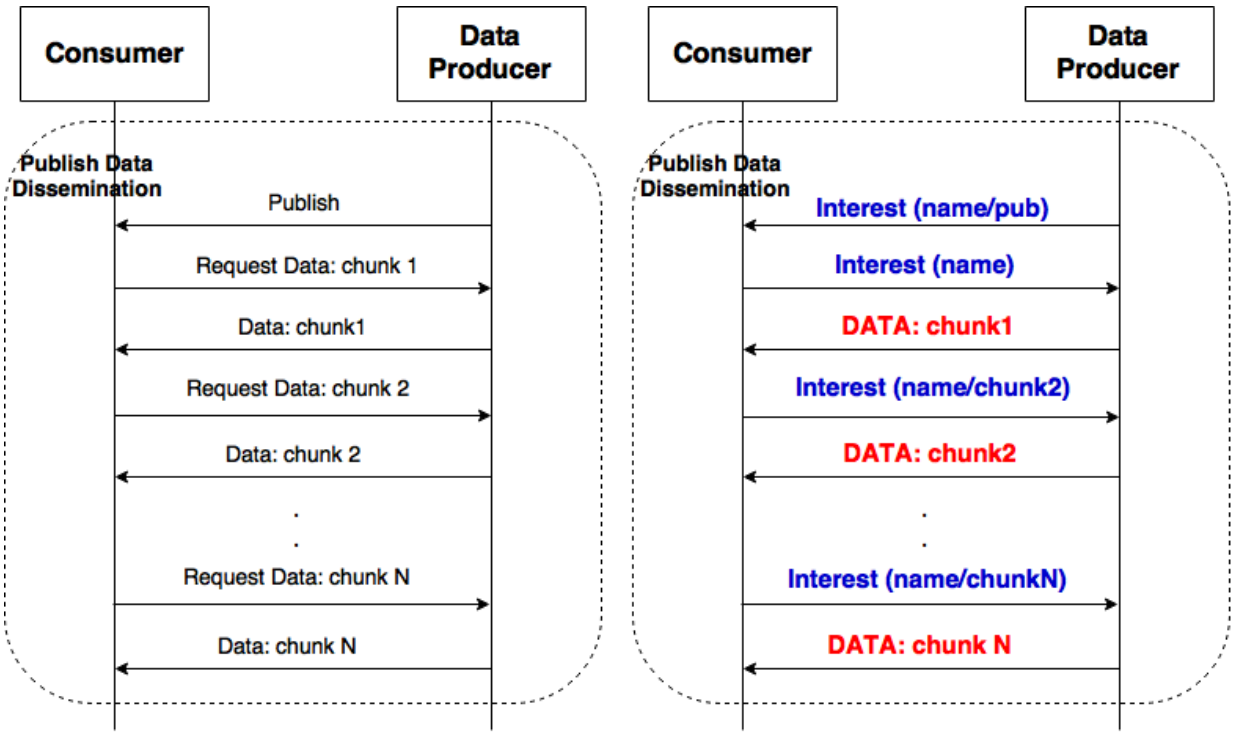

Figure 24. Push-Publish data dissemination communication model

Push services are applied in several modules of UMOBILE architecture.

- The monitoring module requires the push based model to fetch the monitoring data from the UMOBILE hotspot.
- The UMOBILE hotspot uses the Push - Interest notification model to report the node usage (e.g., memory usage, CPU load) to the monitoring module.
- The decision engine in service migration platform also requires the push - Publish data dissemination to migrate the services to the selected UMOBILE hotspots.

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

4.2.3. KEBAPP application sharing platform

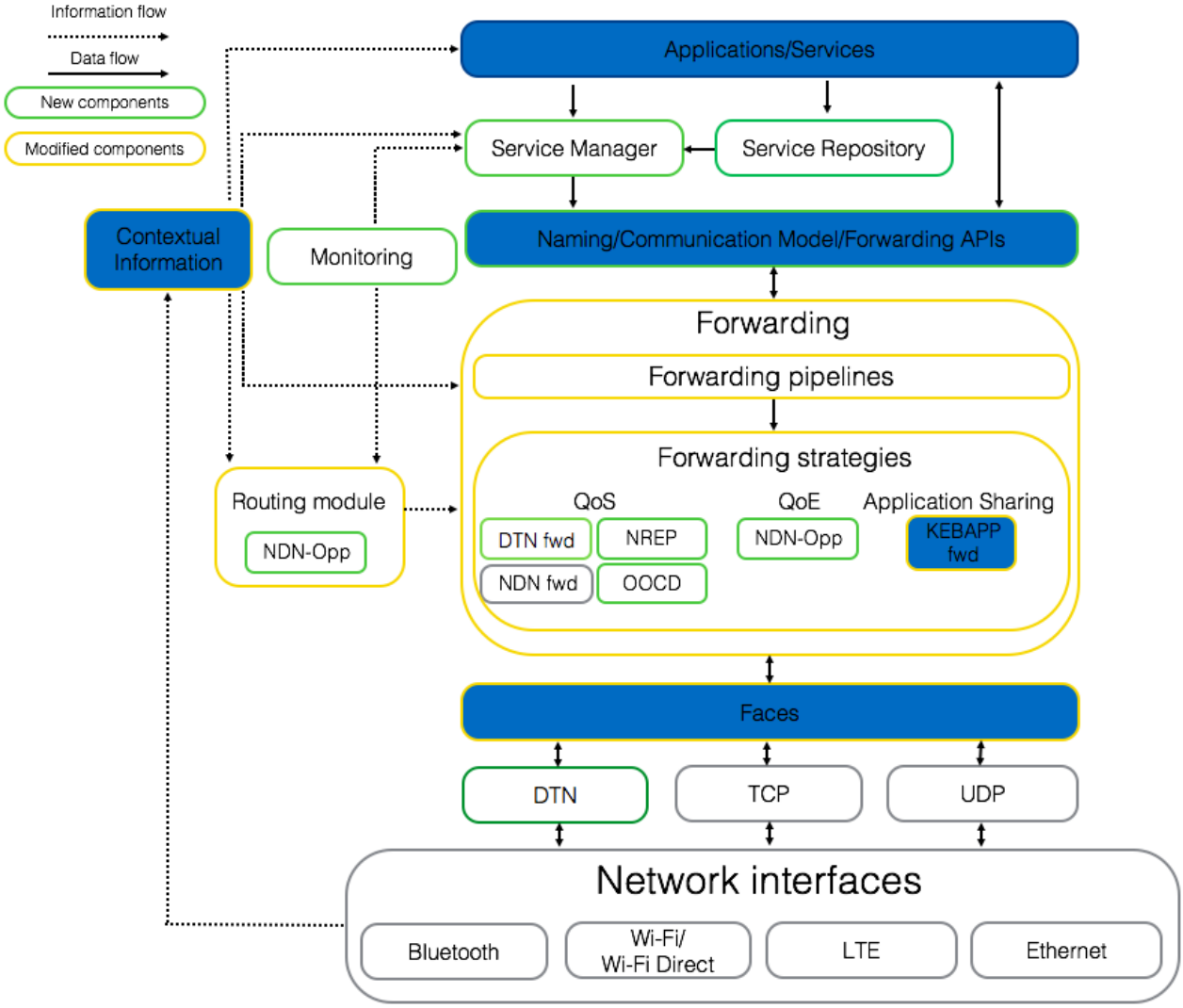

Figure 25. KEBAPP modules

Within the UMOBILE project, we present KEBAPP [9], a new application-centric information sharing framework oriented to support and provide opportunistic computing to mobile devices (smartphones, tablets, etc.). Our approach targets scenarios where large numbers of mobile devices are co-located presenting the opportunity for localised collective information exchange, decoupled from Internet-access.

In KEBAPP, we propose the creation and use of 802.11 broadcast domains for the support of particular applications i.e., KEBAPP-enabled hosts or APs advertise one or more Basic Service Set(s) (BSSs) for the support of one or more application(s). The creation of application specific BSSs aims at enabling mobile devices to connect only when their counterparts support the same application and/or namespace. The advertising AP or host, through a WiFi Direct Group, acts as a mediator to connect different users willing to share the same application in a single broadcast

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

domain. In the case of APs, functionalities such as access control, association, encryption, etc., can be supported without imposing computation and/or battery overheads to mobile devices.

In this context, KEBAPP employs application-centrism to facilitate/enable (i) the exchange of processed information, in contrast to merely static content, and (ii) the discovery and delivery of information partially matching user interests. Figure 25 presents the structure of a KEBAPP-enabled host. KEBAPP provides a new layer between the application and the link layers exhibiting three major design features. Namely, (i) application-centric naming, where applications share common name spaces and further support the use of a keywords, (ii) application-centric connectivity management, where applications manage connectivity by defining and/or joining WiFi broadcast domains, and (iii) information-centric forwarding, extending NDN architecture.

4.2.3.1. Operation

The basic forwarding operation of a KEBAPP node is a modified version of NDN architecture, aimed at reflecting the forwarding of messages within the various Basic Service Sets (BSSs) a node may participate. As explained in the following, since we consider single-hop broadcasting domains, forwarding decisions lead to either the broadcasting of a message in the BSS or its delivery to a local application instance. As such, broadcast domains are considered as (inter)faces of a KEBAPP node.

The KEBAPP forwarding scheme, similarly to NDN, has three main data structures: FIB (Forwarding Information Base), CS (Content Store) and PIT (Pending Interest Table). The FIB is used to forward Interest packets toward potential sources of matching data. The CS is responsible of caching the information requested by the users, providing fast fetching for popular information and avoiding to recompute information already requested by other users. The PIT keeps track of Interests forwarded to any BSS.

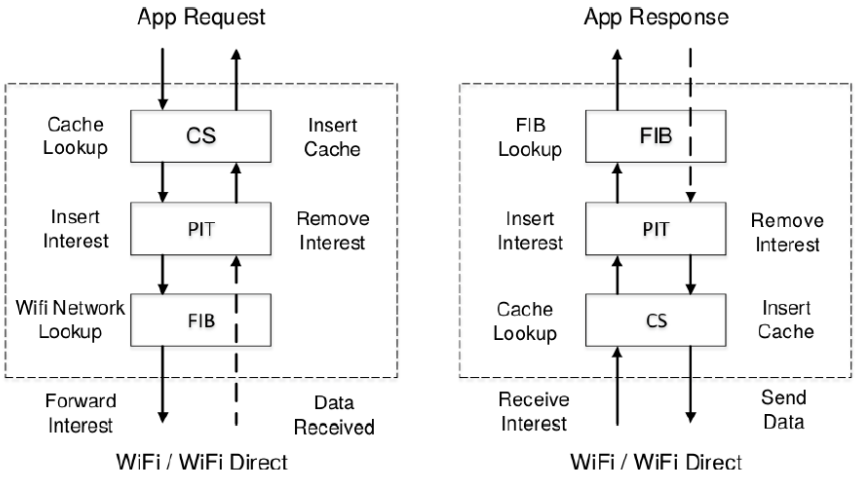

Figure 26. KEBAPP operations

Figure 26 above provides a representation of the KEBAPP packet forwarding engine. Below, we detail the operation of KEBAPP framework for an information requester:

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

1. The application requesting for information creates a new Interest.
2. The application looks for the information in the local CS. If the information exists locally, the data is sent to the application.
3. If the information is not found in the CS, the KEBAPP layer inserts an entry in the PIT table. As in NDN, we use the term "Internal Face" to point to the local application involved in the transaction (either as requester or as provider).
4. The KEBAPP (network) layer checks if there is a BSSID entry in the FIB matching the name prefix of the PIT.
5. If an entry for the requested name prefix exists, the connectivity manager connects the WiFi interface to the BSSID in the FIB and broadcasts the Interest message with a corresponding time-out value.
6. Each time a new FIB entry is added because a new prefix name is discovered on a new BSS (e.g., through WiFi NAN), the KEBAPP layer checks if a pending PIT entry for this prefix exists. As mentioned above, this corresponds to PIT entries created for Interest messages that could not be forwarded. In case an entry exists, the Interest is sent through the recently added BSSID, and the entry is updated with the BSSID value.
7. When a response is received with the information requested, the KEBAPP layer looks for the internal face that points to the application in the corresponding PIT entry and forwards the response to it. The PIT entry is removed and the information requested is cached in the CS.

Next, we describe the operation of the KEBAPP framework for an information provider:

1. The user receives an Interest through the interface connected to a certain BSS related to an application.
2. The KEBAPP layer checks the CS for a matching entry.
3. In case there is no entry in the CS matching the Interest, a PIT entry is first created. This entry allows the provider device to serve multiple, concurrently arriving, identical requests with a single message i.e., applying multicast, as in original NDN. In this case, the Requesting Face list of the entry includes the BSSID of the current BSS. Subsequently, the FIB table is looked up and the "Internal Face" is used to forward the Interest message to the corresponding application. For completeness, the "Internal Face" is also added to the PIT entry as an output face.
4. The response from the application is cached in the CS and sent back to the broadcast domain indicated by the BSSID value of the local PIT entry, which is subsequently removed.

4.2.3.2. Manual

Tables

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

NDN Forwarding *Daemon* (NFD) is a network forwarder that implements NDN protocol. We plan to extend NFD reusing most of its components and doing some modifications in order to provide a KEBAPP framework implementation. The tables module provides main data structures for NFD. In the following we describe the main tables used in NDN and the modifications required by KEBAPP:

- CS: The CS is the cache of the data packets arriving. In this case, we do not need to modify the CS NFD implementation. However, the name used to index the data, will include the name hierarchical prefix name, but also the different hashtags used to define the data included in the name.
- FIB: The FIB is used to forward Interest packets toward potential source(s) of matching Data. It is almost identical to an IP FIB except it allows for a list of outgoing faces rather than a single one. Here, the FIB structure is modified compared with the NDN FIB. The KEBAPP FIB requires to faces: the BSSID of the WiFi network where the application is available, and an internal face with a pointer to the local application that is willing to be shared with other users.
- PIT: The PIT keeps track of Interests forwarded looking for application sharing. In this case, we need to extend the PIT in the same way we extend the FIB, and we need to include a second face, and input face that keeps track of the origin of the Interest (the local application or the WiFI network) and the output face that keeps track of the Interest destination.
- Other tables: Other tables, such as the Network Region Table, the Dead Nonce List or the Interest Table for loop detection do not need to be modified since KEBAPP does not need those tables for its operation. The Measurements Table can be used to store measurement information regarding certain apps.

Managers

Connectivity management plays a vital role in KEBAPP. KEBAPP focus on WiFi-enabled (IEEE 802.11) connectivity, including WiFi Direct, which enables mobile devices to act as APs by forming communication groups. The creation of an application-specific BSS requires the ability of mobile devices to identify the mapping between the BSS and the corresponding application. The recently announced WiFi Neighbour Awareness Networking (NAN) protocol can support this requirement. It is noted that a device can be connected to more than one BSSs at the same time, thus acquiring or providing information across several applications. .

- Face Manager (Faces): The Face Manager is responsible of creating and destroying faces. The face manager uses the information provided by the WiFi/WiFi-Direct interfaces to create or removing faces to the multiple WiFi networks or certain local applications
- FIB Manager (FIB table): The FIB Manager is responsible of updating the FIB table. FIB manager uses the information provided by the face manager about the networks

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

available in the vicinity and adds new entries for the KEBAPP applications available in each network.

4.2.3.3. Modules to be modified

- Naming scheme: The naming scheme should use the proposed keyword-based naming scheme in order to improve scalability and provide flexibility to applications to retrieve useful information.
- Content store: The content should take into account this keyword-based naming scheme to index the data properly and forward this data when required.
- Application/services: Applications network interfaces should be modified in order to share the data with other users and retrieve data not only from central servers.
- Forwarding engine: The NDN forwarding engine should be modified in order to provide new strategies.
- Face manager: The Face manager should be used to create and destroy WifiDirect groups, discover other peers that are sharing the same application, and populate FIB tables with the information of the users in the vicinity.
- PIT: The PIT table should be modified in order to keep the information necessary not only to forward the data to the network but also to identify the local app at the same time.
- FIB: The FIB table should be modified in the same way the PIT table should be.

4.2.3.4. Network requirements

- *UMOBILE systems MUST be able to exchange data by exploiting every communication opportunity through WiFi (structured, direct), 3G and bluetooth, among UMOBILE systems, operating even in situations with intermittent Internet connectivity.*

  KEBAPP will provide application sharing capabilities in order to provide processed information to users that have no Internet connectivity exploring other connections such as WiFi Direct.

- *UMOBILE systems MUST be able to exchange data taking into account user data interests and context.*

  Since KEBAPP is application-centric, users that are not interested in the data because don't have the application or are not willing to share it, won't participate in the communications

- *UMOBILE systems MUST be able to provide the services to the end users when there is no Internet connectivity.*

  KEBAPP is aimed at providing services to those users that have no Internet connectivity.

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

- *UMOBILE gateways are able to convert the ICN traffic to traditional IP packet format and vice versa*

  KEBAPP can be used for gateways to provide to UMOBILE users, without Internet connectivity, information from the Internet to the UMOBILE NDN domain.

4.2.3.5 Code

A first version of the NFD, including the modifications required to enable the KEBAPP sharing framework, has been already developed. In the following, we add the link to the code repositories with the modified version of NFD and the Routefinder sample application presented during the Midterm review in October 2016 to the European Commission.

The KEBAPP NFD *daemon* code is provided in the following link:

https://github.com/srene/KEBAPP

The Routefinder sample KEBAPP application can be found in the following link:

https://github.com/srene/routefinder

4.2.4. Short Messaging Application

The Short messaging application Oi! was developed to operate above the SOCIO framework (described in section 4.4.2). The Oi! application uses the SOCIO User Interface (UI) allowing the user to compose a message, choose its recipient from a list of known contacts, and to see received messages. Though the UI interacts Oi! interacts with SOCIO's Content Manager (CM) module, which is responsible to manage all the messages to be sent, as well as received messages. The primitives `Send(destination, message)`, allows the application to pass to the CM the data needed for this one to build a message to be sent to a given destination node; *Receive(message),* allows the application to get from SOCIO's CM a received message; and `GetContactsList()`, allows to fetches a list of known contacts to display on the UI.

Fig 35 shows how Oi! works on top of the SOCIO framework, to allow users to opportunistically exchange messages based on their level of social engagement. So, the first check (1) is whether the SOCIO framework is available. If not, the application asks the user to proceed with SOCIO's installation (2). Otherwise, Oi! connects to the framework (3).

Once connected to the SOCIO framework, Oi! requests the contact list (4) and shifts to idles state (5), in which one of the following actions may take place (order is irrelevant and serves for the purpose of explanation only):

Contact list is received: Oi! then updates the list of contacts on the user interface (6).

Messages are received: Oi! displays the incoming messages to user (7).

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

User composes a message: Oi! forwards the newly composed message to SOCIO framework (8).

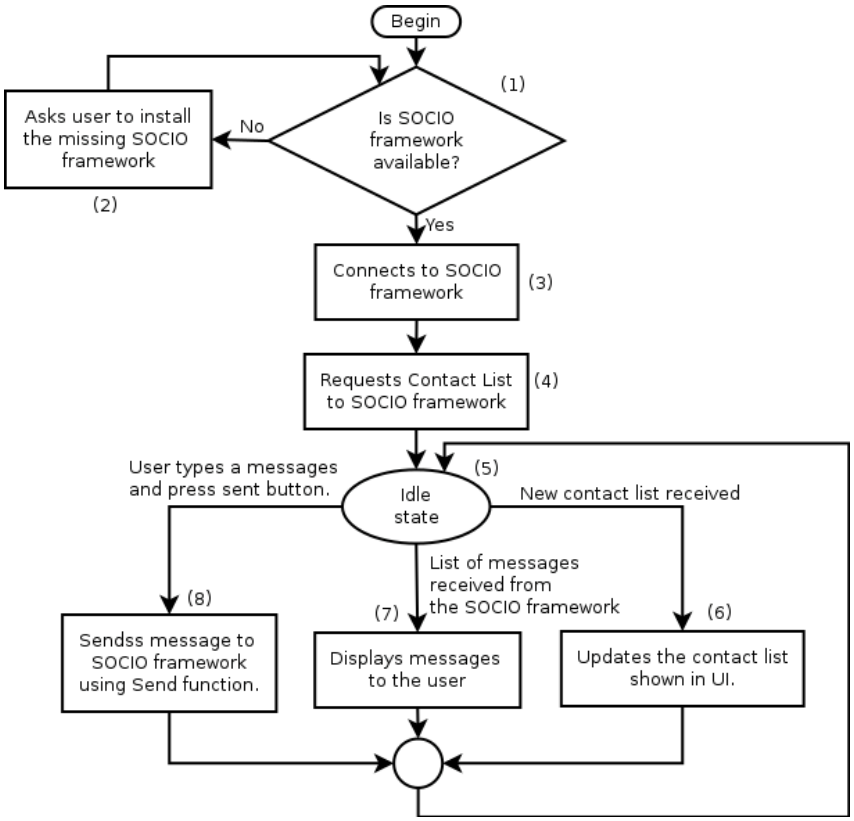

Figure 27. Oi! application operation flowchart

For a full description of Oi! code documentation, refer to Annex 1. The open source code of Oi! is available at GitHub: https://github.com/COPELABS-SITI/Oi

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

4.3. QoS and Congestion Control

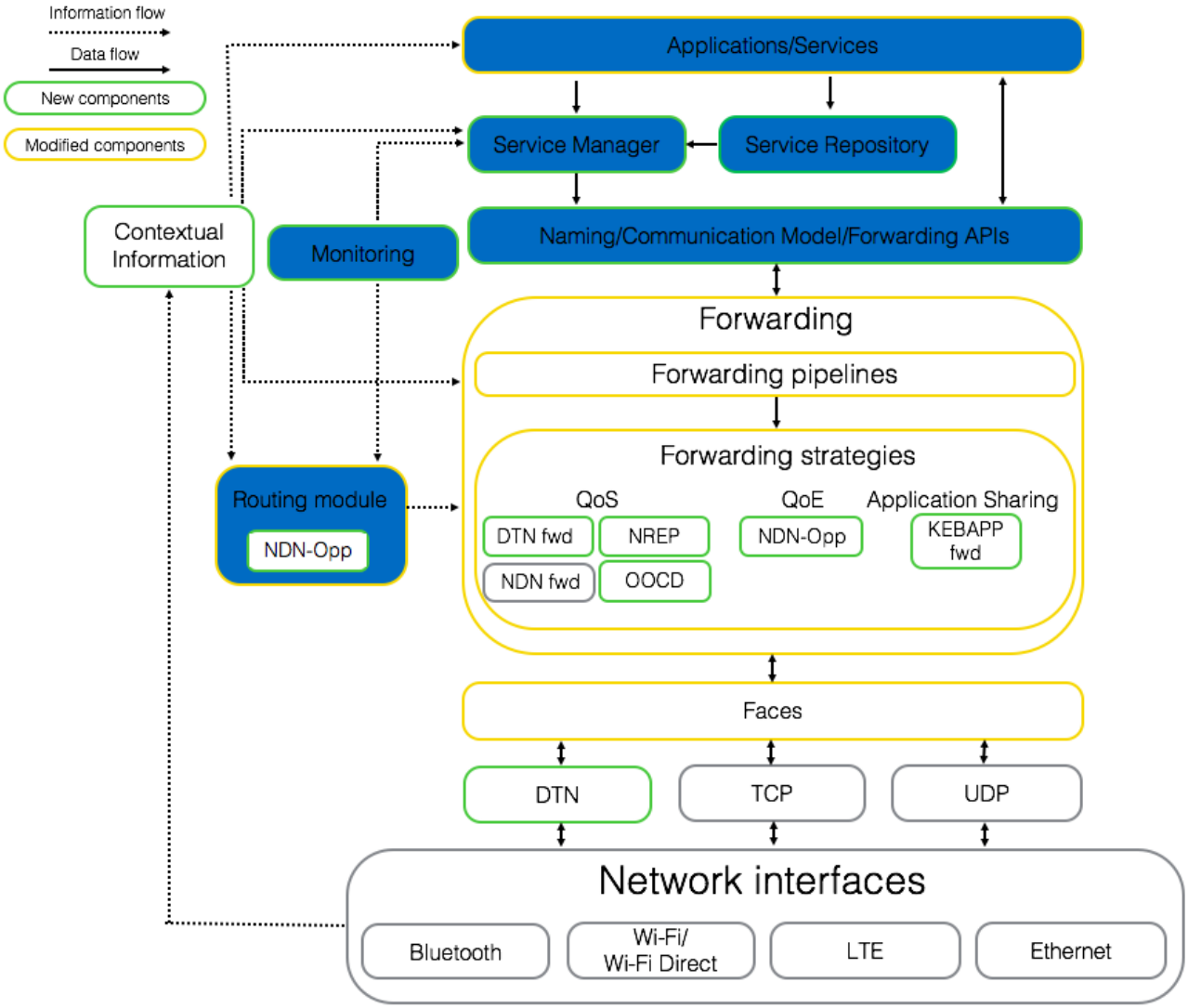

Figure 28. Service migration function blocks

As stipulated in Project Summary and Task 4.1, the UMOBILE architecture is expected to develop the needed mechanisms to supports different requirements of QoS. We address the issue by means of mechanisms implemented and deployed at the application level and network level that operate independently but compensate each other.

Application level QoS mechanisms exploit parameters that easy to measure and manipulate from the application level such as latency, inflicted load on servers, number of users interested in a given service, number of replicas of a service, physical location of replicas within the network, physical resources available in a service (memory, disk, CPU), and so on. On the other hand, **network level QoS mechanisms** are based mainly on network traffic engineering and exploit parameters that are easy to measure and manipulate at network level such as link throughput and link outages. The discussion of this section is focused only on application level

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

QoS mechanisms. Note that application level QoS mechanisms rely on both, the optimisation of the usage of available resources and the possibility of deploying additional ones such as more virtual machines.

4.3.1 Service migration

In this section we discuss the design and implementation of a service migration framework that we are in the process of implementing as a measure to address QoS requirements as expected within the work to be carried out by Task 4.1. We also explain how the features provided by the service migration framework help satisfy the system and network requirements stipulated in the D2.2 Deliverable (*System and Network Requirements Specifications).*

4.3.1.1 System and network requirements

The motivation for the implementation of the service migration framework is to enable the UMOBILE platform to satisfy some of the system and network requirements stipulated in the D2.2 deliverable. The following requirements are those related to service migration. By service migration we mean the service migration framework that we are in the process of implementing to perform service migration and replication as explained in subsection 4.3.1.

- R-6 UMOBILE systems MUST be able to ensure data reliability and availability (e.g. taking into account data usefulness - time to live; manage duplicated pieces of information) among a set of distributed surrogates.
- R-8 UMOBILE systems MUST be able pre-fetch data in order to improve service performance.
- R-10 UMOBILE systems MUST be able to deliver information within geographic regions and time frames that are relevant to different types of data.
- R-11 UMOBILE systems MUST provide users only with relevant information, i.e., matching user interest.
- R-13 UMOBILE systems MUST be able to provide the services to the end users when there is no Internet connectivity.
- R-16 UMOBILE system SHOULD be able to pre-fetch data based on user interests (e.g., parking places near recommended art gallery) and behaviour (e.g. mobility patterns), in order to reduce delays in data delivery.
- R-19 UMOBILE systems SHOULD provide information about the network status (e.g. network diameter, average path length, link bandwidth, network delay) is order to allow authorities to take corrective measures (e.g. deploy UAV infrastructure).
- R-20 UMOBILE systems SHOULD be able to create opportunistic communication infrastructures, instantaneously deployed using UAVs.
- R-22 UMOBILE systems SHOULD be able to provide local services when the system cannot connect to the Internet.

The relationship between the requirements listed above and service migration is explained in the following lines:

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

- R-6: Data reliability and availability can be achieved by means of migrating services in anticipation of potential network outages, that is, upon detection of threatening problems.

- R-8, R-16: The service migration platform implements service pre-fetching mechanism aimed at improving service performance.

- R-10: Service migration accounts for geographical locations in its service migration strategies, for instance, a service is migrated to the geographical region where users have expressed demand for the service.

- R-11: A service is migrated only when the service migration engine estimates that there is enough user interests to justify the migration costs.

- R-13, R-22: Service are migrated as close as possible to the end user (normally to the network edge) and as a measure to mitigate connection problems including slow connectivity and or no connectivity at all.

- R-19: The service migration framework will include monitoring mechanisms for collecting metrics (topology, latency, link bandwidth) that help determine the current status of the network.

- R-20: This requirement is to some extend related to the service migration framework in that, the latter is based on opportunistic service migration.

4.3.1.2 Service migration

A well-known technique, and the subject of discussion of this section, that we are exploring to build application level QoS mechanisms is service migration and replication. The main idea is to exploit the latest advances in monitoring and virtualisation technology. In particular, we take advantage of monitoring and Docker virtualization technology to tackle the problem using informative service deployment, static and live service re-deployment (migration) to move services closer to the consumers. To place service migration within a deployment context let us have a look at Figure 29.

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

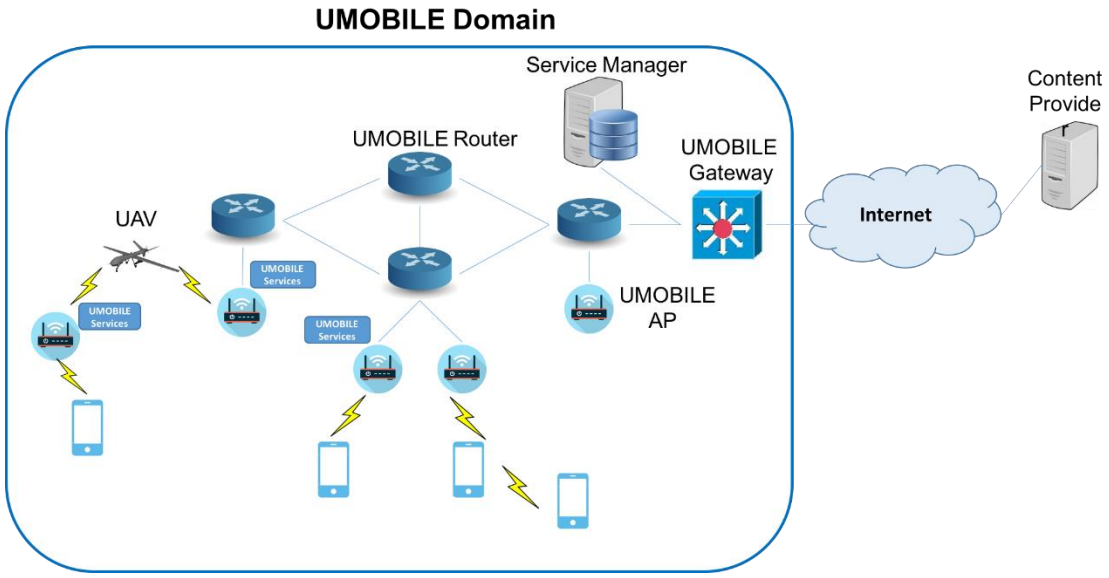

Figure 29. A view of UMOBILE platform deployment with focus on service migration

- **UMOBILE Platform**: By UMOBILE platform we understand several pieces of software that can be deployed in the components shown in the left side of Figure 29. Such pieces of software include the core components of the NDN proposal and software developed by the UMOBILE consortium. In other words, the UMOBILE platform is the software needed for building, enacting and operating the UMOBILE Domain shown in the figure.

- **UMOBILE Domain**: We conceive the UMOBILE Domain as a set of nodes deployed with UMOBILE software so that they are able to take advantage of the ICN-DTN facilities. All the components shown in the UMOBILE domain require the UMOBILE platform software to support applications such as the 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.

- **UMOBILE Gateway**: The UMOBILE Gateway is responsible for converting NDN Interest Requests to HTTP requests and HTTP responses to NDN Response.

- **ISP provider**: We assume that the UMOBILE Routers and UMOBILE Access Points/Hotspots are deployed by an ISP and overlaid over an existing IP-infrastructure. The assumption is that the original IP infrastructure is not reliable enough to satisfy different QoS (no-effort, best-effort, guaranteed QoS, etc.) demanded by end-users. Examples of end-users to be supported are the end-users involved in the emergency scenarios that UMOBILE is planning to use to demonstrate the functionality and performance of the UMOBILE platform. To simplify the problem, we assume the existence of a single IPS shared by all the components.

- **UMOBILE Router**: A UMOBILE Router is an NDN router. We assume that UMOBILE Routers are in possession of storage that they use for providing in-network storage for in-network caching.

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

- **UMOBILE Access Point (AP):** A UMOBILE AP is a UMOBILE Access Point. It is a node deployed with the UMOBILE platform and provided with storage, processing and wireless communication facilities. A UMOBILE AP can be used to store services for the benefit of end-users who can access them from mobile devices equipped with wireless communication. Within UMOBILE, we use Raspberry Pi computers to realize the UMOBILE APs. Note that not all UMOBILE APs are directly connected to UMOBILE routers.

- **Service/Content Provider:** The Content Provider is repository of services that are likely to be of interest to the end-users. Examples of such services are Oi, and Kebab. A service in the Content Provider is stored as a compressed image (e.g., Dockerised images) that can be retrieved by the Service Manager and deployed upon request. We assume that the Content Provider delegates the ISP provider the responsibility of making the services available to the end-users.

- **Service Manager:** The Service Manager is responsible for deciding what services to migrate and replicate, when and where to deploy them. It includes a Decision Engine and a Service Repository (it will be explained below).

- **End-users:** An end-user (represented by a mobile phone in the figure) is a person in possession of 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 APs over conventional HTTP.

- **UAV:** An Unmanned Aerial Vehicle is a drone provided with storage, processing and communication facilities and capable of communicating with UMOBILE Hotspots. As shown in the figure, in the UMOBILE platform UAVs are used for transporting content (for example, plain data or compressed images of services). UAVs are used for enhancing the UMOBILE platform with DTN features that are needed to satisfy QoS requirements such as availability under no time constrains.

- **End-user to End-user direct communication:** As suggested by the figure, some End-user's mobile devices might choose to deploy software for communicating with each other directly. An example of such software is KEBAPP that the mobile devices can download from the UMOBILE APs.

To support service migration, we have implemented a service migration framework that consists of the functional blocks shown in blue colour in Figure 28.

- **Application/Services:** The Application (also called Services) are compressed dockerised images produced by content providers and delegated to the Service Manager for deployment under the observance of QoS requirements. They are stored in the Service/Content Provider shown in Figure 29.

- **Service Manager:** The Service Manager is responsible for deciding what services to migrate and replicate, when and where to deploy them. It includes a Decision Engine and

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

a Service Repository. For example, it might decide to migrate a given service to one of the five UMOBILE APs shown in Figure 29.

- **Decision Engine:** At the heart of the Service Manager is a Decision Engine that makes decisions about service migration on the basis of the QoS required by each service and monitored parameters that impact their QoS. The QoS requirements are provided by the owner of the services as deployment descriptors with QoS constraints. Similarly, the monitored metrics are provided by the Monitoring component. This strategy is related to service placement algorithm (Task 4.1) running inside the decision engine.

- **Service Repository:** The Service Repository is responsible for storing compressed (dockerised) images of services subject to migration and replication.

- Service Execution**:** Images are migrated to hosts with storage and processing facilities, such a Raspberry Pi. For example, Service Execution (not represented in Figure 35) can be performed in the access point shown in Figure 36. We assume that the recipient hosts are capable and willing to receive compressed images of services from the Service Manager, uncompress, execute them and grant access to end users.

- **Monitoring:** The monitoring component is responsible for collecting metrics about parameters that impact the QoS and of interest to the Service Manager. It can be realised as a set of monitors deployed to collect metrics about the status of the parameters of interest. Typical examples of QoS parameters are the number of users requesting a given service, load inflicted on existing replicas, performance of replicas, resources of the computer executing the replicas (memory, CPU, disk), network traffic, latency, delays, bandwidth, and so on. The monitoring components should be placed in location where the metrics of interest are observable, for instance, in Figure 29, they will be placed within or near the UMOBILE Hotspots. Information collected by monitors can be used to by the Service Manager to infer and construct the current network topology.

- **Communication Model:** This component implements the communication models discussed in the Push Service section, namely: pull-based, push-Interest poling, push-Interest and push-publish data dissemination. Service Manager contacts the Communication Model to make decisions about what communication model to use to migrate a given service.

- **Naming Scheme:** This component implements a naming mechanisms capable of naming proper name for each Interest/Data message. For instance, the Naming Scheme module adds a “push” prefix for push-based communication.

- **Routing Module:** This component implements routing strategies. The Service Manager contacts the Routing Module to choose the best routing strategy. For instance, when a less than best effort service is required, the Routing Module chooses a DTN routing alternative and activates a DTN face for this service .

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

4.3.1.2 Message flow

The service migration framework relies on monitored data provided by the Monitoring component shown in Figure 30. We have implemented the monitoring component as a set of monitors deployed within the UMOBILE APs shown in Figure 29. We will explain next the flow of messages involved to collect the monitoring data. The message flow is operated as the following steps:

1) The monitoring function component periodically collects monitoring data from the UMOBILE APs by providing descriptions such as name of UMOBILE APs, collecting parameter name of request service (service monitoring) to the Naming/CommunicationModel/Forwarding APIs module.
2) The Naming/CommunicationModel/Forwarding APIs module implements a specific naming scheme to generate the corresponding Interest message and chooses a communication model. A pull-based model is used for this operation.
3) The Forwarding pipelines of a forwarding engine sends Interest message to the UMOBILE network where it is delivered to a UMOBILE AP that has the same registered name as the name in the Interest message
4) The UMOBILE AP responds to the Interest message by sending back a Data message with the information. In this example, the monitoring module collects two metrics namely CPU load and memory usage.

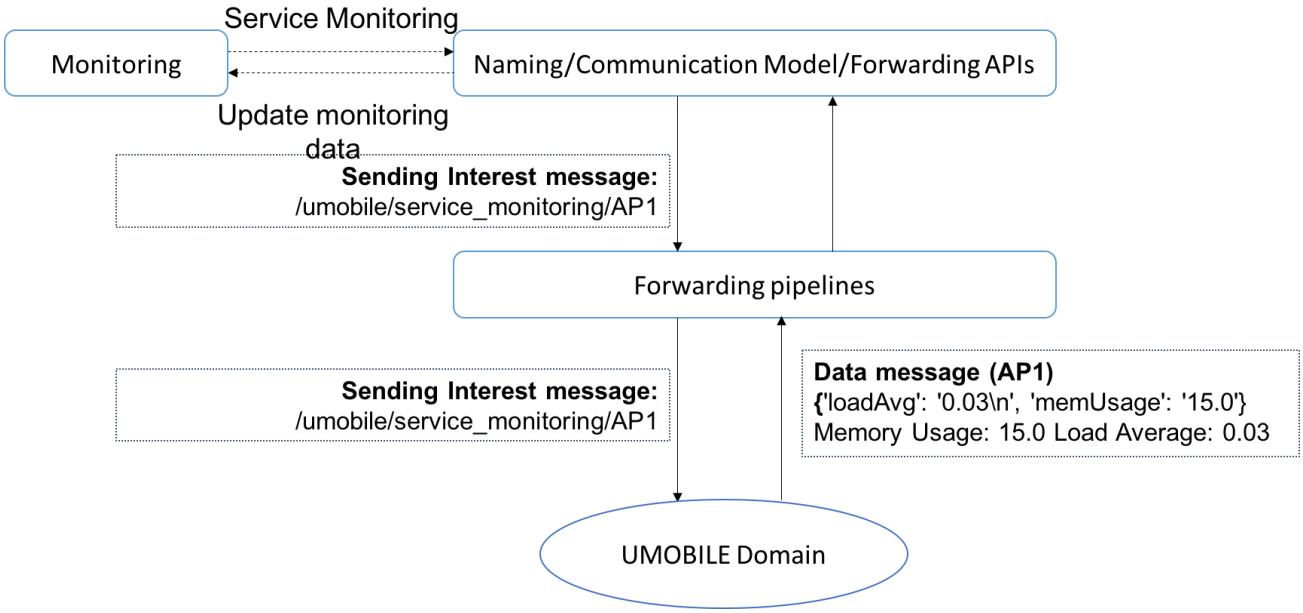

Figure 30. The message flow of monitoring system

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

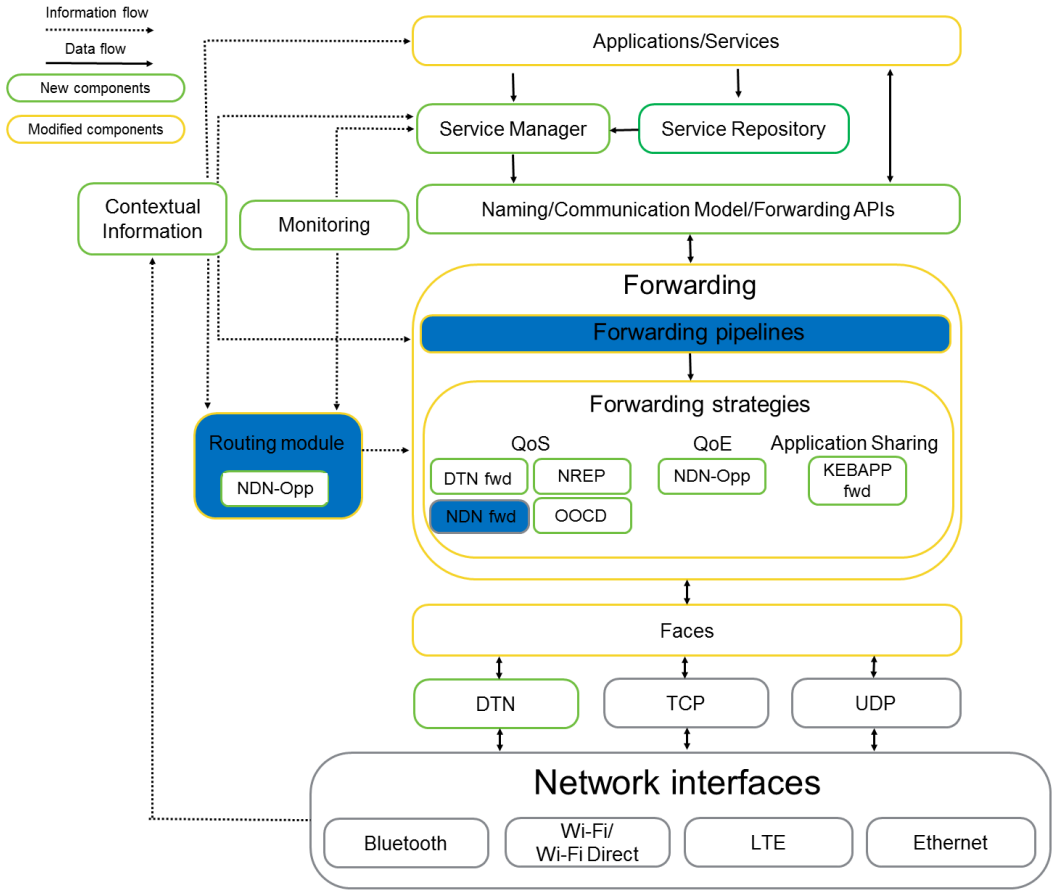

Figure 31. INRPP modules

Within the UMOBILE project, users may need to access Internet services and/or services may need to be pushed to the edge of the network. In this case, it is necessary to deal with congestion to provide services and/or push services as fast as possible. Current end-to-end (e2e) congestion control (e.g. TCP) deals with uncertainty using the "one-out one-in" principle. E2e approaches effectively deals with uncertainty by (proactively) suppressing demand and end-points have to speculate on the resources available along the e2e path.

First NDN/ICN congestion control proposals are based on hop-by-hop congestion control. Despite are based on interest shaping, they still follow the "one-out one-in" principle as well and they do not explore in-network caching capabilities to improve congestion control neither multipath capabilities.

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

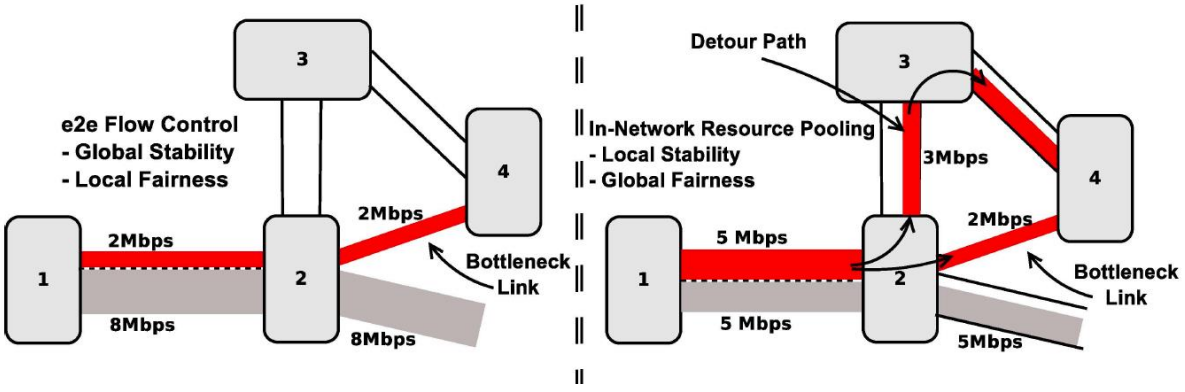

Figure 32.
Left: E2E flow control. Bandwidth is split according to the slowest link on the path.
Right: In-Network Resource Pooling. Bandwidth is split equally up to the bottleneck link (global fairness).
Detour applies to guarantee local stability.

Within the UMOBILE project, we aim to design and evaluate the In-Network Resource Pooling Protocol (INRPP)[15], which pools bandwidth and in-network cache resources in a novel congestion control framework to reach global fairness and local stability, taking profit of the hop-by-hop design and the caching capabilities inherent in the NDN networks, or adding caches (i.e., temporary storage) and breaking the end-to-end principle.

Given this functionality of in-network storage, INRPP comprises three different modes of operation:

1. push: content is pushed as far in the path as possible in an open-loop, processor sharing manner, based on the path's hop-by-hop bandwidth resources to take advantage of under-utilised links.
2. store and detour: when pushed data reaches the bottleneck link, the excess data is cached and simultaneously forwarded through detour paths towards the destination.
3. backpressure: if detour paths do not exist or have insufficient bandwidth, the system enters a backpressure mode of operation to avoid overflowing of the cache. During the backpressure mode, the nodes enter a closed-loop mode, where an upstream node sends one data packet per one received ACK to the backpressuring downstream node.

The INRPP congestion control is out of the scope of WP3 and it is being developed within the Task 4.1 "Providing different levels of QoS and flow controlr of the WP4.

4.3.2.1. Modules to be modified

- Routing module: The routing module should be modified in order to provide routing information not only for the shortest path, but also for the one-hop detour path to the content.
- Forwarding strategy: The NDN forwarding strategy should be modified in order to forward content using the three modes of operation defined by INRPP.
- PIT: PIT tables should be modified in order to follow the "breadcrumbs" through the detour path as well.
- FIB: FIB tables should provide information for the detour paths as well.

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

4.3.2.2. Network requirements

The INRPP proposal does not fulfil any specific network requirement of D2.2. However, it helps sending information in an efficient way, improving the QoS perceived by the users, an inherent requirement in the UMOBILE project.

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

4.4. Routing module

This section provides a description of the routing module being developed for the UMOBILE mobile node, in order to allow the exploitation of any wireless communication opportunity (e.g. over WiFi direct links, WiFi infrastructure or over a DTN link). Since the NDN framework was selected as the de-fact reference of an information centric networking architecture in the UMOBILE project, the UMOBILE routing engine is embedded in an enhanced version of the NDN framework for Android devices.

In this section we start by describing, in section 4.4.1, the *NDN framework for Opportunistic Networks* (NDN-Opp), which is being developed aiming to support opportunistic forwarding based on users' interests and their dynamic social behaviour. The NDN-Opp framework includes some changes in relation to NDN in order to enable social-based information-centric routing over dynamic wireless networks. In order to evaluate the feasibility of using social-aware routing metrics in some of the UMOBILE use-cases we started by implementing an opportunistic routing framework, called SOCIO, based the Time-Evolving Contact Duration algorithm [10] used by the dLife protocol [3,11]. To evaluate the operation of SOCIO, a short messaging application was developed [1], since this type of application is used in several UMOBILE use-cases. The SOCIO framework is described in section 4.4.2 (the oi! application is described in section 4.2.4)

Section 4.4.3 provides initial highlights to the work being done to develop a smart routing approach aiming to handle the heterogeneous set of connectivity that an UMOBILE mobile node may face, such as WiFi direct for communication with peer nodes, DTN links for delay tolerant communications, and WiFi links for communication with the Internet (based on an NDN edge router).

4.4.1. NDN-Opp: NDN framework for Opportunistic Networks

The NDN-Opp framework is being developed to fulfil the following UMOBILE requirements, as listed in deliverable D2.2:

R-2: UMOBILE systems MUST be able to exchange data by exploiting every communication opportunity through WiFi (structured, direct), 3G and bluetooth, among UMOBILE systems, operating even in situations with intermittent Internet connectivity.

R-3: UMOBILE systems MUST be able to exchange data taking into account user data interests and context.

R-5: UMOBILE systems MUST have an interface to support the following T3.2 applications: Chat; File exchange/synchronization; Content request/publish.

R-7: UMOBILE systems MUST be able to make messages available to different receivers simultaneously.

R-11: UMOBILE systems MUST provide users only with relevant information, T3.3 i.e., matching user interest.

R-15: UMOBILE system SHOULD be able to provide users only with information that matches their interests (e.g. art exhibitions).

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

R-17: UMOBILE mobile systems SHOULD be able to sense user context (geo - T4.2 location, relative location, proximity, social interaction, activity/movement, roaming, talking) in a non-intrusive manner.

The current specification of NDN-Opp aims to handle the dynamics of opportunistic wireless networks by forwarding interest packets towards best neighbours, which are selected based on their probability to deliver packets to a node carrying the interested data. This means that the current specification of NDN-Opp makes use of:

- The existing best route forwarding strategy to deliver interest packets. In this case the cost is related to the social weight computed by the social proximity module.
- The NDN "breadcrumb" approach to deliver data packets based on the information stored on the PIT.

The cost of each name prefix (stored in the FIB) is the value of the social weight compute for the best next hop, as computed by the social proximity module.

NDN-Opp provides support to the NDN natural pull communication model (e.g. data sharing applications) and to the push communication model used by interactive applications, such as the short message application, Oi!, developed with SOCIO [1]:

- Support to pull model: NDN-Opp allows a receiver to express its interest in receiving some type of data, by sending Interest packets to name prefixes such as /app/Now@/Topic used by the Now@ application (data sharing application being developed to demonstrate NDN-Opp support to pull communication models).
- Support to push model: NDN-Opp allows a node to express its interest in receiving data from a specific application (name prefix). For instance, if a node Nx would like to start using the short messaging application, Oi!, it send an Interest packet with name prefix /app/oi!/Nx/ or of type /app/oi!/Nx/Ny if node Nx would like to receive Oi! short messages only from node Ny. **To support the pull model, NDN-Opp included the concept of Long Lived Interests**, allowing one Interest to serve more then one data packet.

In order to handle the intermittent nature of the wireless links, the NDN-Opp includes the concept of virtual faces, which send and receive packet via a WiFi direct link. Each virtual face can be in two states (ON and OFF), and is named after the identified of the neighbour node. Since packets may not be handle to be dispatched after a SendInterest or SendData issued by the forwarder, each virtual face implements two queues:

- An Interest Queue (IQ), which stores Interest packets to be sent.
- A Data Queue (DQ), which stores pointers to the Data block (hold in the Content Store) to be send.

The NDN face management creates virtual faces after the first contact with a new neighbour node.

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

In summary, NDN-Opp encompasses the following novel contributions to the NDN framework in order to allow its operation over opportunistic networks:

- Novel social-aware routing engine, which is able to implement different algorithms able to select best routes based on the social weights computed from a social proximity module.
- Novel forwarder able to support opportunistic communications.
- Integrates the concept of Long Lived Interest in order to support push communication model.
    - Users manifest their interest in getting data from some type of application (e.g. short messaging) or from some specific node.
    - Interests persist in the network for as long as established by the Interest source by setting a Long Lived Interest parameter.
    - While Interests persist in the network, data can flow towards the established intermittent path.
- Integrates the concept of virtual intermittent face, which can be in two states (ON/OFF).
    - Virtual interfaces include queues of interests and data, aiming to speed up transmission over short-lived wireless links.
- Makes use of:
    - Best route forwarding strategy to handle Interest packets based on social weights;
    - The default "breadcrumb" approach to forward data..
- Two new fields in the PIT and FIB:
    - Social Weights in the FIB
    - Long Lived Interest (TTL) in the PIT.
- Novel NDN face: WiFi direct.
- Updated Bluetooth face (we need to gather info from neighbour, and not to send/recv data).

The operation of an NDN-Opp node, as illustrated in Fig. 32, goes over a main set of operations when it gets in contact with another mobile NDN-Opp node (e.g. when a virtual face goes UP). As shown Fig. 32 the operation of an NDN-Opp node is similar to the basic NDN operation: the node state is changed by the arrival of an Interest or a Data packet.

The major constraint of the NDN operation over opportunistic networks is the intermittent nature of wireless connectivity. In order to accommodate this property of wireless links, virtual faces are used. In this case, each mobile node has a virtual face towards to each encountered

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

node (e.g. Face A and C in the case of Node B in Fig 32). Each virtual interface implements an Interest Queue (IQ) and a Data Queue (DQ), as depicted in Fig 33.

This means that in NDN-Opp the existing of virtual faces creates an indirection: The normal operation of NDN finishes with a packet being send through a face; with NDN-Opp the packet handling operations end with the packet being stored in the respective queue of the selected virtual interface. Packets are transmitted by serving the interest and data queues of the virtual face that got active.

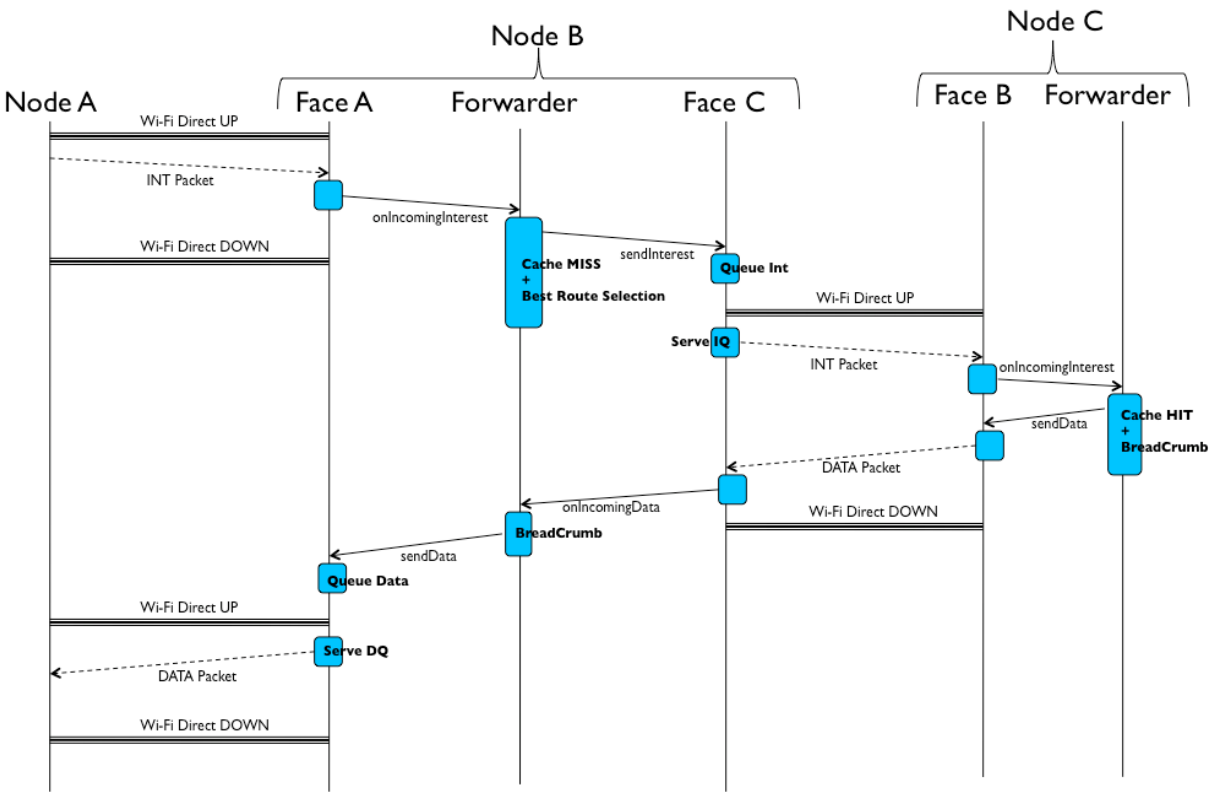

Figure 32. Illustration of NDN-Opp operation

The **forwarder** is able to send and receive packet from several virtual faces, as well as the application face. Upon the reception of an Interest packet, the normal NDN state machine is used, but the PIT will store also the information related to the duration of the Interest (LLI). To forward Interest packets, the best route forwarding strategy is used, in which the FIB stores the cost associated to each name prefix (the cost is computed by the routing engine). When a data packet arrives, the normal operation occurs, being the output virtual face selected based on the information stored in the PIT.

The **social-aware routing engine** is responsible to select the best neighbour to reach data related to a certain name prefix. For this propose the routing engine can run any type of social-aware opportunistic routing algorithm [2,12]. The first algorithm used by NDN-Opp is the Time-Evolving Contact Duration approach (TECD) [10] used by the dLife and SCORP protocols [3,11].

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

Computing routing information in opportunistic network based on social interactions has great potential as less volatile proximity graphs are created [13]. We believe that the accuracy level of social interactions is mainly dependent on the statistical duration of contacts over different periods of time, since people have daily habits that lead to a periodic repetition of behaviour, and will more accurately reflects the real evolution of social ties than relying solely on the contact between nodes or well-defined social structures.

The TECD algorithm aims to compute the Social Weight (SW) of a node towards a name prefix (which can represent another node, such as used by short messaging applications). By computing SWs, the TECD algorithm is able of capturing the evolution of social interactions in the same daily period of time over consecutive days, by computing social strength based on the average duration of contacts. The TECD algorithm can also be used to capture the Importance of any node (TECDi) in a daily sample, based on its social strength towards each user that belongs to its neighbour set in that time interval, in addition to the importance of such neighbours. The TECDi is based on the PeopleRank function [14]. However, TECDi considers the social strength between a user and its neighbours encountered within a daily sample, while PeopleRank computes the importance considering all neighbours of that node at any time.

The social-aware routing engine is responsible for the computation of social weights based on information collected about neighbour devices. For that the routing engine makes use of an interface towards the Contextual Manager developed by the UMOBILE architecture.

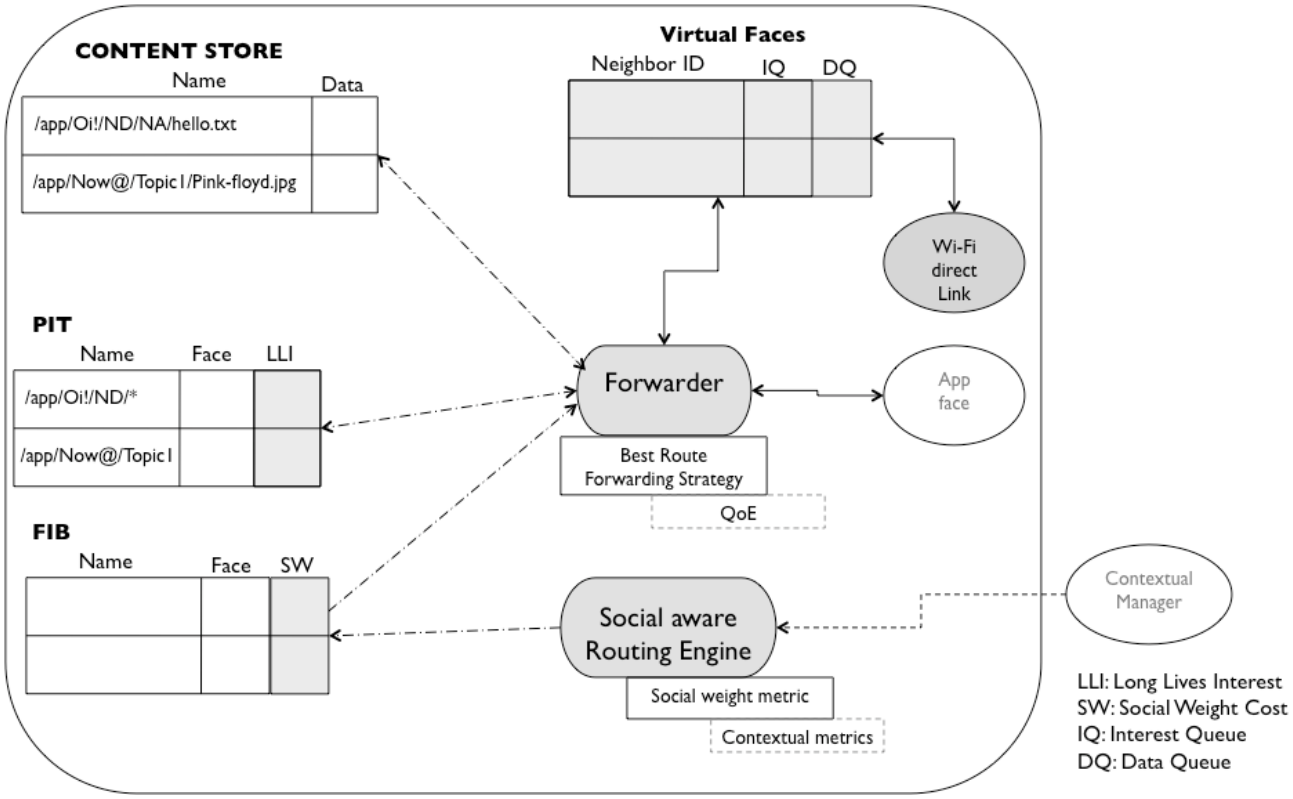

Figure 33. NDN-Opp design

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

The **Contextual Manager** shall be able to provide information about neighbour devices, to ensure the proper operation of the routing engine. The basic operation of the NDN-Opp routing engine requires information about the average contact duration between pair of nodes in specific daily samples, as illustrated in Fig. 34.

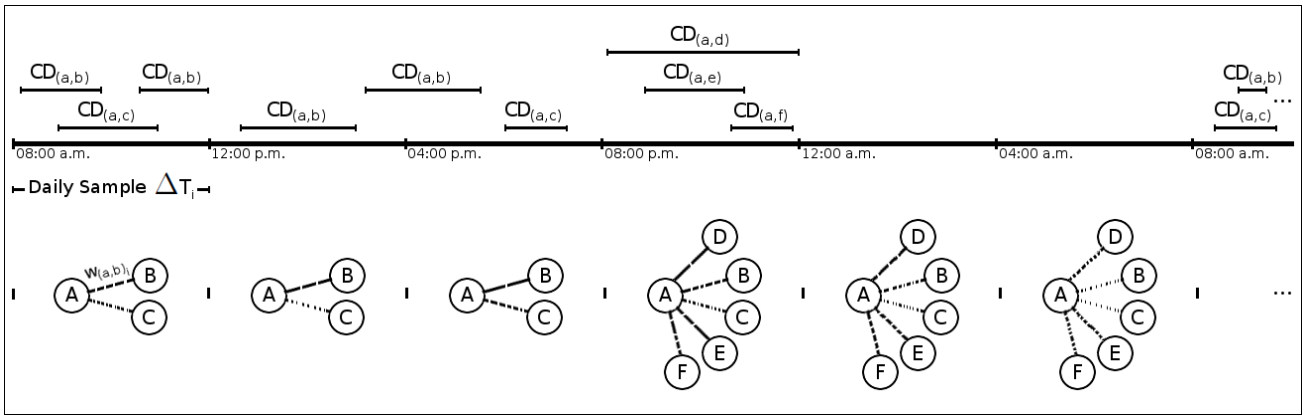

Figure 34. Contacts a node A has with a set of nodes x in different daily samples

The operation of the **social-aware routing engine** is based on a social weight measurer, a social weight repository and a gatherer of information about social weight and data carried by neighbour nodes:

- **Social Weight Measurer (SWM)** – responsible for keeping track of the contact duration information provided by the Contextual Manager. SWM maps this duration to the name prefixes carried by the encountered nodes. Based on that, this component computes the level of social interaction of the current node towards the name prefixes carried by encountered peers. This social weight determines how good the encountered node is to reach nodes that carry data related to such name prefixes. The social weight computation takes place at every hour as to allow for a better view of the dynamic social behaviour of users.
- **Social Weight Repository (SWR)** – responsible for storing the list of name prefixes that the current node comes across (obtained upon encountering a peer). SWR holds the name prefix, time it has been first encountered, and the cumulative duration of within a specific hour. In the case, the current node still 'sees' such name prefix (i.e., respective peer node is still in the vicinity), the time of first encounter is updated and contact duration is accounted by SWM for the new hour.
- **Social Weight and Carried Data Gatherer (SWCDG)** – responsible for obtaining the list of name prefixes and social weight towards them of encountered node. This element is also responsible for obtaining information concerning the content carried by encountered nodes.

The computation of social weights is triggered when a specific face notifies the SWM of the presence of a neighbour node. The SMW creates entries for this encountered node in the SWR considering the name prefix information obtained by the SWDCG. The SWM keeps track of the

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

contact duration between current and encountered nodes by querying the Contextual Manager and updates the SWR accordingly. Since nodes (i.e., their users) present different patterns of behaviour in different time periods, the computation of social weights takes place in an hourly fashion. Information obtained by the SWCDG from the encountered node is used to populate PIT and the content store.

Future work in the development of the NDN-Opp framework will be done based on exploiting different contextual information provided by the Contextual Manager for the development of novel routing approaches. The major action points are:

- **Interface with the Contextual Manager**: The collection of context information about neighbour nodes, done currently by scanning the Bluetooth face, should be done also by scanning the WiFi direct face. The goal is to allow the collection of neighbour information to be done over the same wireless range as the one used to exchange data. Subsequent evaluation about the performance of both approaches will be done based on a UMOBILE testbed.
- **Novel opportunistic routing mechanism**. Currently NDN-Opp makes use of the social-aware algorithm used by existing routing approaches (dLife/SCORP) [3,11]. We will investigate an alternative proposal for the NDN-Opp framework based on novel strategies to forwarding interest packets in dynamic networks. In a first stage we will keep using the NDN default "breadcrumb" strategy to forward data. In a second stage we will investigate a more intelligent method also to forward data, aiming to exploit any contact with a neighbour node able of forwarding data towards the receiver(s), and not just the wireless contact with the node that previously sent the Interest packet (which in the approach followed in the default NDN "breadcrumb" approach). The performance of both alternatives will be assessed in real scenarios.

4.4.2. SOCIO: Social-aware opportunistic networking framework

The SOCIO framework follows a modular design, aiming to allow easier extensions, such as the inclusion of other routing schemes, a novel algorithm to infer social proximity, or a new method to manage data.

As illustrated in Fig. 35, the SOCIO framework comprises **an application interface** (used by Oi!). Thought this interface, SOCIO gets application data, which is stored in the Content Manager (CM) module, which is responsible to manage all the data to be sent, as well as received. The primitives `Send(destination, message)`, allows CM to build a message to be sent to a given destination node; `Receive(message)`, allows CM to pass the received data to the application interface; and `GetContactsList()`, fetches a list of known contacts to display on the UI. It is worth mentioning that a new application (e.g., dissemination app) will make use of these primitives provided by SOCIO's CM.

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

Based on collected data, CM starts by composing a new message, which is stored together with the `<sender, destination, timestamp>` tuple. The sender and destination fields are the deviceID of the message's source and recipient devices, respectively. The timestamp is used to detect possible message duplication. This file is then stored in the Internal Storage (IS). All messages to be sent are stored in toOthers.xml file. Upon getting a new message from the WiFi Direct through the primitive `Receive(received message list)`, CM considers the tuple `<sender, destination, timestamp>` to filter messages. The filtering process is needed to check: i) whether the same message has been previously received; ii) if the message destination is the device itself, in which case checks if UI is available. If so, CM passes the received message to be shown in UI. Otherwise, CM saves the message to the localCache.xml file so it can be delivered when UI is available.

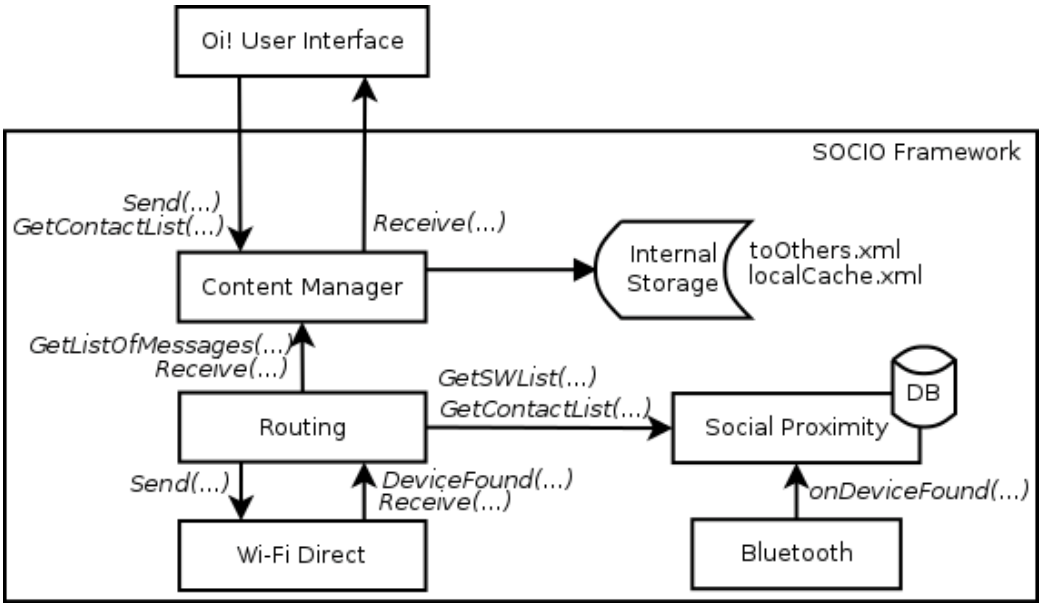

Figure 35. SOCIO high-level design

Routing (RT) module implements dLife, a social-aware opportunistic algorithm based on daily routines [3,11]. Forwarding decisions are based on the level of social interaction among users: message replication only takes place when the social weight of an encountered node towards a specific destination is higher than the social weight of the current carrier of the message towards such destination. When a device is found, through the primitive `DeviceFound(deviceID, SW list)` triggered by the WiFi Direct module, RT gets the identification and list of social weights of the device and by using the primitives `GetListOfMessages(list of destinations)` and `GetSWList(list of destinations)`, RT gets a list of carried messages and the social weights towards a specific list of destinations, respectively, to help with forwarding decisions. When the forwarding decision is made, RT uses `Send(destination, message list)` to provide the WiFi direct module with a list of messages to be replicated to the destination (i.e., encountered Oi! device).

WiFi direct module comprises the notification and data exchange phases. In the notification phase, this module detects the presence of other SOCIO devices and notifies RT. As for the data

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

exchange phase, this module i) considers the list of messages from RT to replicate to an encountered device; and ii) provides CM with a set of messages received from another device.

Social Proximity (SP) module computes the social weight towards encountered devices through Bluetooth sightings and stores this information on its database. This social weight is given by the Time-Evolving Contact Duration (TECD) [10], which via the primitive `onDeviceFound(deviceID, encounter time)` compares the encounter time against the current time after a Bluetooth scan to determine the contact duration towards a specific device (i.e., deviceID). Social weight is computed in a hourly base, which can include different contacts of various durations. SP provides the UI with a list of known contacts through CM, and RT with a list containing the potential destination devices and the social weight towards them.

Bluetooth (BT) module is used to detect the presence of neighbouring devices. BT provides SP with the identification of the neighbour device and the time of encounter.

A full description of SOCIO code documentation is provided in Annex 1. The open source code SOCIO is available at GitHub SOCIO: https://github.com/COPELABS-SITI/SOCIO

The next subsection provides a description of the operation of SOCIO major components: the Content Manager; Routing Module; Social Proximity Module; WiFi Direct Interface; Bluetooth Interface.

4.4.2.1 Content Manager

The Content Manager (CM) is the module that provides the primitives that allow SOCIO to exchange data with a set of applications. As shown in Fig. 36 this module starts by initializing the TTL verification mechanism (1), which allows the removal of stale data, and the Routing module (2).

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

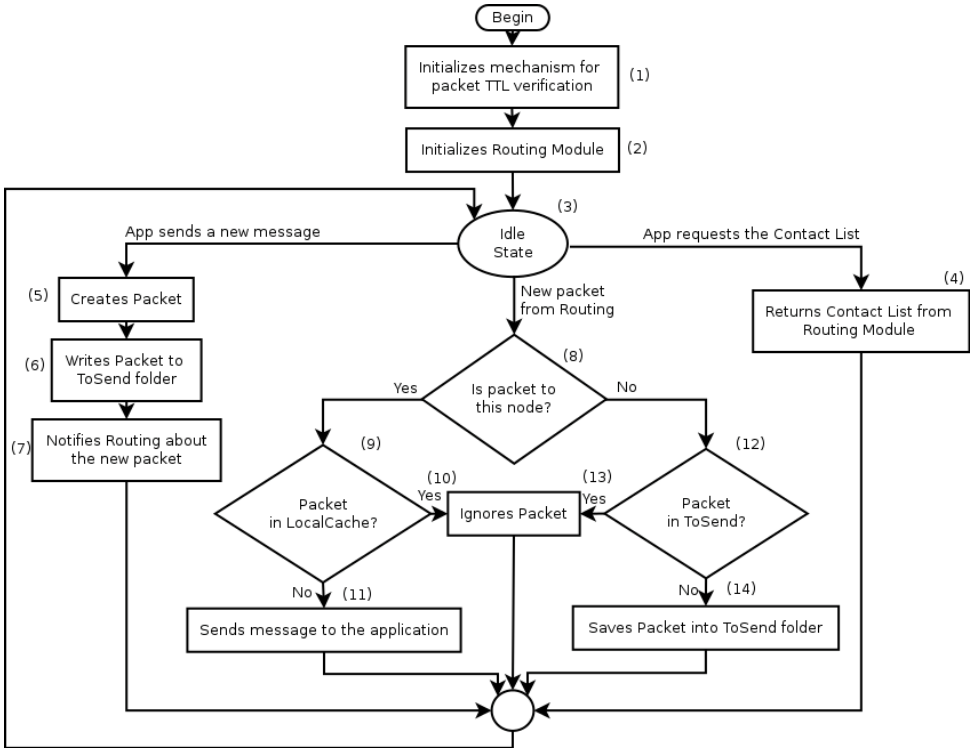

Figure 36. Content Manager operation

CM then reaches the idle state (3), in which three possible actions may take place: Application request contact list: CM provides the contact list obtained from the routing module (4).

Application produces an outgoing message: CM creates the packet with message in the payload (5), writes this packet to the outgoing folder - `toSend` (6), and notifies the routing module on the existence of a new packet to be disseminated (7).

New packer arrived: CM checks whether the current node is the recipient for this new packet (8). If so, it checks whether the packet has been received before (9). If this packet has been received (already in `LocalCache`), CM ignores it (10). Otherwise, CM extracts the message from the packet, and sends it to the application (11). In the case the packet is too a different destination, CM checks whether the packet has been received before (already in `ToSend`, 12). If so, CM ignores the packet (13). Otherwise, it saves the packet in the `ToSend` folder for further dissemination (14).

4.4.2.2 Routing Module

The routing module (RT) is responsible to deciding whether a packet is forwarded to an encountered device. This decision is made based on the levels of social interactions between the communicating parties and the packet's destination. The exchange of meta-data and packets take place over WiFi.

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

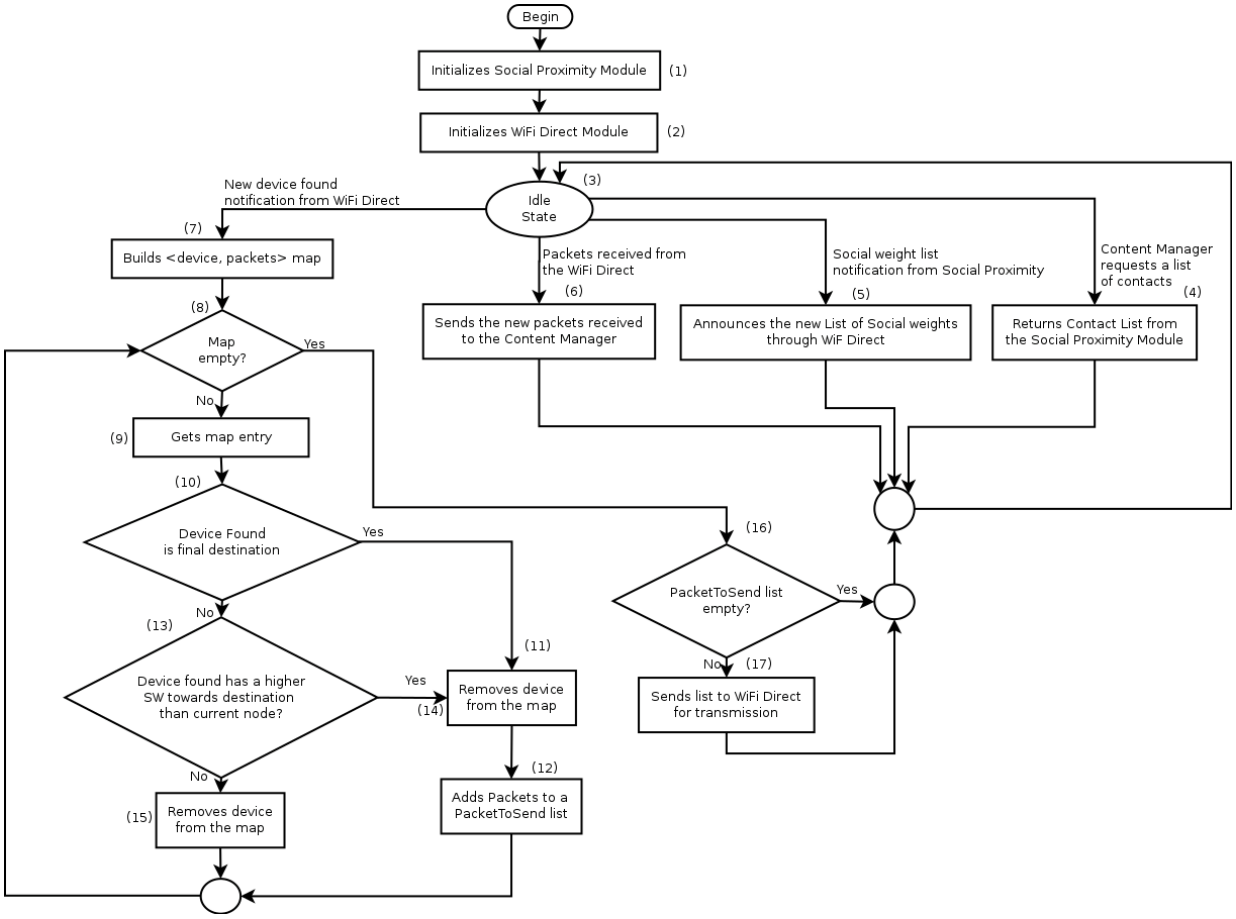

Figure 37. Routing operation

As illustrated in Fig. 37, RT starts by initializing the Social Proximity (1) and WiFi Direct (2) modules, shifting to an idle state (3). While in idle state, RT expects one of the four actions to occur:

CM requests contact list: RT provides the contact list obtained from the social proximity module (4).

Social weight list notification received from Social Proximity module: RT announces the received SW list to neighbouring peers as this information is used to perform routing decisions (5).

Packets received from WiFi Direct module: RT sends these packets to CM, which decides if they are new or have already been received (6).

New device found notification received from WiFi Direct module: RT builds a map (with devices and packets towards them), which includes the found peer, and users that the found peer has listed on its social weight list and that the current node also has on its `ToSend` folder (7). RT then checks whether the map is empty (8). If not, RT gets the first map entry (9) and check whether the <device> matches the device found (i.e., final destination, 10). If so, RT removes <device> from map (11), and adds the respective packets to the `PacketToSend` list (12). In the case that <device> does not match the device found, RT checks whether the device found has

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

higher social weight than the current node towards <device> (i.e., destination, 13). If so, RT removes <device> from map (14), and adds the respective packets to the `PacketToSend` list (12). Otherwise, RT only removes <device> from map (15). This process continues until the map is empty. Then, RT checks whether the `PacketToSend` list is empty (16). If not, RT sends this list to WiFi Direct module for transmission (17).

4.4.2.3 Social Proximity

The Social Proximity module (SP) is responsible for computing the social weights towards other devices. For that, it relies on Bluetooth scans to detect neighbouring devices and keep track of the contact duration between users.

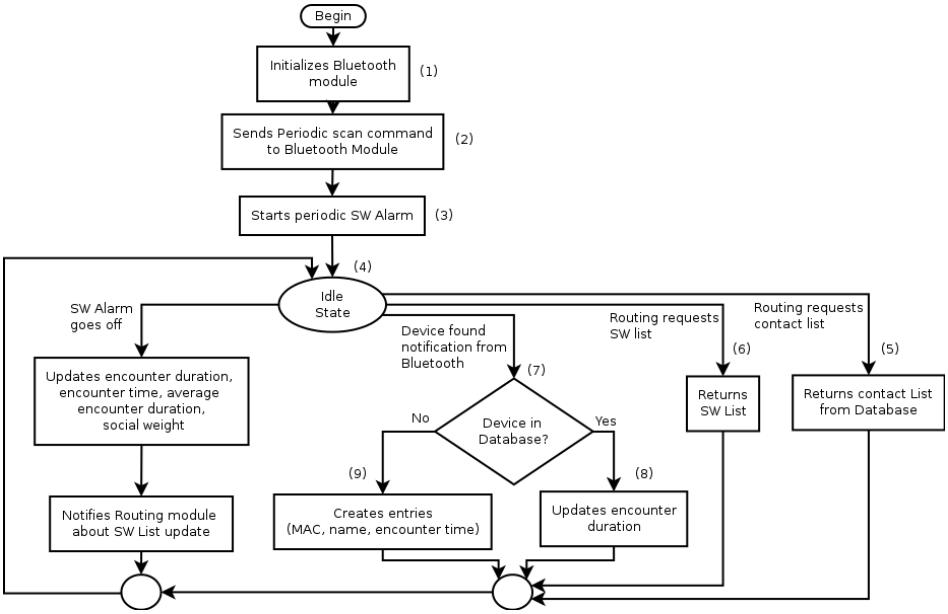

Figure 38. Social Proximity operation

As illustrated in Fig. 38, SP starts by initializing the Bluetooth module (1) and sending it a command to start periodic scans detecting neighbouring peers (2). After that, SP starts the periodic alarm for social weight computation (3), and shifts to idle state (4). Social Proximity operation flowchart. While in idle state, SP may receive one of the four actions:

RT requests contact list: SP returns a list of contacts based on the information of devices added to its Database (5).

RT requests SW list: this request may come with a set of specific devices (for forwarding decisions), or it may require a complete list (for announcing the current node's social information). Either way, SP responds by returning a list updated social weights (6).

Device found notification received from Bluetooth module: SP checks whether the found device is already in the Database (7). If so, SP updates encounter duration information for this device (8). Otherwise, the device has not been seen before, and SP then creates entries (9) in the Database with its information (MAC address, name, encounter time).

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

Periodic social weight alarm goes off: SP updates encounter and social weight information to all devices it has on the Database (10), and notifies RT about this update (11).

4.4.2.4 WiFi Direct Interface

WiFi Direct is responsible for transmitting and receiving packets to/from others devices by using the WiFi P2P technology.

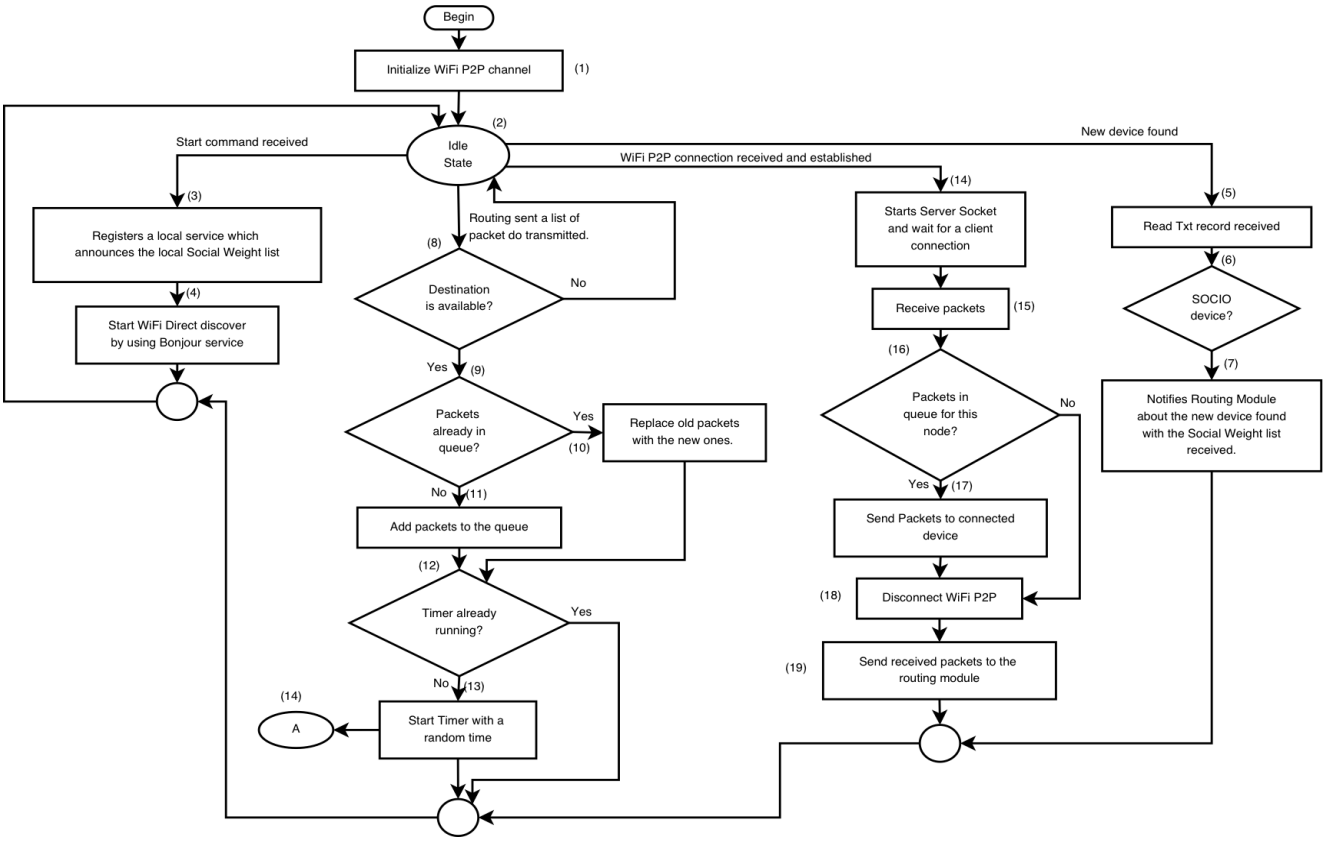

Figure 39. WiFi Direct operation

As illustrated in Fig. 39, WiFi Direct starts by initializing a WiFi P2P channel that will be used by this module to announce local services, receive neighbour services and also to establish WiFi P2P connections (1). After this initialization process finishes, WiFi Direct shifts to an idle state (2).

WiFi Direct operation flowchart.

While in idle state, WiFi Direct expects one of the four actions to occur:

Start command received from the RT: RT provides the contact list obtained from the social proximity module to be announced as local service (3). This start command also initializes the discover process from Bonjour Service (4). This process will detect neighbouring devices that have services being announced.

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

New device found: After WiFi Direct detects a new device nearby, it reads the TXT record received (5) and checks if there is a SOCIO service available (6). If so, WiFi direct notifies RT about this new device by sending the peer info and the SW list available inside the received TXT record. (7).

List of packets received from RT: WiFi Direct checks if remote destination is still available (8). If so, WiFi Direct verifies if it already has packets inside the queue to send to the remote device (9) and it replaces the old packets with these new ones (10). If so, it just adds these packets to the queue (11). Otherwise, WiFi Direct ignores these packets since it cannot reach the remote device. By having new content to send, WiFi Direct checks if the timer is currently running (12), starting it if that is not the case (13). The purpose of this timer, further detailed next, is to avoid WiFi collisions (e.g., two devices trying to connect to each other at the same time) by adding a small random delay to each connection. If timer is already running, it means that there is a pending connection and these packets will be transmitted when this pending transmission finishes.

A WiFi P2P connection received: WiFi Direct module creates a server socket and waits for a client (14). When the client connects to this server socket, the WiFi Direct module receives the packets from the remote device (15). After receiving all packets, WiFi Direct searches the queue for packets to this remote device (16). If there are packets to be sent, then WiFi uses the server socket to transmit these packets to the remote device (17). Otherwise, WiFi Direct closes the server socket and disconnects the WiFi P2P connection (18), and sends the received packets to the RT module (19).

The timer mentioned when WiFi Direct has packets to be sent, is illustrated in Fig. 40. WiFi Direct operation flowchart (thread). When the timer finishes (1), a WiFi P2P connection is created to the remote device (2). When the WiFi P2P connection is established, WiFi Direct creates a client socket (4) and starts transmitting the packets (5). After transmitting all packets, WiFi Direct checks if the remote device has packets to send. If so, it receives the packets (6) and send them to RT (7). Independently whether the remote device has packet to send, after this process the WiFi Direct checks if the queue is empty (8). If there are still packets to be sent to other remote devices, the WiFi Direct starts again the timer with a small random delay (9). Otherwise, the thread is closed.

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

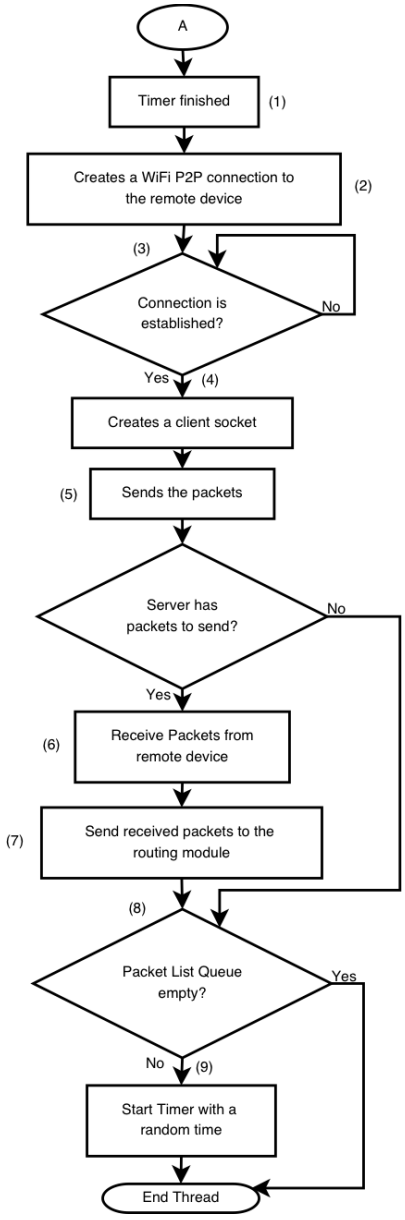

Figure 40. WiFi Direct operation flowchart (thread)

4.4.2.5 Bluetooth Interface

BT is responsible for detecting the presence of neighbouring devices for the purposes of social weight computation carried out by the Social Proximity module. As illustrated in Fig. 41, BT starts by initializing the Bluetooth communication interface (1) and registering the framework UUID (2). This UUID is used to identify among the neighbouring devices which are actually running the framework.

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

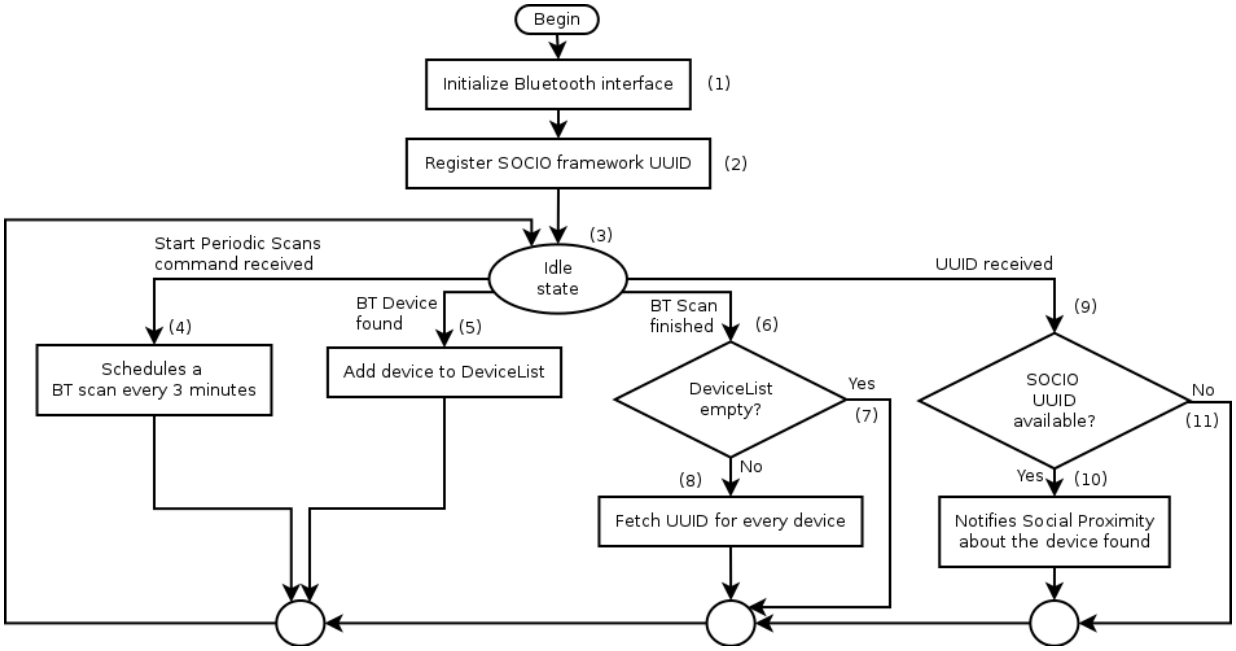

Figure 141. Bluetooth operation

Then, BT shifts to the idle state (3), where it expects for one of the four actions to take place:

Periodic scan command received from social proximity module: BT starts scanning for devices, and sets a scan to be performed at every 3 minutes (4). Bluetooth device found: BT adds the found device to the `deviceList` (5).

Bluetooth scan finished: BT checks whether the `deviceList` is empty (6). If so, it goes to idles state (7). Otherwise, BT fetches the service UUIDs for every device in `deviceList` (8).

UUIDs received: BT checks whether the SOCIO UUID is among the received ones (9). If so, that means that the device is running SOCIO, and BT notifies the Social Proximity about such device (10). Otherwise, BT shifts back to the idle state (11).

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

5. Conclusion

This document covers the progress of the project so far towards the development of the unified UMOBILE architecture. In particular, we first present the overall vision of the architecture and then describe in detail the different components that have been developed, their status and the next steps for their integration. For all components that have been deployed, we also attach the relevant code, as well as their manuals.

This is the initial version of the UMOBILE implementation. At this stage, our version includes at-least initial implementation of all critical UMOBILE modules, and in particular, the integration of NDN and IBR-DTN, the migration of a simple web server, the Oi! app and the SOCIO framework, as well as the PerSense Mobile Light service. The implementation so far demonstrates that:

- ICN/DTN integration is feasible indeed. More specific functions and services for a wider range of UMOBILE scenarios will be integrated shortly;
- Service migration is feasible;
- UMOBILE representative applications that incorporate different interfaces, including WiFi direct, have been developed;
- A socially-aware opportunistic communication framework has been implemented and can be exploited;
- A service that assists in performing network contextualization has been developed, and
- Push services can be successfully integrated into NDN.

All individual modules are the building blocks that lead us towards reaching our final goal of a unified information-centric, delay-tolerant networking platform offering extended services to its users. Further optimizations will be performed during the integration, optimization and evaluation period. The final integrated version will be submitted to the European Commission as D3.2 on M30.

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

References

[1] Waldir Moreira, Manuel de Souza, Paulo Mendes, Susana Sargento, "Study on the Effect of Network Dynamics on Opportunistic Routing", in Proc. of AdhocNow, Belgrade, Serbia, July 2012

[2] Waldir Moreira, Paulo Mendes, Susana Sargento, "Social-aware Opportunistic Routing Protocol based on User's Interactions and Interests", in Proc. of AdhocNets, Barcelona, Spain, October 2013.

[3] Waldir Moreira, Paulo Mendes, Susana Sargento, "Opportunistic Routing based on daily routines", in Proc. of IEEE WoWMoM workshop on autonomic and opportunistic communications, San Francisco, USA, June, 2012.

[4] Ioannis Psaras, Lorenzo Saino, Mayutan Arumaithurai, KK Ramakrishnan, and George Pavlou. Name-based replication priorities in disaster cases. In Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, pages 434{439. IEEE, 2014.

[5] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos, A Survey of Information-Centric Networking Research. Communications Surveys Tutorials, IEEE, 16(2):, Second 2014.

[6] J Seedorf, M. Arumaithurai, K. K. Ramakrishnan, and N. Blefari Melazzi, "Using icn in disaster scenarios," IRTF, 2015. [Online]. Available: https://datatracker.ietf.org/doc/draft-seedorficn-disaster/

[7] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, and M. Waehlisch, "Icn research challenges," IRTF, 2014, August 2014.

[8] O. Ascigil, V. Sourlas, I. Psaras and G. Pavlou, "Opportunistic off-path content discovery in information-centric networks," *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, Rome, 2016, pp. 1-7. doi: 10.1109/LANMAN.2016.7548860

[9] I. Psaras, S. Reñé, K. V. Katsaros, V. Sourlas, G. Pavlou, N. Bezirgiannidis, S. Diamantopoulos, I. Komnios, and V. Tsaoussidis. 2016. Keyword-based mobile application sharing. In *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture* (MobiArch '16). ACM, New York, NY, USA, 1-6. DOI: http://dx.doi.org/10.1145/2980137.2980141

[10] Waldir Moreira, Manuel de Souza, Paulo Mendes, Susana Sargento, "Study on the Effect of Network Dynamics on Opportunistic Routing", in Proc. of AdhocNow, Belgrade, Serbia, July 2012

[11] Waldir Moreira, Paulo Mendes, Susana Sargento, "Social-aware Opportunistic Routing Protocol based on User's Interactions and Interests", in Proc. of AdhocNets, Barcelona, Spain, October 2013

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

[12] Waldir Moreira, Paulo Mendes, "Social-aware Opportunistic Routing: The new trend", Springer Book on Routing in Opportunistic Networks, ISBN 978-1-4614-3513-6, August 2013

[13] Waldir Moreira, Paulo Mendes, "Dynamics of Social-aware Pervasive Networks" in Proc. of IEEE PERCOM workshop (PerMoby), St. Louis, USA, March 2015

[14] A. Mtibaa, M. May, M. Ammar, and C. Diot, Peoplerank, "Combining social and contact information for opportunistic forwarding" ,in Proceedings of INFOCOM, (San Diego, USA), March, 2010.

[15] I. Psaras, L. Saino, and G. Pavlou. "Revisiting resource pooling: The case for in-network resource sharing." Proceedings of the 13th ACM Workshop on Hot Topics in Networks. ACM, 2014.

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