Objective FP7-ICT-2009-5-257448/D-4.2

Future Networks

Project 257448

“SAIL – Scalable and Adaptable Internet Solutions”

D-4.2
(D.C.2) Architecture and Mechanisms for Connectivity Services

Date of preparation: **2013-02-28**
Start date of Project: **2010-08-01**
Project Coordinator: **Thomas Edwall**
Ericsson AB

Revision: **2.0**
Duration: **2013-02-28**

Document Properties

Document Number:	D-4.2
Document Title:	**(D.C.2) Architecture and Mechanisms for Connectivity Services**
Document Responsible:	Lucian Suciu (FT-Orange)
Document Editor:	Andreas Timm-Giel (UHB), Lucian Suciu (FT-Orange)
Authors:	Ramón Agüero (UC), Pedro A. Aranda (TID), Philippe Bertin (FT), Roksana Boreli (NICTA), Luisa Caeiro (IST-TUL), Rami Cohen (Technion), Reuven Cohen (Technion), Luis M. Correia (IST-TUL), Fariborz Derakhshan (ALUD), Luis Francisco Diez (UC), Lúcio S. Ferreira (IST-TUL), Marta García-Arranz (UC), Guy Grebla (Technion), Heidrun Grob-Lipski (ALUD), Sofiane Hassayoun (IT), Liran Katzir (Technion), Anna Levin (Technion), Carmen López (UC), Marco Marchisio (TI), Ronald Marx (FHG), Olivier Mehani (NICTA), Avi Miron (Technion), Gabi Nakibly (Technion), Susana Perez (Tecnalia), Danny Raz (Technion), Horst Rößler (ALUD), Simone Ruffino (TI), Peter Schefczik (ALUD), Peter Schoo (FHG), Michael Soellner (ALUD), Golam Sarwar (NICTA), Lucian Suciu (FT-Orange), Andreas Timm-Giel (UHB), Asanga Udugama (UHB), Iñigo Urteaga (Tecnalia), Yasir Zaki (UHB), Xi Li (UHB)
Target Dissemination Level:	PU
Status of the Document:	Second edition
Version:	2.0

Production Properties:

Reviewers:	Luis M. Correia (IST-TUL), Fabian Schneider (NEC), Benoit Tremblay (Ericsson)

Document History:

Revision	Date	Issued by	Description
1.0	2012-09-30	Lucian Suciu	Final Version
2.0	2013-02-28	Lucian Suciu	Revised version to expand architecture and orchestration concepts

Disclaimer:

This document has been produced in the context of the SAIL Project. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2010–2013) under grant agreement n° 257448.

All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

Abstract:

The Deliverable D.C.2 from the SAIL project reports on the work and the results from the Open Connectivity Services (OConS), tackling various networking issues through the OConS mechanisms developed and assessed. The document also presents a comprehensive OConS framework, further specifying the architectural concepts such as: service orchestration, information model, related interfaces, and necessary procedures. We have also applied the OConS approach to the overall SAIL flash crowd scenario, detailing two main use-cases, namely OConS supporting CloNe, and, respectively, OConS supporting NetInf.

Keywords:

open connectivity services, orchestration, cloud networking, network of information

Contents

1 Introduction

Future networks need to satisfy new and dynamic service IP based Internet was not designed for. Numerous mechanisms have been developed to enhance and supplement existing Internet protocols, but there is lack of coordination between mechanisms. In this deliverable, the Open Connectivity Services (OConS) approach, architecture, and mechanisms are presented. The main idea behind OConS is that through analysis of connectivity mechanisms, identifying their common control functionalities and interconnecting those parts with open protocols and interfaces, the way will be paved for orchestrating new, more effective combinations of these mechanisms (known as OConS services), and easily designing new mechanisms reusing common elements and interfaces.

Based on these ideas, the OConS approach is to propose and define a comprehensive architecture for the orchestration of open connectivity services. The orchestration functionality serves an explicit connectivity request by a user, which can be e.g. an application, NetInf or CloNe, or an implicit request triggered based on the monitored network state. Then, to address the connectivity requirements, the Orchestration identifies the most appropriate OConS mechanisms, from the set of those available and, finally, it instantiates them.

This approach is developed within the Scalable and Adaptive Internet Solutions (SAIL) project to an extent that allows at least the assessment within the considered use case scenarios, but it can be applicable in other general cases, too. This deliverable extends the basic architectural ideas from the deliverable D.C.1 [1] and its addendum [2].

We first give the motivations, in Chapter 2, to better explain the background behind OConS design. Then, in Chapter 3, a description of OConS architecture will be provided, covering its functional entities, the orchestration process and the related interfaces. Particular emphasis will be put on the orchestration logic, which can automatically combine multiple mechanisms, according to the network state, operators' rules and policies, and users and applications' needs.

In Chapter 4, we give a comprehensive discussion of the underlying OConS mechanisms, we present the extensive research work carried out on each and every OConS mechanism, and we summarise their performances and benefits; likewise, the Annex A contains more details about the OConS mechanisms and the results obtained so far.

Then, we give further evidence that the developed concepts are indeed useful in a context where innovative and demanding requirements are expected, by applying this framework to the overall SAIL flash crowd scenario throughout two main use-cases: **OConS supporting Cloud Networking (CloNe)** in "Data-Centre Interconnection and Seamless Access for Mobile Users" in Chapter 5, and, respectively, **OConS supporting Network of Information (NetInf)** in "Wireless and Multi-P Support for Information Centric Networks" in Chapter 6 (where 'Multi-P' stands for multi-point, multi-path and multi-protocol).

Finally, in Chapter 7, we preliminarily discuss and assess the OConS results, i.e., the added-value of the Orchestration in relation to the SAIL scenario and use-cases, as well as the specific contributions of the OConS mechanisms. This will be continued in depth in the forthcoming deliverable D.C.4 [3], dedicated to the overall validation of the OConS approach and OConS mechanisms.

In addition, the plans for implementation and prototyping, as well as the experimentation details and findings are reported in the companion deliverables D.C.3 [4], and, respectively D.C.5 [5].

Last but not least, we outline in the corresponding Annexes the comprehensive scope of OConS research work on specific OConS mechanisms, protocols and algorithms, and we provide additional information on the OConS interface specifications, information model and security aspects.

2 Motivation and Approach

From a very high-level perspective, all of today's networks, whether fixed and mobile, international or local, evolved from many years of adaptations and improvements around a few basic and very successful technologies: TCP/IP, Ethernet 802.11, or the cellular 3GPP are good examples for such successful technologies. During the years, new mechanisms have been designed as "add-ons" or overlay to those technologies, which were experimented, engineered, standardized and finally deployed inside the network nodes or on the user equipment, as appropriate. Many of these mechanisms were designed to work for very specific use cases, in a specific network domain, typically exploiting cross-layer functionality for improved performance. The features they provide are usually confined to those environments, and it is seldom possible to launch them from an external entity. Mechanisms are isolated and unaware about each other, and there is no upper-layer entity to coordinate their performance. As an example, congestion control and mobility management mechanisms usually do not exchange any information that could harmonize their logic and take advantage of each other's knowledge of the state of the managed resources, users and nodes. Moreover, the scientific and technology community is continually proposing new improved mechanisms, protocols and algorithms, designed again for specialized use cases, which can potentially optimize network behaviour (and eventually improve the user's Quality of Experience (QoE)): several innovative connectivity and control mechanisms have also been designed in the context of SAIL and will be detailed later in this deliverable. But, the threshold to adopt new concepts and mechanisms in production networks is still high, due to the inability of the systems to easily integrate new mechanisms in the large installed base of communication infrastructure.

On the other hand, there are upcoming connectivity requirements that go beyond the capabilities of currently available networking technologies, with their pre-defined services. These connectivity requirements can come from information-centric networks, cloud networking or social-community driven flash crowds scenario.

The main problems OConS tackles can be detailed as follows:

- There is a lack of coordination among mechanisms. Mechanisms usually work "in isolation" from other mechanisms that are working at same or different levels (link, flow, or network levels). This lack of coordination is both horizontal (i.e. within a level) and vertical (i.e. across levels). As such, they don't take any advantage from exchanging relevant information across the mechanisms, regardless whether they work on same or different network segments (e.g. link, node, network, service infrastructure). This lack of coordination results in redundant spread of information state and sub-optimal operating points.
- The network control lacks flexibility and expressiveness. This issue is tightly coupled with the previous one. Policies and rules that control network nodes and services are usually specific to one application domain and to one specific mechanism (e.g. policies for the mobile core network, policies for firewalls, policies for fixed networks, forwarding rules for switches). The specified policies usually affect only one segment of the network, which clearly has the advantage to simplify the decision process in these nodes, at the expense of expressiveness. This is major pain spot, due to the need to differentiate the treatment of traffic belonging to different users, which might have very different profiles. If the policies are not sufficiently expressive and powerful, they can only be enforced on one network segment, where it could be necessary to have a control over many level and nodes across the network.

OConS aims at solving the problems highlighted above, by designing an architecture that manages and provides connectivity services in a coordinated and consistent manner, facilitating easy integration of new and enhanced mechanisms, which can be applicable to multiple and different network segments. The OConS approach does this by:

- Abstracting the mechanisms' common functionalities in terms of sensing/monitoring, decision making and actuating/executing and by providing an easy way to model them by means of elementary entities;
- Providing the means to register the new mechanisms, and detect their availability within the network;
- Providing the orchestration functionalities that coordinate, compose and integrate the mechanisms;
- Designing open interfaces to interconnect the various mechanisms (i.e., a northbound SAP, as well as southbound and east-west interfaces);
- Providing a comprehensive information model to capture all the necessary concepts;
- Providing the means to create more expressive policies to be enforced in the network; OConS also facilitates the consolidation of multiple independent policies into one set of policies, governing all mechanisms;
- Controlling the e2e path: not only network nodes, but also end nodes; thus, in OConS, many mechanisms are executing on the end nodes and the orchestration process controls those mechanisms;
- Providing security not weaker than with the existing architectures (and aim at even better security, if that is achievable at economically viable costs).

Solving these important problems, OConS will bring several added benefits:

- Ability to open-up the networking mechanisms(i.e., flexibility) and to be able to integrate more of them together in combination, almost in a "plug-in" manner;
- More efficient use of network resources and improved performance, leveraging the coordination and information sharing among different mechanisms;
- Ability to deploy new services, made by the integration of many mechanisms, when and where is needed, especially in the network access part (i.e., scalability and possibly lower cost);
- Employment of new service will be easily repeatable (if the integration is done through a more "automatic" and "programmatic" manner);
- Ease the sharing of various procedures (e.g., monitoring can be shared by various mechanisms);
- More easily to promote the innovative mechanisms in the network, thus bringing connectivity improvement to current Internet.

Having in mind the continuous integration of the control mechanisms and functions in the networking ecosystem, an emerging approach nowadays is the Software Defined Networking (SDN) (e.g., see [6] and [7]); OpenFlow is the most prominent example of SDN implementation. We now provide another view of OConS architecture, highlighting the differences between OConS and SDN.

In the SDN/OpenFlow world, one or more logically centralised controllers manage the data-plane switches, enforcing policies on them (i.e., using flow entries). The user/application/operator requirements are integrated in the controller or in the applications/control-functions residing on top, yet they are expressed in a non-transparent manner (these requirements are thus implicit in the flow rules). A large part of the decisions are made within the controller/applications level and then only the forwarding rules are pushed down in the forwarding nodes. OpenFlow controllers usually do not control end-user nodes. In summary, SDN functionality can be expressed as follow:

SDN/OpenFlow = IF (packet/flow/tunnel info, network state/context) THEN (apply Executions in pipeline)

The OConS concept is similar, but also going beyond SDN: It is similar in a sense that one mechanism receives information from the network-side and end-terminal nodes, takes decision using this information and then applies policies back to the data-plane execution entities. However, OConS significantly extends this concept, by enabling the orchestration of several mechanisms. A SAP interface is defined from each mechanism towards the Orchestration entity (North-South interface), along with East-West interfaces between the mechanisms. Moreover, OConS provide a way to define policies that are more expressive than simple flow entries, taking into account user, application and operator requirements in a more open approach. The enforcement of OConS decisions and policies result in a "chain" of execution parts of these different mechanisms. Finally, OConS provide enhancements that goes far beyond packet forwarding, also integrating link, flow and network enhancements, and involving any OConS node in the network (and not just the switch or the router). In summary, OConS functionality can be expressed as follow:

OConS = IF (packet/flow/tunnel info, network state/context, user/app/operator requirements) THEN (apply a chain of Decisions/Control-Functions + Execution chain)

As a final note, OConS does not intend to have a clean slate approach but is rather building on and enhancing the current Internet. Thus, to improve quality of experience, usage of the networking resources, and to enable new network services, we have sought more interactions between the constituent link, network and flow mechanisms. We are not removing the boundaries between those types of OConS mechanisms for modularity reasons, but at the same time facilitating the benefits of cross-layer functionality in a structured manner. Furthermore, to fully benefit from these new and/or enhanced mechanisms, we have specified the OConS architecture which comes notably with the necessary management and orchestration functionalities to better interact and use those mechanisms.

3 OConS Architecture

In this chapter we will provide a detailed description of the OConS architecture. Capitalizing on the work and the notions introduced in [1, 2], we have designed a comprehensive functional architecture for OConS, using two different conceptual levels.

- **Architectural framework**: starting with the basic building blocks, we have used the OConS elementary functional entities to build our mechanisms, i.e., we have designed every new OConS mechanism according to this architectural framework, as presented in detail in [1, 2]. In the following section 3.2, we will only provide a brief summary of these basic architectural concepts.
- **OConS Functional Architecture**: these OConS mechanisms have to be orchestrated (i.e., discovered, configured, deployed) to build and provide the OConS services. Accordingly, in section 3.3, we propose the corresponding OConS Functional Architecture and we introduce the core Orchestration functionalities (which are further detailed in 3.7).

While the architectural framework introduces specific modelling concepts to be applied [8], an OConS Functional Architecture is a functional system architecture that results when mechanisms are orchestrated according to that framework; thus, even if general statements are made sometimes about the OConS Functional Architectures, they should get clearer when considering the whole context.

Likewise, in order to bring OConS into real networks, we describe in 3.4 the two scopes where the OConS orchestration takes place: **Intra- and Inter- OConS Node views**; in the Inter-Node view, the concept of OConS domain is also introduced.

3.1 Overview

A brief overview of the OConS architecture is presented in this section, with its key components and functionalities detailed in the further sections.

Generally speaking, OConS is a control framework that provides the capability to orchestrate a set of connectivity services, running on one or more interconnected nodes. An OConS connectivity service is formed by a specific combination of OConS connectivity mechanisms. In order to make the design of new mechanisms easy, to be able to "compose" them together and to share and reuse their functionalities, OConS mechanisms are modelled following a mechanism-level architecture that decomposes them into information monitoring (i.e., IE), decision making (i.e., DE), and execution and enforcement (i.e., EE) functional entities.

A representation of the OConS functional architecture is presented in Figure 3.1. At the centre of the OConS functional architecture sits the Service Orchestration Process (SOP), capable of orchestrating an OConS service composed of one or several OConS mechanisms. OConS users (i.e., generic applications or CloNe/NetInf applications) communicate with the SOP by means of the Orchestration Service Access Point (OSAP). Through OSAP, users communicate their requests regarding the desired connectivity services, to be set-up by SOP, and receive notifications about the status of the requested connectivity services. In order to store the data of the various mechanisms, rules and policies, as well as the network state, the SOP is connected to a database, named the Orchestration Registry (OR). The Intra-/Inter- Node Communication (INC) functionality takes care of exchanging the messages among the architecture components, both locally and remotely.

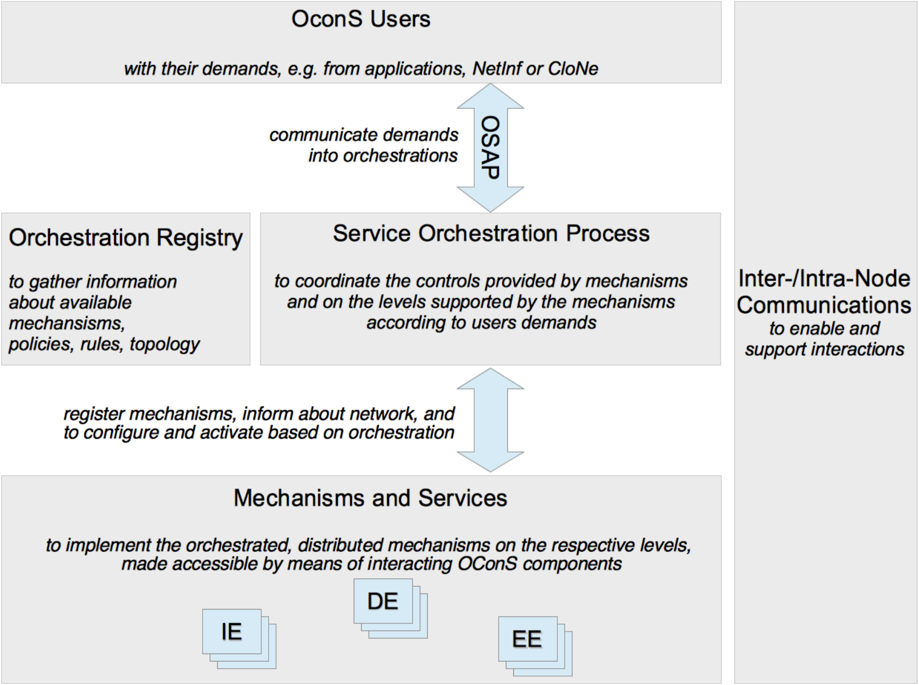

OconS Users
with their demands, e.g. from applications, NetInf or CloNe
communicate demands into orchestrations
OSAP
Orchestration Registry
to gather information about available mechansisms, policies, rules, topology
Service Orchestration Process
to coordinate the controls provided by mechanisms and on the levels supported by the mechanisms according to users demands
Inter-/Intra-Node Communications
to enable and support interactions
register mechanisms, inform about network, and to configure and activate based on orchestration
Mechanisms and Services
to implement the orchestrated, distributed mechanisms on the respective levels, made accessible by means of interacting OConS components
DE
IE
EE

Figure 3.1: OConS Functional Architecture overview.

An OConS node must have an INC, and may have a SOP and/or OR. Only nodes with a SOP may orchestrate an OConS service. Depending on the mechanism's specificity, the orchestration of an OConS service may span a single link, a group of links and nodes (network level), or affecting the complete end-to-end flow. The INC supports this vertical, i.e., within an OConS node, and horizontal (i.e., between nodes) OConS orchestration and control.

Figure 3.2 provides an example of orchestration of two OConS services in a generic network environment. OConS service A is provided to multiple users within a Wireless Challenged Network (WCN), which are all streaming the same music. The service is the composition of specific OConS connectivity mechanisms for the WCN, to guarantee the best connectivity. OConS service B is an answer to a connectivity request of a user to access a file storage in the cloud. The result of his request is the orchestration of an access selection mechanism combined with a multipath routing mechanism, that provides the user with the optimal access network and, at the same time, with a multi-path connection in the backbone to the file storage, which provides better performance and reliability.

It is worth noting that different types of OConS nodes are represented in Figure 3.2. Thus, any OConS-capable node has the basic capability to communicate with other peer OConS nodes via the INC functionality. In addition, nodes with a SOP are capable to orchestrate an OConS connectivity service within a set of OConS nodes. Certain nodes have an OR that stores information about the available components (mechanisms and services), then the OR is accessed either locally or remotely by a SOP orchestrating the services for a given set of nodes. Other nodes with only an INC are remotely orchestrated by a SOP, and, thus, these nodes are only hosting and launching some OConS mechanisms. The orchestration can be done either in a fully-distributed (e.g., OConS service A in the wireless challenged network) or in a domain-centralized manner (e.g., OConS service B in the wireless heterogeneous access network).

OConS is designed to coexist with the current Internet. In fact, as shown in the above example, non-OConS nodes (i.e., those not upgraded with OConS-related software, intermediate or even end-nodes), although not involved in the orchestration of an OConS service, can carry the data-stream

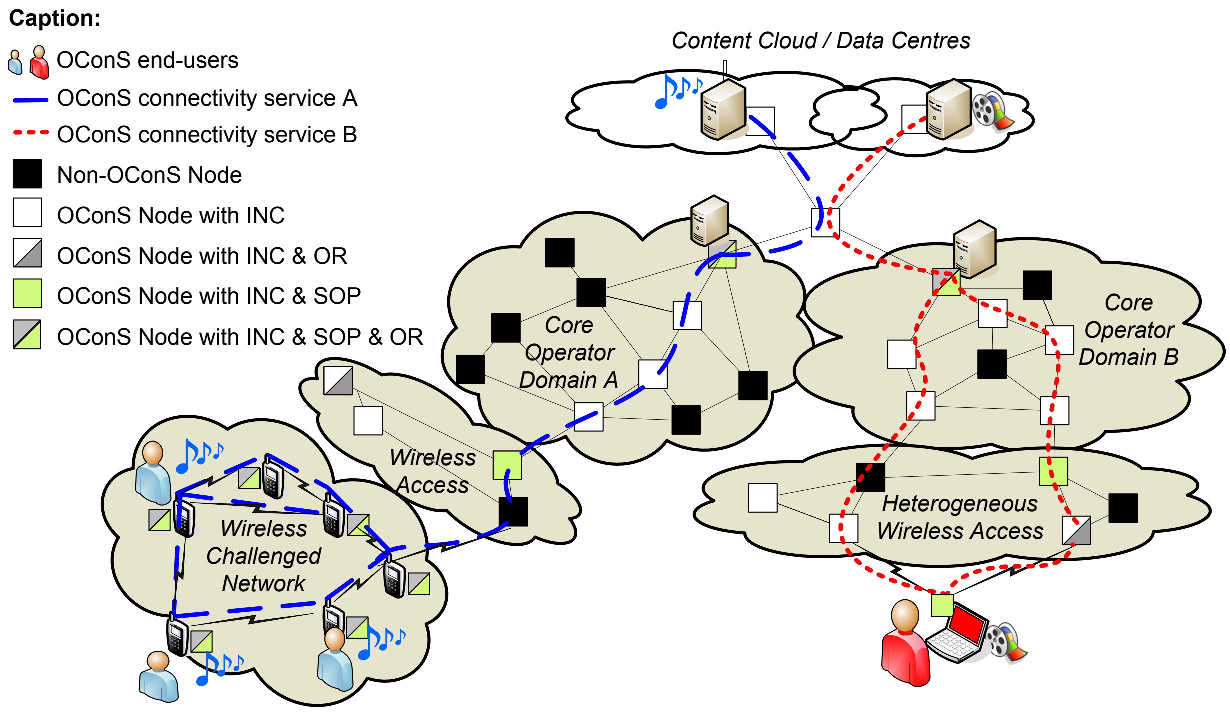

Caption:
OConS end-users
OConS connectivity service A
OConS connectivity service B
Non-OConS Node
OConS Node with INC
OConS Node with INC & OR
OConS Node with INC & SOP
OConS Node with INC & SOP & OR
Content Cloud / Data Centres
Core Operator Domain A
Core Operator Domain B
Wireless Access
Wireless Challenged Network
Heterogeneous Wireless Access

Figure 3.2: Generic network environment with two orchestrated OConS services.

which is controlled by an OConS service.

3.2 Architectural Concepts

In this section, the basic concepts of OConS, as introduced in past deliverables [1, 2], are recalled.

3.2.1 Elementary Functional Entities

As motivated in Chapter 2, a large number of "mechanisms" have been proposed and adopted in today's networks, to provide specific features or to overcome specific deficiencies. For example, Mobile IPv6 (see [9]) is a mobility management mechanism that enables a mobile node to maintain its IP address when it is moving across different IP access subnets. Or for instance, the 3GPP Policy and Charging Control (see [10]) is a mechanism that allows a mobile operator to define and "push" user-based QoS policies to the mobile core network gateway, e.g. when he attaches to the network for the first time.

To be able to design and combine novel mechanisms together, they must be modelled according to a standard framework, that can capture the basic elements of each mechanism and represent apparently different mechanisms in a uniform way. Analysing some existing and newly proposed mechanisms, it can be noted that some of their functionalities are similar or duplicated, e.g., monitoring link states or buffer sizes. Hence, we propose to "decompose" a mechanism in smaller and more manageable entities. Using basic functional bricks facilitates sharing, reusing and easily designing of the networking functions: for example, when collecting information, predicting network states or taking the needed control and management decisions.

Based on this motivation, we defined by means of the OConS architectural framework three generic Functional Entities (detailed in [1, 2]), that abstract the monitoring, decision making and enforcement components of any mechanism:

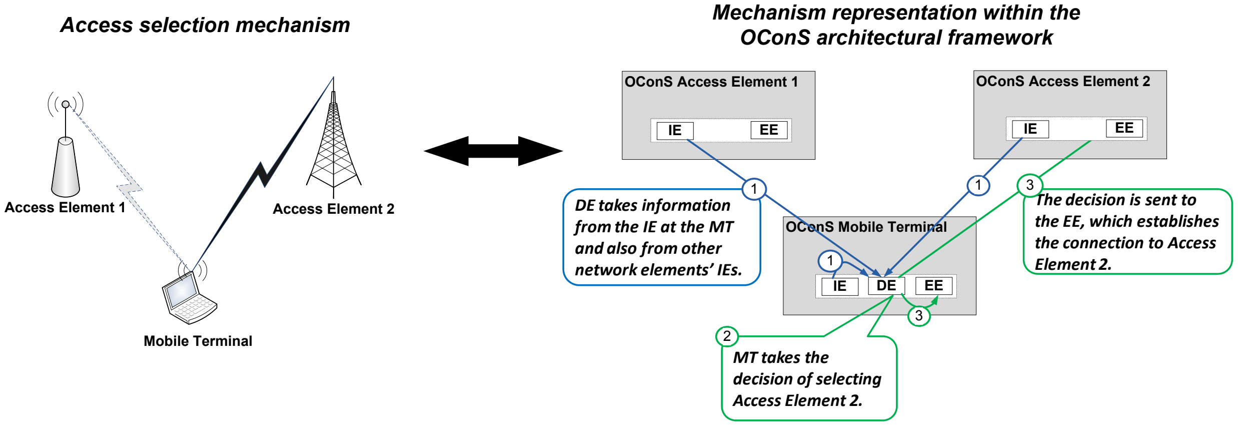

Access selection mechanism
Access Element 1
Access Element 2
Mobile Terminal
Mechanism representation within the OConS architectural framework
OConS Access Element 1
IE
EE
OConS Access Element 2
IE
EE
OConS Mobile Terminal
IE
DE
EE
1
2
3
DE takes information from the IE at the MT and also from other network elements' IEs.
The decision is sent to the EE, which establishes the connection to Access Element 2.
MT takes the decision of selecting Access Element 2.

Figure 3.3: Example of network access selection mechanism modelled with the functional entities of the architectural framework.

- Information Management Entity - IE, responsible for information gathering and monitoring; the gathered information can be processed (e.g., aggregated or filtered) before being provided to decision making entities requesting it or being subscribed to it.
- Decision Making Entity - DE, where the decision algorithms are implemented; a Decision Making Entity (DE) uses information from Information Management Entitys (IEs) or the output of other DEs as an input to take a decision; the decision taking can be located in one node (centralized) or in multiple ones (distributed), where several DEs interact to make a decision.
- Execution and Enforcement Entity - EE, which implements the decisions.

These entities are processes (pieces of software) that can be distributed, instantiated, and executed within the OConS-capable nodes; they have well-defined IDs, thus enabling their basic configuration and handling using well-defined messages. We will describe the interfaces that enable communication between these entities in Section 3.5: in summary, interfaces are used to collect "information" from the IEs and to enable the DE to communicate decisions (or, policies) to the Execution and Enforcement Entity (EE), which will then enforce them.

An illustrative example of how a *simple* access selection mechanism can be modelled with the OConS functional entities, is presented in Figure 3.3. A Mobile Terminal (MT) is equipped with various interfaces able to access different Radio Access Technologies (RATs). It collects information available at the MT's IE and from the various access elements RAT IEs. Based on this information and user connectivity requirements possibly available in an IE, the DE on the MT takes the decision for one or several Access Elements. This is communicated to the MT's EE, which executes the required actions so as to initiate the flow through the selected RATs.

3.2.2 OConS Mechanisms and Mechanism Manifest

The three entities defined above can be used to easily model legacy mechanisms, but also to design new ones from-scratch. It can be stated that an OConS mechanism is a process made of at least one DE, and one or multiple involved IEs and EEs (please see [1, 2]). Every OConS mechanism can be defined using them; the abstractions of the Functional Entities are independent from any layer or protocol. By adopting this design methodology, this new type of mechanisms can be natively orchestrated with other similar mechanisms.

In OConS a large variety of mechanisms has been initially proposed in the past deliverables [1, 2] and in this document we will further detail their logic in Chapter 4. However, in order to simplify the orchestration process, we also proposed a categorization of the OConS mechanisms, which can

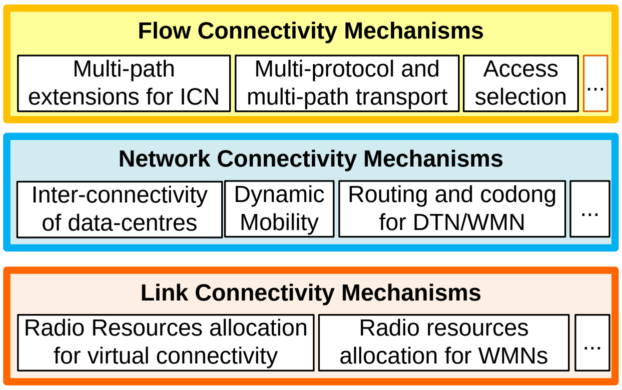

Flow Connectivity Mechanisms
Multi-path extensions for ICN
Multi-protocol and multi-path transport
Access selection
...
Network Connectivity Mechanisms
Inter-connectivity of data-centres
Dynamic Mobility
Routing and codong for DTN/WMN
...
Link Connectivity Mechanisms
Radio Resources allocation for virtual connectivity
Radio resources allocation for WMNs
...

Figure 3.4: OConS mechanisms levels, with some examples of OConS mechanisms.

be divided in three main functional groups, depending on whether they work at flow, network or link level:

- **Flow Connectivity Mechanisms**: they are flow and session specific, either end-to-end or edge-to-edge; they are composed of routing and transport mechanisms, and they depend on the corresponding application.
- **Network Connectivity Mechanisms**: they are node specific services composed of routing and transport mechanisms, which are in this case independent from the current application; they involve two or more nodes, spanning over one or several hops (e.g., end-, access- or core-nodes).
- **Link Connectivity Mechanisms**: they are specific to particular links, not spanning more than one hop; thus, they are typically implemented in the physical or data-link layers.

Figure 3.4 illustrates the OConS mechanisms levels, with some examples of OConS mechanisms for each level. An overview of such, newly developed, OConS mechanisms is provided in 4.1, as well as in Annex A.

Moreover, a common way is needed, to represent each mechanism characteristics, in terms of what it provides, where it is running, what are the needed resources, constraints and requirements. Hence, OConS defines the **Mechanism Manifest**, a data structure, represented in Table 3.1, which contains all relevant and necessary information on a specific mechanism[1]. The table has three columns: "Field" contains the names of relevant characteristics that define OConS mechanisms, "M/O" contains an indication if that field is Mandatory or Optional and "Values" contains a list of permitted values for each field.

Some fields are worth a more in-depth description:

- **Mechanism Target** contains an indication of the object which a given mechanism acts on, and the type of the action performed: e.g. selection of next hop, configuration of transport parameters, selection of interface etc.
- **Metrics improved** describes which metric is primarily affected by the mechanism
- **Mechanism Constraints** describes under which constraints the mechanism can run (the type of network required, type of application allowed etc.). Moreover, the "trigger" sub-field contains the metric ("based_on_metric") and associated thresholds used to trigger the mechanism.
- **Required DEs** contains the FQDN name of the main DE of the mechanism. This field is mandatory, since any mechanism is associated with at least one DE, which the SOP will contact.

It is worth noting, the Manifest contains only the capabilities of each mechanism and not real-

[1] For a definition of the "names" used in the Manifest Table, see D.4

time state of the entities of the same mechanism. Real-time updates come from entities, during the life of a mechanism instance.

Table 3.1: Mechanism Manifest

Field	M/O	Values
Mechanism ID	M	– Primary Key (Integer)
Security certificate	M	e.g. X.509 / RFC 5280
Mechanism Name	M	– FQDN name of the mechanism
Service Level	O	– flow – network – link
Mechanism Target	O	– selection – path – next hop (aka access) – interface (local on a node) – channel – configuration – air interface parameters – transport parameters – access network parameters – core network parameters – network coding
Scope	O	– node(local) – immediate neighbour – domain – application
Metrics Improved	O	– packet delay – packet delay variation – throughput – resource utilization – handover delay – reachability – CPU load – packet loss – application_specific_metric

Table 3.1: Mechanism Manifest

Field	M/O	Values
Mechanism Constraints	O	– network technology [DTN, Wi-Fi, LTE, EPC, etc.] – applications [VoIP, VM_migration, Video_streaming, etc.] – trigger = (based on metrics, starting_on_threshold, stopping_on_threshold) – needed runtime resources (CPU, memory etc.)
Required DEs	M	– Main DE name – Other optionally needed DEs names
Required IEs	O	– Needed IEs names
Required EEs	O	– Needed EEs names

3.2.3 OConS Services

There are several possible relations among those different mechanisms: some of them are complementary, others can be combined, while others are conflicting. Nonetheless, we reckon that we can achieve potential synergies by combining existing mechanisms in more powerful and optimized solutions. Hence, the OConS mechanisms can be used as a standalone module or in combination with other mechanisms to form an OConS Service, which in turn is offered to the OConS users.

It is the OConS Orchestration functionality which takes care of the selection, combination, and instantiation of the OConS mechanisms, according to:

1. the Mechanism Manifest (Table 3.1), which is used in the initial phase of the Orchestration process, to register the mechanisms capabilities;

2. certain consistency rules and composition policies, as detailed in Sec. 3.7.4

The creation of an OConS service is detailed in sec. 3.7, which describes the whole Orchestration process.

It is safe to say that any mechanism (legacy or future) can be defined as an OConS mechanism, as long as it is modelled by the functional entities, it is using the OConS interfaces and exposes a well-defined Manifest. Thus, OConS approach facilitates the implementation, the instantiation and the launch of the mechanisms, as well as the provision of the appropriate OConS Services.

3.2.4 OConS Nodes and Domains

The OConS nodes can be defined as networking elements that host the OConS functional entities. Therefore, the OConS node is either an infrastructure node (e.g. base-station, router, switch) or an end-user terminal, providing computing, storage and networking resources (including virtualized resources) to the OConS entities. It is the place where the OConS entities are instantiated, and

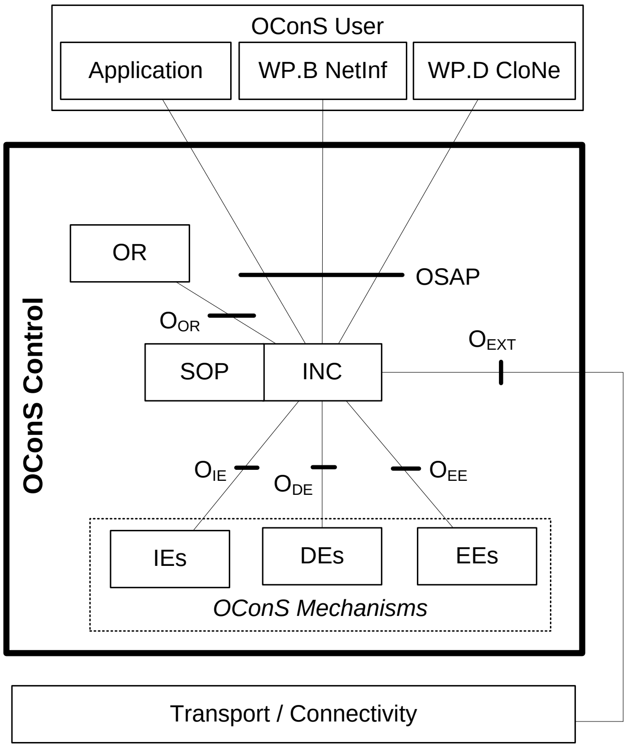


Figure 3.5: OConS Functional Architecture.

executed, enabling the launch and the usage of the OConS mechanisms (and services). It is worth noting that an OConS mechanism can make use of multiple functional entities, residing on multiple OConS nodes, as in the example of access selection above. An OConS Node can be a new node, or an existing one upgraded with OConS-related software. Finally, without loosing generality, OConS nodes have also minimal local orchestration functionalities (as described in the next section) to be able to orchestrate their local mechanisms, as a minimum requirement.

The OConS domain comprises a set of nodes (and links among them) which provide connectivity services to the applications and users by implementing a given set of OConS mechanisms (e.g., the support of Distributed Mobility Management, Multi-Path/Protocol, combination of these, and so on).

Moreover, an OConS domain can span several administrative domains (i.e., trust/management). However, each OConS node is associated with one administrative domain, as the basic notion for ownership and the power to control security. Likewise, basic inter-domain interworking (e.g., routing) may re-use exiting mechanisms (such as Border Gateway Protocol (BGP)), whereas the complete specification of the inter-domain services orchestration (i.e., inter-domain SOP) is for further study.

3.3 OConS Functional Architecture

Figure 3.1 provided a high level overview of the OConS functional architecture. Based on the previously introduced architectural concepts, it is further detailed in this section.

Consequently, Figure 3.5 represents the OConS functional architecture as a reference model. All represented architectural components and associated interfaces are described below.

The **IEs**, **DEs**, **EEs** functional entities abstract/decompose the monitoring/information gathering, decision making and enforcement components available in OConS nodes. They ease the building of OConS mechanisms, specified by DEs.

The **Service Orchestration Process (SOP)** is the central element of the OConS architecture. The SOP is responsible for the discovery and validation of OConS mechanisms in a node and/or OConS domain. As an answer to a connectivity request, it is able to instantiate and orchestrate OConS Services composed by adequate OConS mechanisms, which are appropriately configured and launched in a node or set of nodes. For it, a set of rules is used by the SOP for mapping:

- OConS user's connectivity request expressed by a demand profile into connectivity requirements.
- Connectivity requirements and network state into candidate mechanisms.
- Candidate mechanisms into a composed service.

In this sense, the SOP provides the functionalities which are enumerated below:

- Bootstrap, where available OConS entities and mechanisms are discovered, and default OConS services are launched.
- Launch or reconfigure an OConS service, composed of adequate OConS mechanisms, as a response to an OConS user connectivity request or to a change in the network state.
- Monitor launched OConS Services.

When orchestrating an OConS service, depending on the level of the activated mechanisms, the SOP's orchestration may span a single link (link level), a group of links and nodes (network level), or affecting the complete end-to-end flow (flow level).

The **Orchestration Registry (OR)** is where data on the available OConS entities and mechanisms, as well as on the created OConS services is registered. It is used by the SOP to become aware of the existence of entities and mechanisms. The OR contains three types of registers:

- OConS Entities Registry: during the bootstrapping of an OConS node, all available OConS Entities hosted on that node are registered in the OR. This can be done following a discovery procedure, or initiated by each OConS entity. For each entity, the collected information is the OConS ID, the type of entity (IE, EE or DE) and its capabilities. In the case of an IE, the capabilities are the collected parameters; for an EE, where it actuates; for a DE, what it decides. In particular, for each DE controlling an OConS mechanism, dependencies with respect to needed IEs and EEs are also described. There might be mechanisms with multiple DEs; in this case, only one of them is registered at the OR, for the entire mechanism. Besides this, it can also contain collected data on the network state and topology.
- OConS Mechanisms Registry: in the mechanism creation process, each mechanism is registered in the OR with an unique ID, and with the IDs of the associated IEs, DEs and EEs. The core DE of the mechanism also specifies how the mechanism should be built (e.g., needed IEs/EEs) and under which conditions it operates correctly or optimally; the mechanisms capabilities are expressed according to the common information model.
- OConS Services Registry: created OConS services are registered using their ID, and the IDs of associated active mechanisms. Other information can be stored, like the connectivity requirements, or the lifetime of the service.

The **Intra/Inter- Node Communication (INC)** supports the local and remote communication between the various architectural components. It is in charge of receiving OConS messages from local or remote components and route them to their destination (either local in the OConS node, or remote in another node) and vice versa. The INC chooses whatever transport technology is deemed relevant for delivering the messages: it can be Inter-Process Communication (IPC) if the destination is internal to the same OConS node, or it can be an underlying transport communication, if the destination is remote. It is assumed that connectivity between nodes is available. It means that nodes are reachable using existing forwarding schemes to deliver messages(e.g., IP connectivity between two OConS nodes, or "one hop connectivity" in a broadcast medium like Ethernet). OConS entities are agnostic of this communication method in use to carry their messages.

The INC is in charge of resolving IDs into the relevant lower layer locator and the subsequent encapsulation and routing towards it. Summarizing, the INC acts on behalf of the entities to relay these messages towards their destination.

The functional architecture described above can be contained within an OConS node, or distributed amongst a group of OConS nodes within the OConS domain, where each node may contain a subset of these components. Still, as minimal components, an OConS node must have at least an INC, besides the existing functional entities. Within an OConS domain at least one SOP, and one OR, must exist. If no OR is available in an OConS node, registration of available entities and mechanisms is done remotely, i.e., supported by the INC. If there is no SOP in an OConS node, this means that it cannot launch by himself the orchestration of a OConS service; nevertheless, a SOP residing on another OConS node may remotely orchestrate the OConS services in this node. Similarly, at the OConS mechanisms level, the functional entities that build a given mechanism can be all located within a single OConS node, or spread among several ones.

All above components communicate using the following **logical interfaces**:

- O_{IE}, O_{DE}, O_{EE}: interface to manage the functional entities. It enables the advertise and discovery of entities, as well as their registration and configuration.
- $OSAP$: Orchestration Service Access Point, it is an external interface with OConS users (application/CloNe/NetInf), it can be thus seen as an Application Programming Interface (API) used to communicate user connectivity requirements to OConS through a demand profile, and also by the OConS system to communicate to the user the status of the requested OConS service (ready, error code).
- O_{OR}: interface to communicate with the OR. It enables to register entities, publish registered mechanisms, validate mechanisms, store network states, etc.
- O_{EXT}: interface to communicate with a remote OConS node's INC, over any-packet-based system able to encapsulate messages and carry them to other OConS nodes. Examples include UDP over Ethernet or 802.11.

The basic communication mode between OConS entities is in the form of requests and responses. All the necessary messages are presented in 3.5.

Likewise, all the OConS concepts from the functional architecture are captured in a comprehensive Information Model that defines these concepts, their relationships, the semantics of information, and the information processing in the OConS system (see Section 3.6 and also Annex B). However, a fully-fledge specification of the corresponding Data Model (which addresses the actual structures and the exact encoding of data, i.e., depending on the implementation and/or the corresponding programming language) is out of the scope of our work, as we only provide some implementation examples and proof-of-concepts.

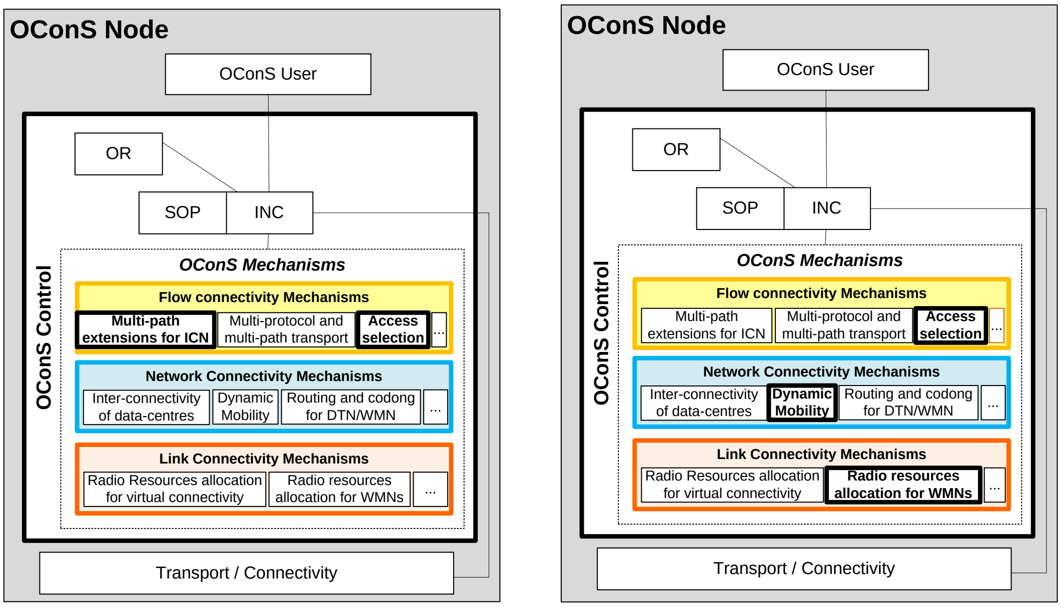
3.4 Intra-Node and Inter-Node Views

In this section, we describe the two main scopes where the OConS orchestration takes place: Intra-Node and Inter-Node.

3.4.1 Intra-Node View

The first scope is the OConS node *per-se*, i.e. within a single end-user terminal, or within an infrastructure node. Multiple OConS mechanisms may be present within an OConS node (e.g. multi-path, multi-homing, etc.), possibly working at different levels, as described in 3.2.2.

Here, the orchestration happens among the mechanisms available on the node and can embrace mechanisms working at the same or different levels, as depicted in Figure 3.6, where two OConS


OConS Node
OConS User
OR
SOP
INC
OConS Control
OConS Mechanisms
Flow connectivity Mechanisms
Multi-path extensions for ICN
Multi-protocol and multi-path transport
Access selection
...
Network Connectivity Mechanisms
Inter-connectivity of data-centres
Dynamic Mobility
Routing and codong for DTN/WMN
...
Link Connectivity Mechanisms
Radio Resources allocation for virtual connectivity
Radio resources allocation for WMNs
...
Transport / Connectivity
OConS Node
OConS User
OR
SOP
INC
OConS Control
OConS Mechanisms
Flow connectivity Mechanisms
Multi-path extensions for ICN
Multi-protocol and multi-path transport
Access selection
...
Network Connectivity Mechanisms
Inter-connectivity of data-centres
Dynamic Mobility
Routing and codong for DTN/WMN
...
Link Connectivity Mechanisms
Radio Resources allocation for virtual connectivity
Radio resources allocation for WMNs
...
Transport / Connectivity

(a) Orchestration within the same level.

(b) Orchestration within multiple levels.

Figure 3.6: OConS Intra-node view examples of orchestration of an OConS service.

nodes are depicted and the link, network and flow levels are represented, highlighting some example OConS mechanisms for each of them:

- In the first case, Figure 3.6a, the orchestration process only recognizes mechanisms at one level, independently of mechanisms running on other levels. Practically, it means that the SOP interacts with the entities of the mechanisms belonging to the same level. In this level-specific orchestration, the number of possible interactions between the applicable OConS mechanisms is limited, and a more homogeneous set of policies, rules and actions can be used.
- In the second case, Figure 3.6b, the orchestration happens at different levels inside an OConS node. Here, the SOP is in charge of enabling the communication among mechanisms working on different levels, to exchange events and requests, or to jointly react on various changes: communication state changes at link layer (e.g. if a link resource gets overloaded), topology changes at the network level (e.g. new IP address, new routes, etc.), or user and operator input (such as user profile, or operator expressed SLAs and policies).

An illustrative example of an intra-node orchestration would be an end-user terminal able to use both an enhanced access selection mechanism and a multi-path mechanism, as illustrated in OConS service B of Figure 3.2. Both mechanisms might be able to work in isolation, but their combination through OConS orchestration might bring additional benefits. Thanks to the OConS framework the DEs of each of these two mechanisms exchange information so as to orchestrate a new OConS service. The enhanced access selection mechanism will therefore select more than one alternative, taking into consideration that they can be used in parallel (as brought about by the multi-path functionality). Other illustrative examples of intra-node orchestration are the combination of the access selection with dynamic mobility management (provided that the corresponding DE is within the end-user terminal) and between multi-path and network coding mechanisms.

It is worth mentioning that an OConS-enabled node does not have to be a fully featured node, i.e., an infrastructure node with all the components defined in the functional architecture in Figure 3.5; however, we do need a minimum orchestration functionality to deal with the mechanisms

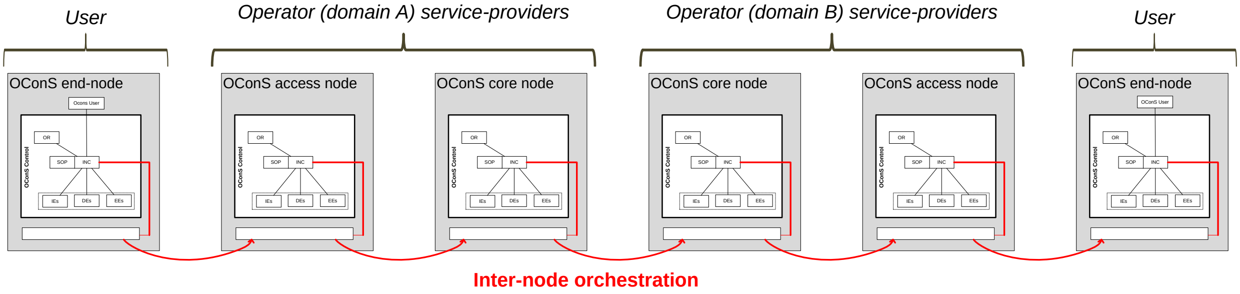


Figure 3.7: OConS Inter-Node View.

instantiated on that node, both node-internal orchestration, as well as the inter-node orchestration of mechanisms, if that is the case.

3.4.2 Inter-Node View

The second scope is an entire network domain, where the orchestration of mechanisms happens across several network nodes. In fact, the functional architecture described in Sec 3.3 can be within one OConS node (as seen in the previous section), or distributed over several OConS nodes (i.e., end-terminals, access-nodes, and core-nodes) within an OConS domain. Likewise, each OConS-enabled node may contain all the components or only a subset of them.

In this inter-node view, the OConS entities on one node communicate their data and policies also to entities hosted on other nodes, by means of OConS interfaces. A generic example of such situation is depicted in Figure 3.7. The red lines show examples of where orchestration can happen: between a user's end-node and an access node, or between access nodes and core nodes within the same administrative domain or, finally, between nodes belonging to different administrative domains, like two service-providers. Orchestration in this environment is more complicated, because it has to deal with discovery of other OConS nodes and, most importantly, of the involved mechanisms in order to provide an OConS service.

As a concrete (but simplified) example, we can consider again the example introduced in sec. 3.1, namely OConS Service B, depicted in Figure 3.2: an end-user node with a multi-path mechanism running on it. But, the access router to which the end-user node is connected, is also an OConS access node (see Figure 3.7) and it is running an admission control (resource management) OConS mechanism. Hence, an orchestrated OConS inter-node service could be thus provided as following:

- The SOP of the end-node, according to specific needs of the application, expressed through the OSAP by a demand profile, decides which mechanisms (between the available ones) need to be orchestrated so as to offer the most appropriate connectivity. It has previously performed the discovery procedure, and is therefore aware of the available OConS mechanisms at the access routers;
- The SOP decides to use the multi-path mechanism together with the admission control mechanisms running on the access routers;
- The DE entity of the Multi-Path mechanism configures the particular operation considering the possibilities which are provided by the Admission Control mechanisms of the selected access routers;
- The corresponding OConS functional entities of the Multi-Path mechanisms (both DE and EE) adjust the packet flows according to the information which is provided by the admission control mechanisms.
- On the other hand, the Admission Control mechanisms on the access routers might adapt their current resource allocation considering that a flow is simultaneously using more than

one path, possibly increasing the capacity which was allocated to other running flows.

In this simple example, the SOP is mainly taking place on the end-node. In more complex scenarios, where the mechanisms can span across several nodes in a network, it can be envisaged that a SOP will be hosted in a dedicated, centralized OConS node, which will "see" entities of the mechanisms from a network domain: in this case, OConS could replace or add up to the current distributed control plane that coordinates the traffic/flows over the infrastructure nodes.

The particular characteristics of the inter-node orchestration procedure depend on the level at which it is applied:

- Link-level orchestration can be done either between the two nodes at the link ends, or among several nodes within a link-broadcasted domain.
- The network level orchestration is typically achieved among several nodes within a network domain (e.g. this can include an arbitrary number of base station nodes, the access routers from a given mobility-enabled domain, etc.). Generally speaking, the orchestration at the network level can be done either in a fully-distributed or in a domain-centralized manner.
- Finally, the flow level orchestration is necessary between the two end-nodes that are communicating, but it may also include arbitrary number of nodes when using certain technologies at flow-level (e.g., a given number of caching/torrent nodes within a well-delimited domain).

Summarizing, one of the main characteristic of the OConS architecture is the ability to enhance the "control plane" for a wide number of mechanisms; this is due to its orchestration-based approach, which combines intra-nodal, domain-distributed (i.e., over a given set of nodes), domain-centralized (one node for a domain), and end-to-end cases. Later in this document, two possible deployment models (with nodal and inter-node views) are described in detail: OConS for CloNe in Chapter 5 and OConS for NetInf in Chapter 6, thus providing two concrete instantiation cases of the OConS architecture.

3.5 OConS Signalling and Interfaces

As was said earlier, the OConS entities and components exchange messages to handle mechanism registration, orchestration and communication between IEs, EEs, DEs. This section describes how this communication works, the OConS identifiers and the OConS messages. More details about the format and encoding of the messages can be found in Annex C.

3.5.1 Intra- and Inter-Node Communication Procedures

All OConS entities are identified by their OConS ID, which is built using a hierarchical scheme: it contains both the entity index and the OConS node on which it runs and it is encoded as an integer divided into two parts.

- The most significant bits serve to uniquely identify the OConS node, we refer to it as the node ID. All entities running on the same node will therefore share the same node ID.
- The least significant bits of the OConS ID are used to address the entity within a node. This entity ID is attributed dynamically by the local INC upon the entity's initial registration.

OConS is envisioned to span multiple nodes and domains, and it is therefore important to ensure unicity of the OConS node IDs. Learning from the design decision of IPv6, 128 bits seems a reasonable choice. Node ID 0 should be reserved for special addresses, such as various messages to be transported over broadcast and towards multicast groups. In addition, another 2 B is used to identify an entity within an OConS node. By convention, entity ID 0 should reach the INC itself.

The basic communication mode between OConS entities is in the form of requests and responses. Not all *requests* may however require a *response*, and some responses may be generated at multiple later times, in the form of *notifications*: the only difference is that a response is made within the context of a recent request, as identified by its message sequence, while a notification is unsolicited.

OConS entities generate messages addressed to local or remote entities using their specific identifiers obtained during the bootstrapping procedure as described in Annex C.3. However, OConS entities are agnostic of the bearer protocols and technologies in use to carry their messages, and only deal with each other by using their OConS ID. The tasks of selecting an appropriate transport[2] and routing the encapsulated message to the destination is mutualised per node in the form of the INC function.

The INC runs as part of the basic OConS infrastructure on each node. Local entities establish connections to it using any type of IPC relevant to the underlying system (*e.g.*, Unix sockets), and send and receive messages through this connection. The INC acts on behalf of the entities to relay these messages towards their destination, either local or remote.The INC system is in charge of resolving ID into the relevant lower layer locator and the subsequent encapsulation and routing towards it. In case of an IP network, the DNS-based mechanism defined in Annex D might be used.

Moreover, the OConS messages can be optionally encapsulated into security headers providing selectable combinations of integrity, authenticity and confidentiality.

In Figure 3.8, both a local and remote communication examples are provided. The local communication consists in a DE that wants to enforce a decision in a local EE. The associated steps are represented in blue in Figure 3.8, and are as follows:

A. The DE sends the OConS message with the enforcement decision towards the INC via the O_{DE} interface by using node-local IPC (e.g. UNIX socket).

B. The INC checks, in the message header, that the destination OConS ID is local, forwarding the message via the O_{EE} interface to the local EE.

The remote communication example consists on an IE that wants to report to a DE a certain information previously subscribed. The steps of this procedure are represented in red in Figure 3.8, and detailed as follows:

1. An OConS message is sent by the IE to the INC via the O_{IE} interface, as in the previous case.
2. The INC checks that the message is addressed to an entity in a remote node. It looks up the OConS ID, mapping to the underlying transport, in this case IPv6.
3. The INC is in charge of resolving the IDs into the IPv6, and the subsequent encapsulation in the IPv6 format and forwarding it towards IPv6 local via the O_{EXT} interface.
4. Transport using the standard method for reaching the destination node.
5. Reception and decapsulation, in the remote node, to recover the OConS message via the O_{EXT} interface.
6. Finally, the INC checks, in the message header, that the destination OConS ID corresponds to a given DE entity, to which the message is forwarded via the O_{DE} interface.

3.5.2 OConS Messages and their mapping on Logical Interfaces

This section describes the interfaces between the OConS entities and components. They will exchange information among them by using the messages defined in Section C.2. We have essentially proposed a Type-Length-Value (TLV) approach, as specified in Section C.1, with a common message header.

A message taxonomy is introduced, for the types of OConS messages and semantic classes. There are three main types of OConS messages:

- **Requests** are unsolicited messages, they elicit a response from the receiver;

[2] We use the term "transport" here in the context of the OConS INC, not in the sense of layer 4 of the OSI model but as a more generic reference to any packet-based system able to encapsulate our messages and carry them on wire.

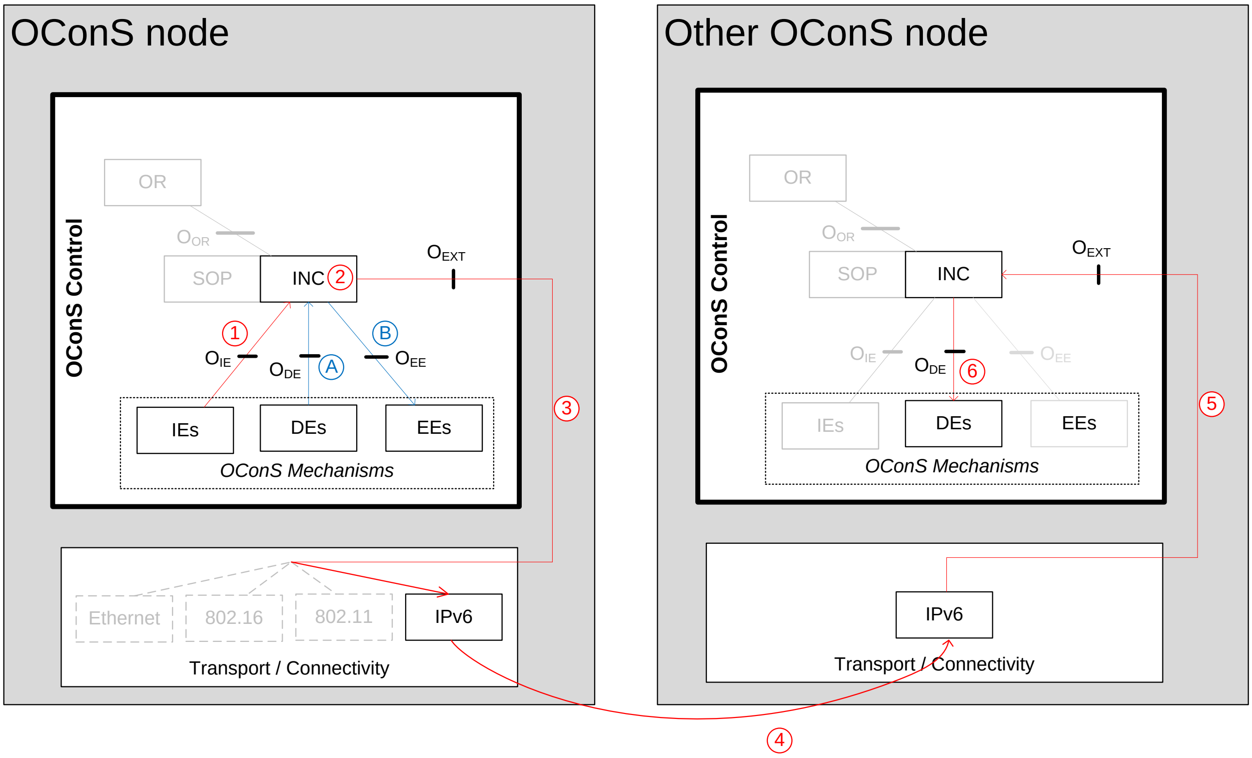


Figure 3.8: Intra- and inter-node communication procedures examples.

- **Responses** are sent as direct replies when requests are received;
- **Notifications** are unsolicited response-like messages; they can be sent, *e.g.*, periodically or when a specific event happens.

Five semantic classes of messages exist, depending on their role within OConS: Entity-handling, Publish/Subscribe, Mechanism-handling and Inter-entity messages, OConS-users-handling. These classes are described next, identifying the interfaces where they are used and defining the associated messages.

Entity-handling messages are used by the orchestration functionalities as to initially register entities and identify the available mechanisms (inspired from the IEEE 802.21 standard messages and operations, see [11]). The following messages are used on the logical interfaces O_{IE}, O_{DE}, and O_{EE}, as well as on the O_{EXT} for the remote operation:

- **DISCOVER_req** sent by SOP towards the entities so as to find their capabilities either after receiving an advertisement or based on some policies.
- **DISCOVER_rsp** sent as response to the previous message.
- **CONFIGURE_req** from SOP towards the entities to configure some parameters.
- **CONFIGURE_rsp** confirmation from the entities with the current configuration status.
- **REGISTER_req** sent from the entities towards OR in order to register their **ID**'s and capabilities, it can be locally or remotely.
- **REGISTER_rsp** acknowledge of a registration from the OR to the entities.
- **INFORMATION_notif** sent by the entities towards OR in according to some configuration or policy.

Publish/Subscribe messages are exchanged within the OR to handle mechanisms and services (i.e., based on the publish/subscribe approach with its benefits). The following associated messages are used on the logical interface O_{OR}, as well as on the O_{EXT} for the remote operation:

- **SUBSCRIBE_req** from SOP to OR to subscribe for a OConS mechanism/service or to be informed about changes produced within a mechanism/service.

- **SUBSCRIBE_rsp** from OR to SOP as acknowledge of the request.
- **PUBLISH_notif** sent by OR towards SOP to inform about mechanism/service as well as change within these if so configured.
- **VALIDATE_req** from SOP to OR so as to validate a given mechanism/service once instantiated.
- **VALIDATE_rsp** acknowledge to the previous message.

Mechanism-handling messages are sent by the SOP to instantiate and enable the required mechanisms (inspired from the CRUD operations used in web services). These messages are mainly used on the logical interface O_{DE}, as well as on the on the O_{EXT} for the remote operation. Likewise, the reading of the information is done through the O_{IE}. These are as follows:

- **UPDATE_req** from the SOP towards mechanism' DEs to instantiate (i.e. to update, create, re-create or launch) a given mechanism.
- **UPDATE_rsp** response to the previous request informing about mechanism status by codes.
- **READ_req** sent by SOP towards any entity to read some state so expanding the information gathered through OR.
- **READ_rsp** from the entities to the SOP with the requested values.
- **DELETE_req** sent by SOP to the mechanism' DEs in order to delete or remove a given mechanism from the current service.
- **DELETE_rsp** acknowledge to the request with status information by means of codes.
- **ASSIGN_req** sent from the SOP towards the mechanism' DEs to establish a relationship between mechanisms to make them able to work together (i.e., we have enriched the classical CRUD approach with this assignment operation).
- **ASSIGN_rsp** response to the previous message with some response codes.

Inter-entity messages are exchanged between the entities, usually involved in the same mechanisms. These messages are used internally by the mechanisms themselves (please see also [1, 2] for more details on these). These are as follows:

- **EXECUTION_req/rsp** it is exchanged between an EE and a DE or other EEs (for hierarchical execution approach) so as to enforce an action. The response contains information about the execution status.
- **INFORMATION_req/rsp/notif** they are messages sent between a DE (or an IE for hierarchical approaches) and an IE so as to gather information. An IE can also send information notification attending to some configuration.
- **CONFIGURATION_req/rsp** they are sent between a DE and an IE so as to configure notifications (subscription).

3.5.3 OConS Messages for OSAP and Demand Profile

OConS-users-handling messages are used by OConS nodes to receive the OConS-user demand profile, that specifies the connectivity requirements, and to send notification messages of the status of the OConS service. The demand profile is defined to express OConS user's connectivity requirements through the $OSAP$[3]. In the simplest form, a set of pre-defined and pre-configured demand profiles is available on the OConS node: the user can simply select one of them, but he can also modify the existing or even create new demand profiles.

Two messages are defined:

- **DEMAND_PROFILE_req**: contains the Demand Profile and a set of User-specific parameters, which include a Callback reference, which the SOP can register with, to provide feedback to the caller application on the status of the OConS Services requested

[3] I.e., an operation quite similar with the SDP declaration from the SIP protocol suite

- **DEMAND_PROFILE_rsp**: provides an acknowledgement of the result (and cause) of the demand profile request, i.e. if it has been accepted/rejected, meaning that an adequate OConS service has been or not launched.

Table 3.2 gives the definition for the demand profile, contained in the DEMAND_PROFILE_req.

Note that the "Values" column contains the possible values for a particular field. We use M for depicting mandatory entries and O for optional entries in this table. The words in parenthesis indicate possible alternative for a specific sub-field. The parameter "value" is a numeric value, meaningful for the specified metric. In the "Preferences" field, more than one sub-field can be included. Finally, the "Weight Factors" is a set of relative weights, one for each Required QoS Parameter included in the profile.

An example of use of the demand profile can be found in Sec.3.7.5.

Another typical usage of this OConS (northbound) interface by CloNe is requesting a connectivity service based on an abstract 'single router network' description in Open Cloud Network Interface (OCNI) format (see example Table B.1 in Annex B). Further details on the use and the message exchange at the CloNe–OConS interface are described in Section 5.2 and in D.A.3 [12].

Field	M/O	Values
Profile ID	M	-
Profile Name	M	-
Source (of the demand)	M	– OConS User ID
Source Type	M	– End-user – CloNe Application – (other) Application – Self-determined by OConS
Parameters	O	– Connectivity Type (i.e., link, network, or flow) – QoS Parameter (zero or more of the following 3-tuple) – metric (data-rate, loss, delay, jitter) – value (it depends on the metric) – type (default, max, minimum) OR – reference address of an extended demand description [see example Table B.1 in Annex B] – description format used (e.g. OCNI , XML, etc.)
Security constraints	O	– Encryption – Integrity – VPN – Domain-scoping
Preferences and User-specific	O	– Access Technology (wireless [3G, 4G, Wi-Fi], wired) – Network Provider – Requested/Excluded mechanisms (from OConS list of mechanisms) – Weight factors – End-points of CloNe Application
Callback reference	O	– A callback to register with it and to provide feedback to OConS user/application

Table 3.2: Demand Profile Table

3.6 OConS Information Model

In the previous sections, some data structures, used by OConS, have been described: the Mechanism Manifest and the Demand Profile. However, as already mentioned in Section 3.3, we also need a more extended OConS Information Model, which formally represents and describes all the OConS concepts, defining their structure, their properties and relationships.

Accordingly, the OConS Information Model is used to capture:

- The abstraction of the main OConS concepts: node, entity, domain, Registry, SOP etc.
- The abstractions of what the SOP sees and manipulates, e.g., mechanisms (which characteristics are defined in the Mechanism Manifest, see 3.2.2), services, policies and rules for orchestration/composition, network view/status, and so on;
- The requirements and policies from OConS users , i.e. as received through the OSAP or implicitly inferred: these are defined in the Demand Profile (see 3.5).
- The information to be used by IE, DE, and EE, when controlling the orchestration of a specific mechanism. It can be re-used by several mechanisms;
- The information a given OConS mechanism depends upon, e.g., QoS profile, traffic profile etc.

Annex B presents a consolidated view for the OConS Information Model, specifically based on the experiences related to the use-cases we have investigated. In particular, the Information Model is linked to CloNe use-case to put it into relation with the overall SAIL context; however, this could as well be applied to other use-cases, e.g, the one concerning the NetInf.

It is worth noting that, in this deliverable, we are not aiming at having an implementation-ready specification for the whole OConS Information Model. Rather, it will serve as a base to further clarify the OConS concepts and their semantics, making them more easy to grasp by the readers or designer of orchestration solutions. Accordingly, we do not present here all the possible levels of abstraction, nor their mappings to each specific OConS mechanism, primarily due to time and space constraints.

Some parts of the OConS Information Model needs eventually to be translated into a Data Model, i.e., specifying how attributes in the Information Model will be implemented in registers or databases to meet structures and data encoding constraints and how they are carried in messages. The Data Model will assure that the necessary implementation structures are identified in a consistent manner.

3.7 OConS Orchestration

3.7.1 Overview

The orchestration of different OConS mechanisms to provide an OConS service is done in three different time spans, as represented in Figure 3.9. First, when configuring OConS nodes, the capabilities and the required elements of the mechanisms need to be specified and attached with the mechanism itself. In the same way, user profiles need to be pre-configured for appropriately responding to the service requests. In a second time span, when booting the nodes, the mechanisms need to be published in suitable Orchestration Registers. Predefined default mechanisms are also launched (legacy and/or OConS mechanisms). Finally, when a user, such as an application, NetInf, CloNe or another network, requests connectivity with specific characteristics, a service needs to be selected or instantiated (i.e., orchestrated). In this sense, the orchestration functionality serves an explicit request by a user or also an implicit request triggered when a given monitored network state changes. Accordingly, the Orchestration identifies the most appropriate OConS mechanisms from the set of those available, to address the connectivity requirements.

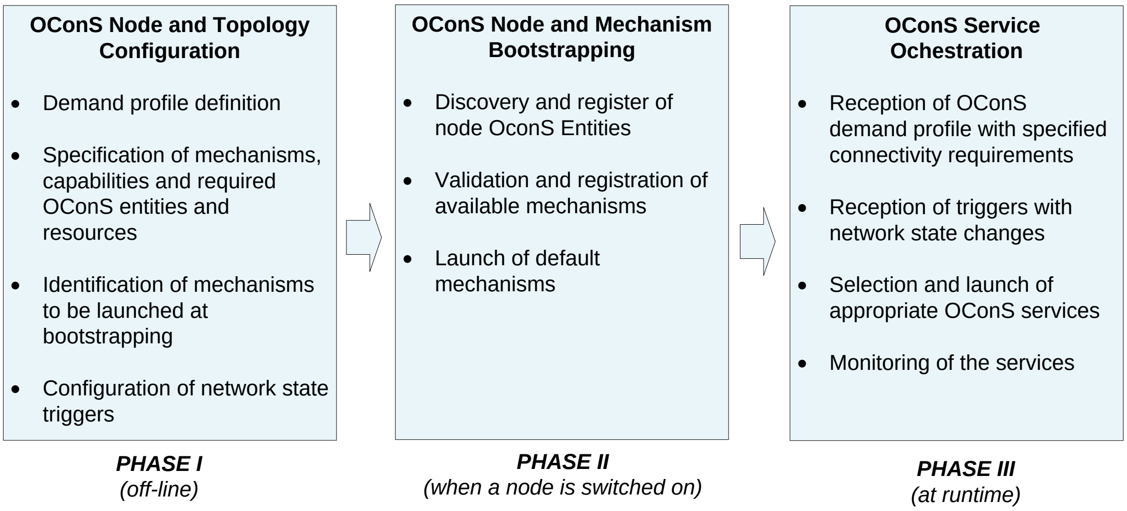

OConS Node and Topology Configuration
Demand profile definition
Specification of mechanisms, capabilities and required OConS entities and resources
Identification of mechanisms to be launched at bootstrapping
Configuration of network state triggers
PHASE I
(off-line)
OConS Node and Mechanism Bootstrapping
Discovery and register of node OconS Entities
Validation and registration of available mechanisms
Launch of default mechanisms
PHASE II
(when a node is switched on)
OConS Service Ochestration
Reception of OConS demand profile with specified connectivity requirements
Reception of triggers with network state changes
Selection and launch of appropriate OConS services
Monitoring of the services
PHASE III
(at runtime)

Figure 3.9: Orchestration phases.

The various phases of the orchestration process identified in Figure 3.9 are described next, starting from the OConS node and topology configuration, continuing with the OConS nodes and mechanisms bootstrapping, and finalizing with the OConS service orchestration.

3.7.2 OConS Node and Topology Configuration

For providing OConS services, the nodes that are exposing an interface to the OConS users need to obtain the user demand profiles through a suitable OSAP. The demand profiles, see Section 3.5.3 for details, allow the mapping of the profile ID (as selected by the users) to the service requirements, the user preferences, and, possibly, to the preferred or default mechanisms required for this service profile. These user demand profiles need to be defined and stored within or accessed by OConS nodes in phase I, and then will be used when a node boots up.

In some cases it is important to preselect mechanisms to be employed for different user demands. They need to be defined in phase I as well and can be determined in different ways: First, the selection can be based on off-line simulation and mathematical analysis, e.g. determining the provided Quality of Service (QoS) for real-time services using certain combination of mechanisms. A second alternative is that this pre-selection can be based on monitoring results of previous usage of the combination of mechanisms; to refine this, machine-learning and artificial intelligence algorithms could be applied. Third, this can be based on the expertise of a network planner and/or the OConS user feed-back.

On the other hand, when configuring an OConS node, the mechanisms supported by that node need to be defined. This means the capabilities of the mechanisms need to be known, same as its requirements to function correctly[4]. Likewise, the mechanisms that need to be launch at bootstrapping independent of a user demand profile could also be pre-configured (i.e., *default* mechanisms). To be able to be correctly executed, the mechanisms themselves require certain further functional elements on the same or other nodes, specified connectivity and computational resources on the node. These requirements can be expressed with the Mechanism Manifest as given in Section 3.2.2.

[4] For some mechanism, the expected performance should be known in advance, for the orchestration to be able to take them into account in selecting the right set of mechanisms. Although it can be very cumbersome to define performances of a mechanism in absolute terms, it could be anyway possible that performances are known in advance, for example, based on simulation or monitoring the previous applications.

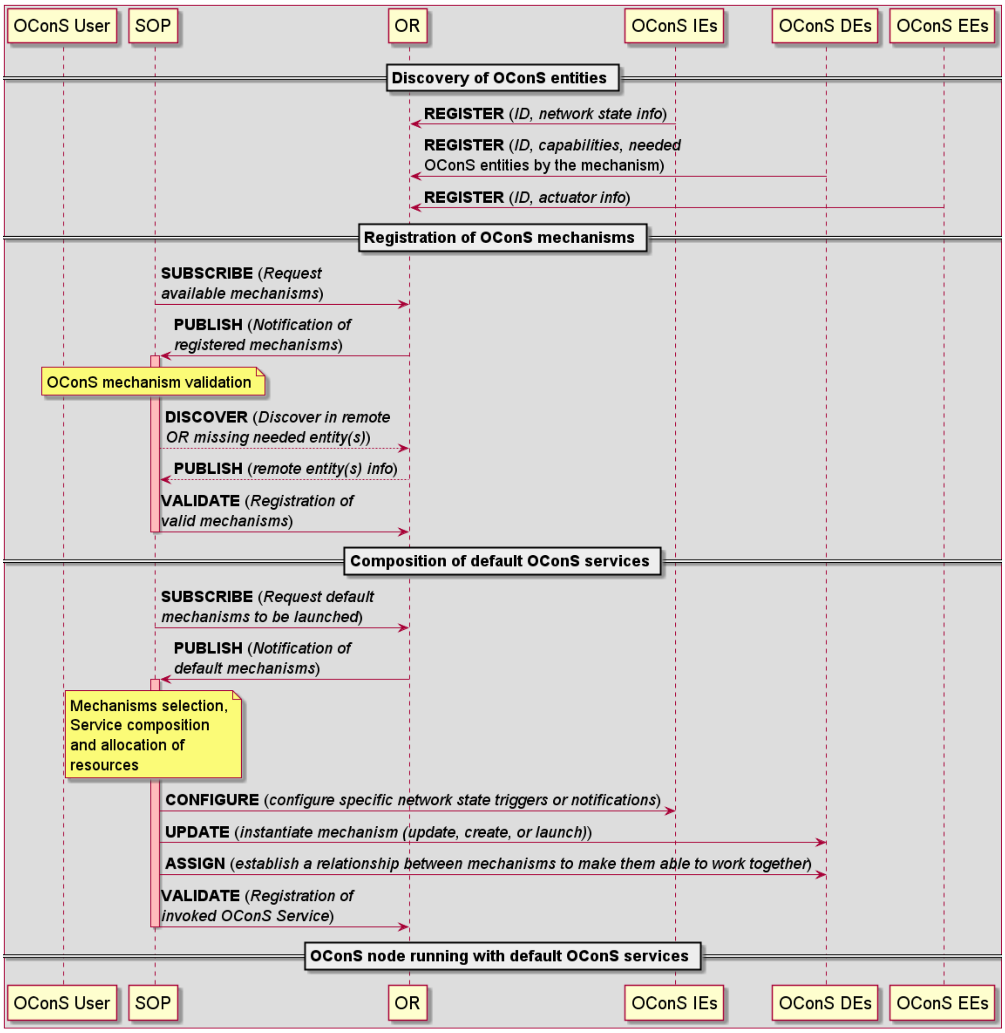

OConS User
SOP
OR
OConS IEs
OConS DEs
OConS EEs
Discovery of OConS entities
REGISTER (ID, network state info)
REGISTER (ID, capabilities, needed OConS entities by the mechanism)
REGISTER (ID, actuator info)
Registration of OConS mechanisms
SUBSCRIBE (Request available mechanisms)
PUBLISH (Notification of registered mechanisms)
OConS mechanism validation
DISCOVER (Discover in remote OR missing needed entity(s))
PUBLISH (remote entity(s) info)
VALIDATE (Registration of valid mechanisms)
Composition of default OConS services
SUBSCRIBE (Request default mechanisms to be launched)
PUBLISH (Notification of default mechanisms)
Mechanisms selection, Service composition and allocation of resources
CONFIGURE (configure specific network state triggers or notifications)
UPDATE (instantiate mechanism (update, create, or launch))
ASSIGN (establish a relationship between mechanisms to make them able to work together)
VALIDATE (Registration of invoked OConS Service)
OConS node running with default OConS services

Figure 3.10: Orchestration bootstrapping.

3.7.3 OConS Nodes and Mechanisms Bootstrapping

Bootstrapping is triggered when an OConS node is switched on. We can structure it according to three different steps, discovery of OConS entities, registration of OConS mechanisms and composition of default OConS services, as depicted in Figure 3.10. This is an interactions procedure diagram, showing the various messages exchanged in the various steps between architectural components. Messages are detailed in Section 3.5.2. For simplification, acknowledgement “reply” messages were not represented in the figures below. Furthermore, for the sake of simplicity, the INC is not represented in this figure, although, in practice, it is always an intermediary in the exchange of any message between architectural components, all existing interfaces being linked to the INC (for details on the INC see also Section 3.5.1). All represented architectural components may be located in the same OConS node, or distributed over different nodes in the OConS domain (e.g., the SOP in one node, the OR in another, the entities in multiple ones). The INC supports this, guaranteeing the transparent inter-node transport of messages.

The first step comprises the discovery of OConS entities. Upon being switched on, the OConS entities send a REGISTER message to the OR (either locally or remotely, as instructed by the SOP with the help of the CONFIGURE messages), indicating their ID, and their specific capabilities: for an IE, the network state information they are related to; for a DE specifying a mechanism, its manifest (see Section 3.2.2) which includes capabilities and identification of needed OConS entities; for an EE, the actuator information. If the local OConS node does not have an OR, the INC shall forward the received REGISTER messages towards a remote OR in charge.

The second step affects the registration of OConS mechanisms. The SOP sends to the OR a SUBSCRIBE request of the available registered OConS mechanisms. Within the identified OConS mechanisms, notified in a PUBLISH message, the SOP validates the ones that are complete in terms of needed OConS entities in order to operate. In the case a mechanism needs an OConS entity not available in the node or that must be located in a specific remote node, a DISCOVERY message is sent to discover the ID of the corresponding entity. The INC is in charge of forwarding this message in the form of a broadcast or unicast (depending if the destination node is known or not). The mechanisms that have available all needed entities are registered as valid OConS mechanisms in the OR via a VALIDATE message.

Finally, default OConS services are launched. For it, the SOP requests the OR for validated default OConS mechanisms to be launched in the bootstrapping. The notified mechanisms are composed in OConS service(s). The mechanisms of each service are instantiated through a UPDATE message and relationships between mechanisms are established through ASSIGN message to make them able to work together. Also, specific network state triggers or notifications are also set via a CONFIGURE message. The OConS Services are invoked and then registered in the OR via a REGISTER message, the node being now running with the default OConS services.

3.7.4 OConS Service Orchestration

When an OConS user (i.e., application, CloNe or NetInf) requests connectivity, the orchestration must provide the most adequate OConS service that satisfies the user specific connectivity requirements. Figure 3.11 provides an overview of this procedure. In the first step, the user demand profile, which consists of the information of connection requirements, is received via the OSAP. In the second step, the most suitable mechanisms or combination of several mechanisms are selected out of available ones. In the third step, the selected mechanisms are instantiated and configured to provide the required OConS service. Finally, in the fourth step, the service and network state is being monitored. Unsatisfied service performance or a change of network state can trigger the Orchestration to start a re-orchestration process.

The service orchestration procedure is described in more detail by the sequence of interactions depicted in Figure 3.12 which are detailed next.

In the first step, OConS user's connectivity requests are expressed by a DEMAND_PROFILE_req message, its content being detailed in Section 3.5.3. The user can either select a pre-configured demand profile, modify an existing one or create a new one. The DEMAND_PROFILE_req message is received via the OSAP interface. It can simply contain the QoS requirements of an OConS user, e.g., at least 2 Mbps and a delay below 40 ms, yet it can also specify that the user wishes to have seamless access to a cloud service. Accordingly, with the help of the user demand profile, the connectivity requirements can be deduced by the SOP. Once the connectivity requirements are obtained, a SUBSCRIBE message is sent to the OR requesting it to PUBLISH the available mechanisms. The network state is also requested through READ messages to be notified by the various IEs, measuring key performance parameters.

In the second step, the mechanisms to be used, are selected. The mechanism selection needs to consider the connection requirements deduced from the user demand profile, the current network topology and network state, and mechanism manifest (see Table 3.1) which provides relevant information of each mechanism (what it provides/improves, what are the needed resources, constraints and requirements). Additionally, in case an appropriate combination of OConS mechanisms would leverage a yet better operation, several OConS mechanisms can be also combined in a coordinated manner to provide one OConS service (see 4.6).

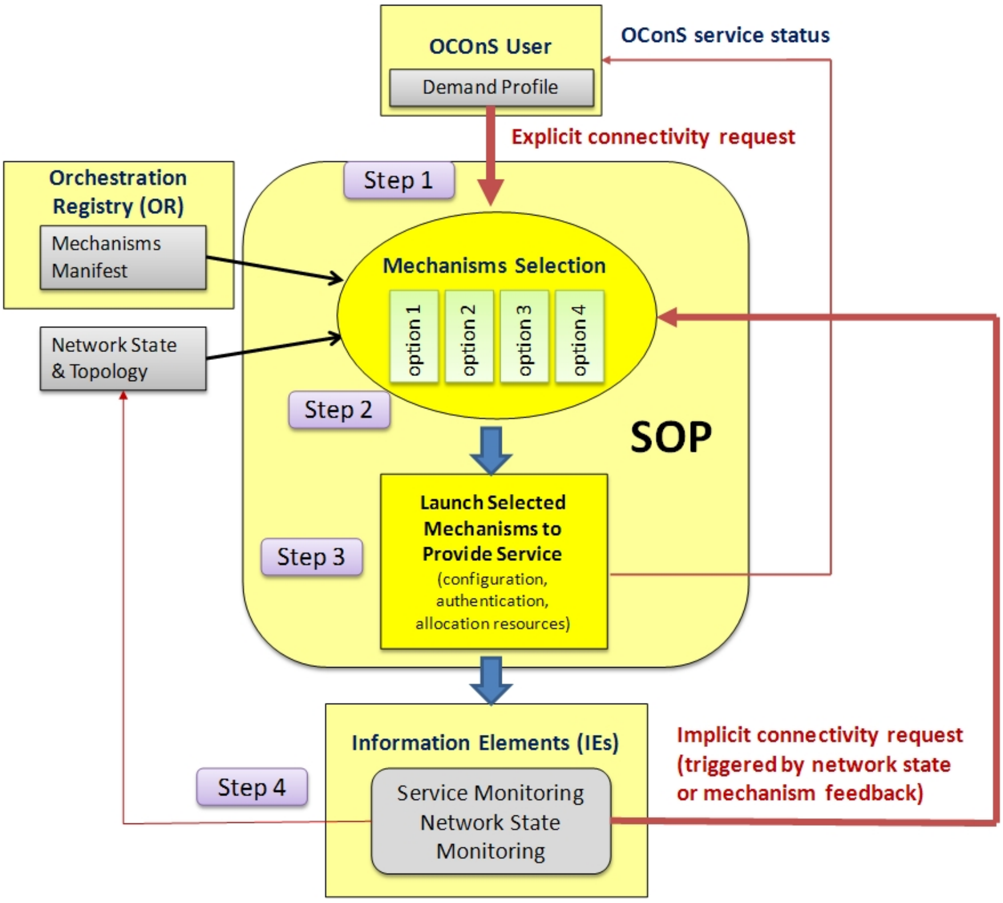

OCOnS User
Demand Profile
OConS service status
Explicit connectivity request
Orchestration Registry (OR)
Mechanisms Manifest
Step 1
Mechanisms Selection
option 1
option 2
option 3
option 4
Network State & Topology
Step 2
SOP
Launch Selected Mechanisms to Provide Service
(configuration, authentication, allocation resources)
Step 3
Information Elements (IEs)
Step 4
Service Monitoring
Network State Monitoring
Implicit connectivity request (triggered by network state or mechanism feedback)

Figure 3.11: OConS service orchestration procedure

To select the suitable OConS mechanisms or combination of them, we have identified four options, other options might be possible:

1. In the first option, we consider an OConS domain with a domain centralized SOP, which is taking the service request and which has an overview on network state; this means information about the link and network load, available bandwidth and available nodes and their mechanisms is known. This also would include the earlier described information on the performance of each mechanism and the potential conflict or benefit of mechanisms' interaction. Then, for finding the optimal combination of mechanisms, an exhaustive search over all possible combinations or another mathematical formulation can be used (e.g., a constrained optimization problem with a multidimensional utility-cost function). This can consider the user requirements expressed, but also network operator interests like minimum usage of resources.

2. The second option considered, is where at an OConS-node's SOP only some local network state information is known. In this case a hop-by-hop approach can be followed (e.g., such as those used for paths computation).

3. In the third option, very limited information of the network state is available. For these cases, default combinations of mechanisms for the user demand profile are defined in the configuration phase. The default mechanisms can be updated based on monitoring the network state as described in the fourth step of the orchestration process.

4. As a fourth option for selecting mechanism, we may have (also from the configuration phase) a preconfigured sets of mechanisms for the demand profiles defined (as described above). There

might be one mechanism or a limited number of mechanisms which can then be combined. This is very similar for the legacy applications, where no explicit user demands are expressed and so default OConS services are started.

In the third step, once the mechanisms have been chosen for the OConS service to be launched, these are instantiated through an UPDATE message with subscription of their status and performance monitoring. Besides this, specific network state triggers or notifications are also set via a CONFIGURE message. Relationships between mechanisms are also established through ASSIGN message to make them able to work together. The OConS Service is invoked and then registered in the OR via a REGISTER message. Closing this, the OConS user is notified about the availability or not of the OConS service through a DEMAND_PROFILE_rsp message via the OSAP.

Finally, we have the service monitoring (i.e., subscribed when the various mechanisms are instantiated) and network state monitoring through configured network state triggers. If the service performance does not meet the user requirement or the network state changes, this may trigger the Orchestration to start a (re-)orchestration process for an OConS service. If needed the selected set of the mechanisms may be changed, e.g., based on the information provided by the mechanism manifest (such as what metrics each mechanism improves). Alternatively, a new OConS service can be created, e.g., in order to increase delivered QoS or decrease networking resources utilization. This procedure is illustrated by the sequence of interactions depicted in Figure 3.13.

Thus, whenever some thresholds of a subscribed network information are reached, the SOP is notified via INFORMATION and PUBLISH messages by the group of IEs in charge of network monitoring. The SOP uses the list of available OConS mechanisms, proceeding to the re-selection of the most appropriate ones and the composition of the OConS service, i.e., similarly to the previous case when an OConS user requests a service.

Distributed Orchestration

In more complex scenarios, with more services spanning over several levels and including larger number of mechanisms, a distributed and hierarchical orchestration can be considered; for example, this can be done using an orchestration function per OConS domain, where the SOP from the initiating domain orchestrates the services together with the neighbouring domains to provide end-to-end connectivity services. Even within one domain, e.g., see the case of the hop orchestration option (Option 2 above mentioned), different link and flow network connectivity mechanisms will be orchestrated and need to monitored after they were launched.

Moreover, one can consider different time spans for the monitoring of the different mechanisms, i.e., within each domain and on the overall end-to-end connectivity service. As examples, the link connectivity mechanisms might need to react in milliseconds, the flow management and mobility decisions might be taken in hundreds of milliseconds or seconds, whereas end-to-end services for data centres can be on much larger time span. Likewise, a hierarchy of the orchestration process (and also the monitoring of the network state and the reactions to it) will allow an improved scalability of the OConS orchestration.

Within the runtime of SAIL the basic concepts for the orchestration have been defined and their feasibility has been shown in demonstrations and prototypes. However, proofing the scalability of (an distributed) OConS orchestration and evaluating its performance in sufficiently large scenarios (and against other networking approaches) was beyond the scope of OConS work.

3.7.5 Orchestration Example

In this section we extend the description of "OConS Service B", presented in 3.1, Figure 3.7, to provide an example of applying a centralized orchestration where, in an OConS domain, the complete network state and performances of mechanisms of every OConS node are known by a

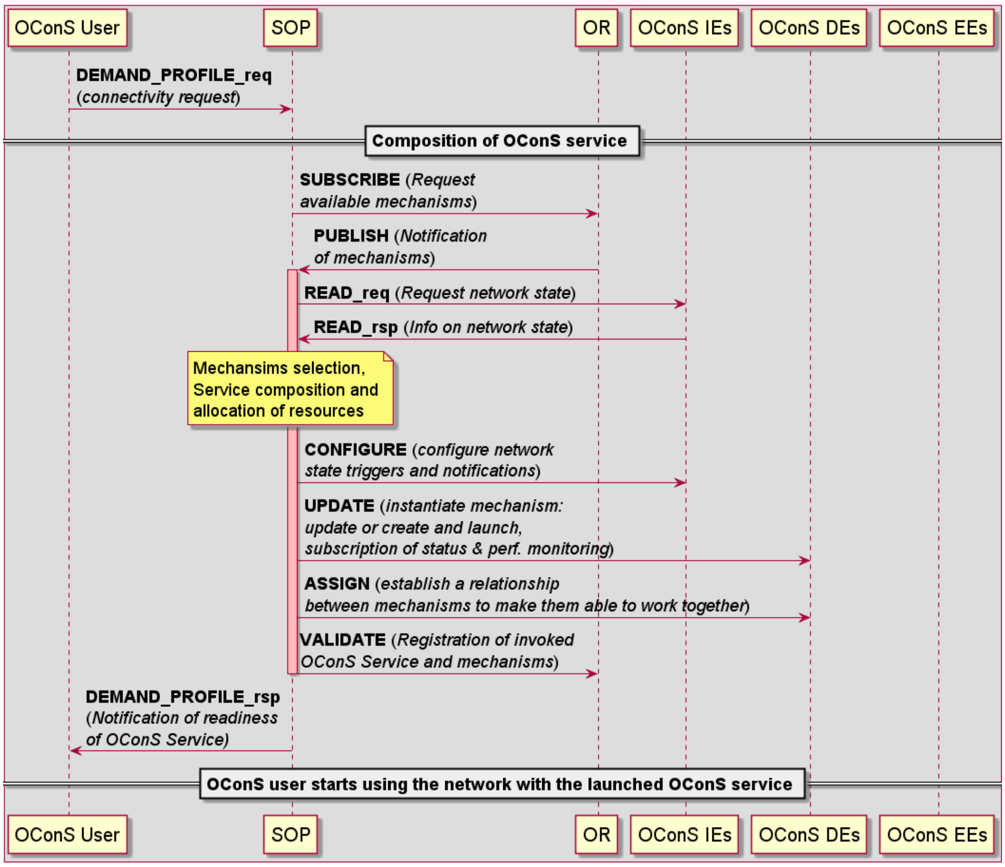

OConS User
SOP
OR
OConS IEs
OConS DEs
OConS EEs
DEMAND_PROFILE_req
(connectivity request)
Composition of OConS service
SUBSCRIBE (Request
available mechanisms)
PUBLISH (Notification
of mechanisms)
READ_req (Request network state)
READ_rsp (Info on network state)
Mechansims selection,
Service composition and
allocation of resources
CONFIGURE (configure network
state triggers and notifications)
UPDATE (instantiate mechanism:
update or create and launch,
subscription of status & perf. monitoring)
ASSIGN (establish a relationship
between mechanisms to make them able to work together)
VALIDATE (Registration of invoked
OConS Service and mechanisms)
DEMAND_PROFILE_rsp
(Notification of readiness
of OConS Service)
OConS user starts using the network with the launched OConS service
OConS User
SOP
OR
OConS IEs
OConS DEs
OConS EEs

Figure 3.12: Response of the Orchestration to a connectivity request from an OConS user.

domain centralized SOP. In the original example, the end-user simply accesses a storage area in a Data Centre and has two mechanisms running on his node, namely access selection and multi-path selection. [5]. In this section, we provide more details on the user demands and on the selection process of OConS mechanisms.

The OConS end-user wants to launch a Video Streaming application, which requests a demand profile "Video Streaming" using a specific profile ID. The demand profile, summarized in Table 3.3, specifies the user requirements, i.e. a guaranteed data rate of 300 kbps, an end-to-end packet delay less than 100 ms, a maximum of 5% loss and the preference of a cheap connection (i.e., low costs).

The network scenario is shown in Figure 3.14. Here the OConS Node 1 (the user) can reach the OConS Node 4 (video server) on two different paths: one via the Node 2 and the other via Node 3.

The orchestration procedure is as follows: in the first step, once the OSAP receives the demand profile ID from the user it forwards the profile to the SOP. Then the SOP can directly get the connection requirements from the user profile. In the second step, based on all the collected information (such as connectivity requirements, network state, the availability of OConS mechanisms including the availability of their required IEs/DEs/EEs, the performance of each mechanism), the SOP firstly selects the most appropriate mechanisms as the candidate mechanisms. For this example, there are in total 2 Flow Mechanisms (FM), 5 Network Mechanisms (NM) and 2 Link Mechanisms (LM) available, as listed below:

- two Flow Mechanisms FM:
  - FM1: unreliable flow mechanism

[5] In sec 3.4.2 we have then extended the example, adding a resource management mechanism running on the access router of the node

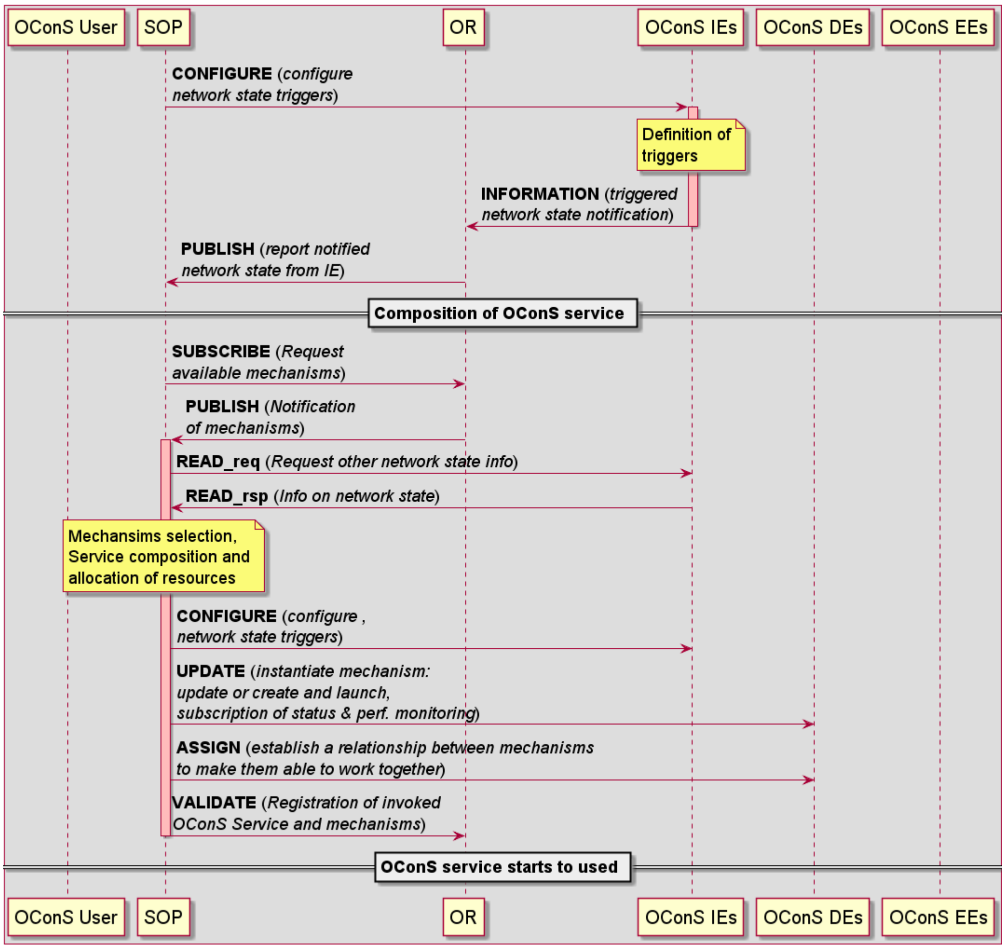

OConS User
SOP
OR
OConS IEs
OConS DEs
OConS EEs
CONFIGURE (configure network state triggers)
Definition of triggers
INFORMATION (triggered network state notification)
PUBLISH (report notified network state from IE)
Composition of OConS service
SUBSCRIBE (Request available mechanisms)
PUBLISH (Notification of mechanisms)
READ_req (Request other network state info)
READ_rsp (Info on network state)
Mechansims selection, Service composition and allocation of resources
CONFIGURE (configure , network state triggers)
UPDATE (instantiate mechanism: update or create and launch, subscription of status & perf. monitoring)
ASSIGN (establish a relationship between mechanisms to make them able to work together)
VALIDATE (Registration of invoked OConS Service and mechanisms)
OConS service starts to used
OConS User
SOP
OR
OConS IEs
OConS DEs
OConS EEs

Figure 3.13: Response of the Orchestration to a network state triggered event.

  - FM2: reliable flow mechanism
- five Network Mechanisms NM:
  - NM1: always take route via Node 2 (legacy mechanism)
  - NM2: always take route via Node 3 (legacy mechanism)
  - NM3: always take best route
  - NM4: Multi-path (flow switching: each flow over one path)
  - NM5: Multi-path (flow splitting: each flow over two paths)
- two Link Mechanisms LM:
  - LM1: reliable link transmission (with Acknowledgment (ACK))
  - LM2: unreliable link transmission (without ACK)

Then SOP needs to choose the best combinations of mechanisms from the selected candidate ones to compose the OConS service at the end. In this example, we apply the first option (Section 3.7.4) for mechanism selection process. At first, we list all possible combinations (i.e., solution space) from all candidate mechanisms.

Then from the list of all possible combinations, we select the best combination. One possible way to solve this problem is to use a cost function. To create the cost function, we need to define the objectives which can be composed of one or multiple criteria/metrics and their weights. For solving the problem the constraints (user requirements) and the given network state etc. need to be considered. With the full network state information, we estimate the performance of each combination, and we can select the best solution by applying exhaustive search, linear programming,

Field	**Values**
Profile ID	123
Profile Name	"Video Streaming"
Source (of the demand)	1235
Source Type	End-user
Required QoS Parameters values	– • type = minimum • metrics = *data-rate* • value = 300kbps – • type = maximum • metrics = *delay* • value = 100ms – • type = maximum • metrics = *packetLoss* • value = 5%
Preferences	– Weight Factors = (1,1,1) – Cost = "cheap"

Table 3.3: Demand profile.

heuristic approaches (e.g, greedy algorithm), or other optimization methods.

For example, the objective function can be formulated as follows.

Objective:

$$\max(\alpha T(throughput) - \beta D(delay) - \gamma L(loss) - \delta C(cost)) \tag{3.1}$$

Here α, β, γ, and δ are the weights for different performance metrics, and $T()$, $D()$, $L()$, $C()$ are scaling functions for different QoS metrics. The metrics are expected throughput, delay, loss when using a combination of mechanisms under the current network state, and the cost is the expense of using this combination.

Constraints:

- the obtained throughput should support the required throughput
- the delay/loss shall be less than the required delay
- other potential constraints can be qualitative properties (e.g., reliability), service compatibilities, etc.

3.8 Coexistence with Legacy and Migration

In order to smoothly introduce the OConS services in today's networks one has to consider the legacy applications as well as the legacy mechanisms, since not all of them are expected to be aware of the OConS framework (see also the upcoming deliverable D.A.4).

In case of the legacy applications, it is envisaged that the orchestration intercepts the legacy applications connectivity requests (e.g., application launching, User Datagram Protocol (UDP)/Transmission Control Protocol (TCP) socket opening, network edge-to-edge connection signalling, and so on) and it translates them into OConS connectivity requirements to be used in relation with the OConS

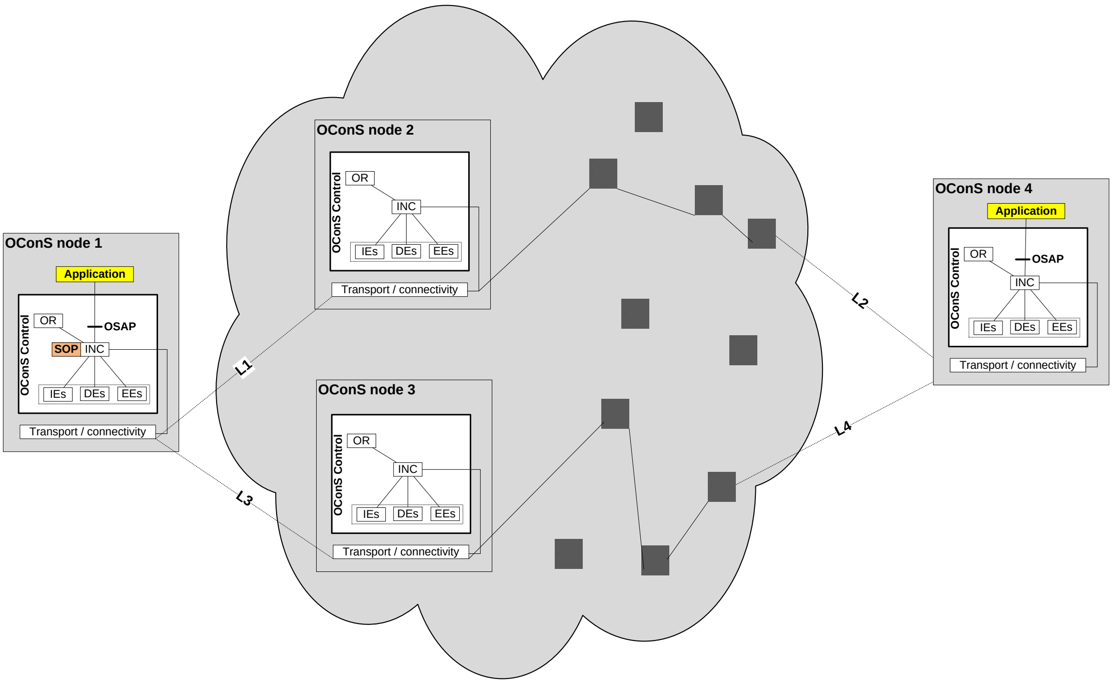


Figure 3.14: Network scenario.

mechanisms or legacy connectivity mechanisms, including the employment of the related OConS orchestration rules and policies.

When it comes to the legacy mechanisms, they need to be pre-registered in the orchestration register with the associated capabilities that should be modelled based on OConS approach. Besides this, the legacy mechanisms should also be included in the orchestration rules/policies in order to be launched whenever the connectivity requirements involve their activation and usage.

Moreover, when the connectivity service is composed by legacy mechanisms, the connectivity service monitoring could be also delegated by the SOP to the mechanisms themselves.

4 OConS Mechanisms

In this chapter, we present a set of dedicated mechanisms which are of particular interest and have been developed within the scope of the SAIL project. Those have been grouped according to three different levels. We then shortly present some illustrative interactions of those mechanisms, showing the benefits gained by combining a few mechanisms together.

4.1 Overview of Developed OConS Mechanisms

Future networks need to rapidly adapt to the changes in traffic patterns that are primarily driven by increased mobility, the smart-phone revolution and social networking. This required dynamic on-demand behaviour is not adequately supported by the connectivity services of the current Internet and mobile systems, due to a variety of deficiencies at various levels, including physical/data link, routing and transport, and flow/session control. These deficiencies include, among others, the lack of exploitation of multi-path, bottleneck situations which could be overcome with load-balancing schemes, or inefficient management of various link resources.

The scientific community (including the work which has been carried out within OConS) has come up with different proposals to tackle some of the previously mentioned deficiencies. Most of them are monolithic and isolated solutions which solve particular problems. Going beyond this we advocate that appropriate combinations of those mechanisms will bring about some additional benefits and OConS aims at achieving this goal, as well as solving some gaps.

The large variety of the OConS mechanisms are managed, combined, and launched in the scope of the OConS architecture that was described in Chapter 3. For instance, appropriate combination of multi-path and routing (e.g. policy-based) might help to alleviate congestion situations which might affect the interconnection of data-centres. Furthermore, looking at the access network part, a proper interaction between flow and link level connectivity procedures is also of high relevance, since for instance, the use of improved mobility mechanisms can help to alleviate the negative effects suffered by TCP-based applications; moreover, appropriate handling of multi-homing systems can also help to provide higher throughput and better QoE to the end-users.

OConS customizes and optimizes connectivity services for various types of networks. Two use cases will be used to illustrate the benefits of applying OConS mechanisms (of particular interest within the scope of the SAIL project): the first deals with Cloud Networking (see Chapter 5), while the second challenges the OConS operation from an Information Centric Networking (ICN) perspective (see Chapter 6).

The OConS mechanisms are grouped into three different levels (see also Figure 3.4); this distribution is done according to the particular focus of the mechanisms, as discussed below:

- **Flow Connectivity Mechanisms** are flow and session specific, either end-to-end or edge-to-edge; they are composed of routing and transport mechanisms, and they depend on the corresponding application.
- **Network Connectivity Mechanisms** are node specific services composed of routing and transport mechanisms, which are in this case independent from the current application. They involve two or more nodes, spanning over one or several hops (e.g., end-, access- or core-nodes).
- **Link Connectivity Mechanisms** are specific to particular links, not spanning more than one-hop. They are composed by OConS mechanisms typically implemented in the PHY or

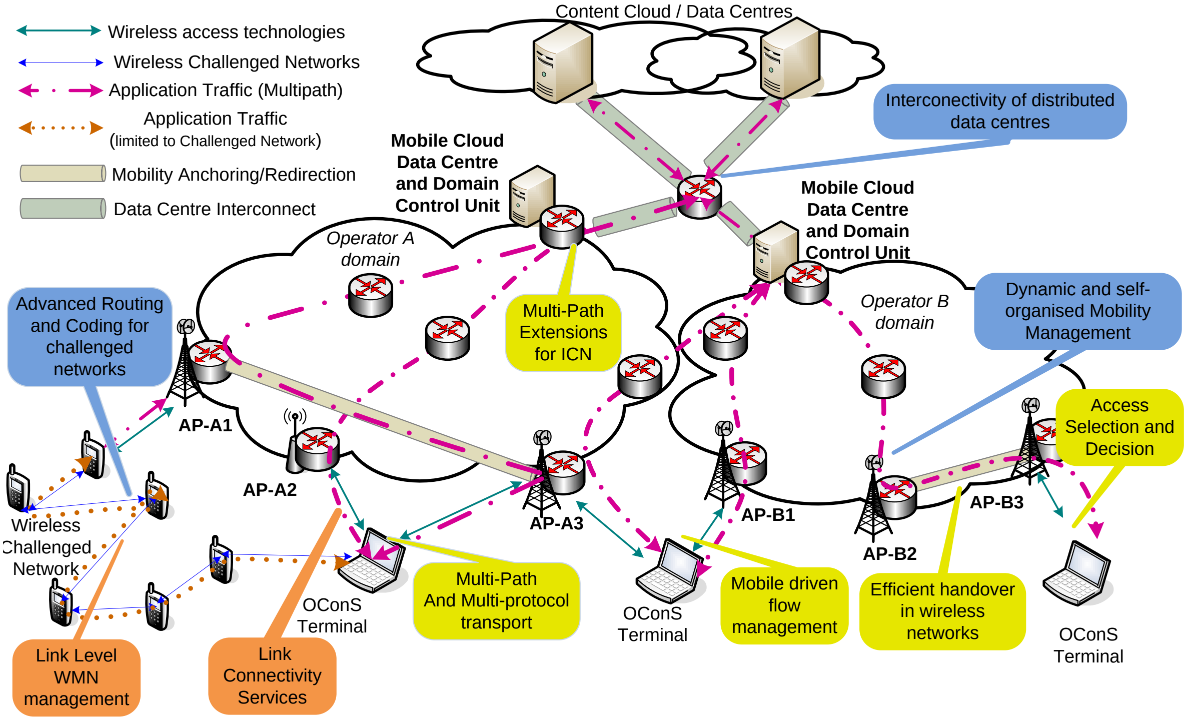

Wireless access technologies
Wireless Challenged Networks
Application Traffic (Multipath)
Application Traffic (limited to Challenged Network)
Mobility Anchoring/Redirection
Data Centre Interconnect
Content Cloud / Data Centres
Interconnectivity of distributed data centres
Mobile Cloud Data Centre and Domain Control Unit
Mobile Cloud Data Centre and Domain Control Unit
Operator A domain
Operator B domain
Dynamic and self-organised Mobility Management
Advanced Routing and Coding for challenged networks
Multi-Path Extensions for ICN
Access Selection and Decision
AP-A1
AP-A2
AP-A3
AP-B1
AP-B2
AP-B3
Wireless Challenged Network
OConS Terminal
Multi-Path And Multi-protocol transport
OConS Terminal
Mobile driven flow management
Efficient handover in wireless networks
OConS Terminal
Link Level WMN management
Link Connectivity Services

Figure 4.1: Scenario which highlights the location of various OConS mechanisms.

data-link layers.

It is worth highlighting that **Resource Management** can be seen as an underlying support function needed by OConS mechanisms.

Additionally, within the scope of OConS, we have also developed a set of **benchmarking algorithms**. These algorithms are not executed over the real-time OConS network infrastructure, and thus, are considered as OConS mechanisms not to be orchestrated. Rather, they are optimisation studies that are executed over a simulated or experimental network, in order to gain better understanding, guidelines, and benchmarks for the specific problems researched. The results of the benchmarking algorithms are used as meta data by the OConS orchestration logic, to better select and configure the OConS mechanisms for optimized performance. The benchmarking algorithms studies and their relevance to the orchestration logic are presented and discussed in section 4.5.

As previously said, providing a complete list of connectivity mechanisms goes beyond the scope of this document; the reader might refer to extensive literature available. In the following we enumerate a number of them, which are of particular interest within the scope of the SAIL project, since they bring about some benefits from the use cases which were selected as the most relevant ones: Cloud Networking and Network of Information. All of them have been investigated within the scope of this workpackage. Later sections in this report (as well as the corresponding annexes) will depict their operation; the reader might refer to [3] and the references therein for a more thorough discussion on their performance results. Figure 4.1, which was already introduced in [1, 2], places some of them within a reference scenario, which will be later used to streamline the discussion of the aforementioned two use cases. Below we enumerate the complete set of mechanisms which have been analysed, details being presented in Sections 4.2, 4.3 and 4.4.

- **Flow Connectivity Mechanisms**
  - **Multi-path extensions for Information-Centric Networks**. This mechanism selects the most appropriate multi-path strategy to be used specifically for Information-Centric Networks, identifying the best paths to forward content along them. Based on

the selected strategy, it guarantees either aggregation of bandwidth or reliable delivery of content.

  - **Multi-path and multi-protocol transport**. Splitting flows into different paths (and possibly using different protocols) might bring about enhancements on flow congestion and load balancing between heterogeneous systems.
  - **Access selection and decision**. They are intended to bring about enhanced QoS/QoE, based on context and particular requirements from the applications, users and network operators/providers.
  - **Mobile-driven flow management**. This mechanism distributes application flows between the most appropriate network paths, so as to offer a better performance to the end-user.[1]
  - **Efficient handover in wireless networks**. This mechanism was conceived so as to improve the performance of end-to-end flows (in particular TCP) by adjusting handover events.

  Further details regarding these Flow connectivity mechanisms are available in Section 4.2.

- **Network Connectivity Mechanisms**
  - **Interconnectivity of distributed data centres**. The mechanism performs resource and path selection and forwarding in (mobile) cloud data centres, aiming at optimal resource (link and CPU load, transport delay) use; in addition we have also analysed the use of overlays and advanced control-plane mechanisms (such as address resolution mechanisms) to interconnect distant parts of distributed data centres over Wide Area Networks (WANs)/OConS domains.
  - **Dynamic and self-organized mobility management**. It combines, on a distributed way, network and host based mobility management procedures. We have also studied the optimization of mobility-related parameters to reduce signalling overhead.
  - **Advanced Resource Management over Heterogeneous Access Networks.** We studied a load-balancing scheme based on pricing strategies and we also carried out a benchmarking analysis to obtain the optimum strategy and the performance which might be expected if using it.
  - **Advanced routing and coding for challenged networks**. We analysed the possibility of using social information to enhance routing in Delay Tolerant Networkings (DTNs) as well as Network Coding (NC) schemes to improve the performance which can be achieved over DTNs and Wireless Mesh Networks (WMNs).

  Further details regarding these Network connectivity mechanisms are available in Section 4.3.

- **Link Connectivity Mechanisms**
  - **Dynamic radio resource allocation for virtual connectivity**. This mechanisms manages the allocation of radio resources to provide the capacity requested for virtual connectivity.
  - **Radio resource management for wireless mesh networks**. By means of advanced management procedures, it brings about a fair throughput distribution amongst all flows.
  - **Spectrum sensing**. It supports the wireless attachment to an Access Point (AP) with a more reliable band selection based on collaborative sensing information collected by wireless nodes.
  - **Improved channel allocation in advanced networks**. The goal is to provide the most appropriate Modulation and Coding Scheme (MCS), which indirectly contributes to a better QoS as perceived by the end-user.

  Further details regarding these Link connectivity mechanisms are available in Section 4.4.

[1] It does not necessarily split a single flow into different paths, as it is the case of multi-path.

4.2 Flow Connectivity Mechanisms

The mechanisms which are presented herewith are applied at an end-to-end flow and they do depend on the ongoing applications; OConS uses these mechanisms either separately or in different combinations to provide the appropriate Flow Connectivity Services. Special attention is paid to the possibility of exploiting multi-path mechanisms as well as enhanced access selection procedures.

4.2.1 Multi-path extensions for Information Centric Networks

The use of multi-path extensions for Information Centric Networks enables the simultaneous use of the multiple attachments an ICN node has to the network. In contrast to multipath in other networks, in ICNs the content is requested (by interest or get messages) from the sink. This means also the multipath strategy has to be decided at the sink by distributing the request messages on the different interfaces (attachments). The key goal which pursued is to increase both the throughput and the reliability in content delivery.

The following three strategies[2] can be used so as to achieve the aforementioned goal (see also Annex A.1).

- Distribution Strategy: Rules are set to transfer multiple content downloads over multiple attachments.
- Splitting Strategy: It distributes the retrieval of one content stream between the multiple attachments.
- Replication Strategy: It uses a replica of the request for the retrieval of content, which is sent to the multiple attachments to increase reliability.

4.2.2 Enhanced access selection and end-to-end mobility

Due to the wide spectrum of access technologies, an important aspect to analyse within the access selection algorithms is the large number of parameters, according to the current communication context, which can be modified to bring about a better network performance. They can be classified in a twofold way: static parameters (different policies and preferences from either the user or the network) and dynamic ones (applications requirements, network load, etc.). OConS enables the possibility to combine different approaches (which could conflict with each other) when taking the decision of which base station to connect to.

By using pieces of information which are monitored by remote elements, it becomes possible to take better decisions about the network accesses and paths to be selected after a connectivity request. This also allows to address the multi-homed flow management problem (MFM), much more adequately (see Annex A.4). In short, the related scenario assumes the presence of multiple flows to several destinations, and the availability of different access networks and technologies. The idea underneath is to select the most appropriate access network(s), distributing the different flows in order to optimize some metrics. One particularly relevant criterion in the case of a mobile user, is the quality that the user perceives from its network use (QoE) which is mapped on network QoS requirements and which is, also, combined with network access costs and battery consumption so as to cover all relevant user criteria.

In particular, for the case of integrating heterogeneous access networks, the question still remains whether the QoS demands of user applications can be satisfied by QoS-unaware non-3GPP access technologies. Within the context of OConS, the effects of the integration of two network types on user QoE in both downlink and uplink directions is investigated by proposing two novel resource estimation and management algorithms (see Annex A.5).

[2] Investigated, implemented, and shown in a demonstrator presented at the MONAMI Conference, September 2012

Under the umbrella of providing the end user with improved and seamless mobility, one particular problem arises in upcoming Long Term Evolution (LTE) network deployments. There is a significant increase of handovers in LTE access technology, due to the use of small cells. One of the challenges consist in how to establish appropriate forwarding mechanisms between evolved Node Bs (eNBs). Annex A.6 describes in detail an analysis of this situation so as to improve TCP performance when used as a bearer of video streaming services.

4.3 Network Connectivity Mechanisms

We refer to mechanisms which, albeit being independent from a particular application, they span beyond one single hop. We focus on (i) procedures related to the interconnection of data-centres; on (ii) enhanced mobility mechanisms, able to instantiate tunnels just when they are needed, reducing the corresponding overhead; and on (iii) advanced routing and coding schemes applied over DTNs and WMNs. Accordingly, the OConS system instantiates one or several of these mechanisms to offer optimal Network Connectivity Services.

4.3.1 Interconnectivity of Distributed Data Centers

The management of the connectivity between distributed data-centres is a rather difficult and complex process today. We select here several aspects of this problem area, which we describe as explicit OConS network level mechanisms to highlight the potential of OConS concepts. Further (standard) legacy mechanisms may be integrated as well without the need of being described here.

The OConS Distributed Data Center WAN Interconnectivity Mechanism (DDC-WIM) allows the management of the connectivity and processing resources within one domain by the Domain Control Units, as detailed in Annex A.7. In short, the responsible control entity of a single OConS domain, collects the measurements like current link loads and load of CPUs of a local data centre and stores it. Then it also performs the path computation (and establishment) from a data centre to the Core Network (CN) or to another data centre either upon request or on the fly. The control entity monitors the utilization of the networking and processing resources along these paths. In case an overload situation in the processing path is detected, it either initiates a redirection of the involved paths or sets up an additional path over less loaded processing nodes in data centres that are served by the same domain. All this is set up and managed by cooperating and distributed servers, as OConS entities that control the resources assigned to the respective mobile cloud data centre resources.

Another aspect which we also explore is the possibility to use OpenFlow as a solution to interconnect data centres, taking advantage from the work carried out in the CloNe work-package, as depicted in Annex A.9.

One basic element on any generic communication (for the interconnection of data-centres in particular) is how packets are routed along the core network. In this sense, we have explored policy-based routing enhancements (see Annex A.10), which might be employed to enhance the communication between data-centres.

The Distributed Data Center Address Resolution Mechanism (DDC-ARM) OConS mechanism presented in Annex A.8 manages the control traffic caused by address resolution procedure between interconnected data centers over the WAN. It provides a network level service for the OConS user (the data center as Customer Edge (CE)) that continues to use usual Layer2 Address Resolution Protocol (ARP) procedures to find the physical MAC address for a given (private, data center/cloud internal) IP address.

4.3.2 Dynamic and optimizing mobility management procedures

Regarding the dynamic on demand behaviour on the access part of the network, the mechanism addressed in [13] (and Annex A.11) focuses on providing dynamic distributed mobility management including per-session handover-decisions, and per-session anchor selection and activation. Currently, the mobile-capable terminals are mostly anchored to the same node, usually centralized and placed deeper within the core networks. In order to optimize the network behaviour (e.g., even for devices that do actually move), new paradigms for mobility anchors location and selection have been considered. In OConS view, the optimal balance between host-centric and network-centric decision points can be dynamically obtained for each session and depending on a given communication context (i.e., resources, requirements, policies). Likewise, for the execution part, we are minimising the maintenance of unnecessary traffic encapsulation, mobility anchors and mobility-related context; thus, the anchor node can be activated and changed only when a device moves, keeping the anchor closer to the terminals to enhance the performances for end-users and also to increase network efficiency. More specifically, the distribution and the dynamic activation of mobility management functions aims at overcoming several issues, such as: the bottlenecks in centralized core networks mobility management entities, the maintenance of unnecessary traffic encapsulation and user's mobility context, or the additional end-to-end traffic delays caused by cascading hierarchical mobility anchors and/or traffic tunnelling functions. For that purpose, the OConS provides a flexible mobility approach, considering both mobility decision and execution functions in a distributed and session based approach.

Besides, it can be said that, generally, mobile users attached to the network share the same settings for mobility management procedures, in particular all users share the same static, pre-configured set of parameters, without taking into account their mobility patterns, i.e. the frequency of cell and tracking area changes in a certain amount of time, for example a normal workday. The disadvantage is that users generate the same amount of signalling messages, related to mobility management, independently on how frequently the mobile device moves and create unnecessary signalling overhead. Integrating OConS entities on network nodes, a centralized controller can use the OConS interfaces to retrieve useful statistical information about network state and behavioural information about users' mobility. It can then use this information to dynamically control the mobility management on a per-user basis, in order to simplify the mobility management procedures and reduce the overhead signalling due to the mobility. This new mechanism that has been studied, named Mobility parameters optimization (MPO), implements a dynamic optimization of the mobility management procedures currently used in the Evolved Packet Core (EPC), as standardized by 3GPP. The annex A.12 recalls some concepts defined by 3GPP in EPC (see [14], [15]), in order to understand the benefits of MPO.

MPO focuses on the 'low mobility' devices: i.e. those devices that don't move frequently (for example: workers that move only twice a day between home and work, 4G USB keys placed in a PC as a replacement for a DSL subscription, sensors, cameras etc.). To enable MPO to identify 'low mobility users', there are two approaches: the user herself selects a 'low-mobility' profile during her/his subscription; or the network has the capability of identifying user behaviours, based on information stored at network nodes. This second alternative is largely the preferred one, since it is seamlessly: MPO collects information from the network about mobility events generated by users and performs statistical analysis on them, to find recurrent mobility patterns.

4.3.3 Advanced routing and coding schemes over DTNs and Wireless Mesh Networks

Challenged networks are the extreme example of dynamic on-demand behaviour required by the network. Wireless mesh and opportunistic networks commonly lack of human interaction awareness, which could help the routing decisions in such challenged environments. That is the key benefit brought by OConS DTN routing: incorporating information on people's social routines into the routing decision. Taking a probabilistic approach as a basis, the work described in A.13 utilises the history of contacts among mobile DTN nodes to prioritize the different available multi-hop routes for a desired destination. The HUman Routines optimise Routing (HURRy) protocol implemented within OConS improves the route taking decision by processing and combining a neighbour profile of surrounding nodes. Basically, the information gathered in the neighbour profile regards to the the history of inter-contact and contact duration values of previous physical encounters. The mechanism also incorporates a tuning parameter that allows the user or application service to prioritize a specific parameter if need be.

This OConS HURRy protocol collects and exploits the same basic information as the NC mechanism for M-to-N transmissions in DTN (see A.14).

Although it has been theoretically shown that NC techniques allow increasing throughput over WMNs, the combination of NC and TCP does not necessarily always yield the expected high gains. Picking the right flows to 'mix' is crucial to achieve the required performance enhancement. We have carried out a simulation-based study of the improvements brought by NC to TCP when used over WMN [16] (see also Annex A.15). The proposed mechanism, integrated with OConS, works as follows [17]: when the request of connectivity is received, if the use of NC is suitable, a decision-making entity will be in charge of choosing the set of packets to be coded together for maximizing the possibilities to have a correct decoding at the destination node. For achieving this goal, information, such as which packets have been already received at the destination, are used so as to select those with which the original packet can be coded.

4.4 Link Connectivity Mechanisms

This section discusses the main aspects of the mechanisms which have been analysed belonging to the radio link level, i.e. they manage connectivity within the scope of a single hop. Those include the management of virtualized resources, advanced control of Wireless Mesh Networks' operation, collaborative spectrum sensing and channel allocation to improve Modulation and Coding Scheme configuration. Based on these various link-related mechanisms, OConS is also able to provide the most suitable Link Connectivity Services, especially in the challenging wireless environments.

4.4.1 Dynamic Radio Resource Allocation for Virtual Connectivity

The virtualization of the wireless access as integral part of Virtual Network (VNet) is a challenging problem, since in wireless networks the changes in capacity/availability of radio resources, due to the inherently limited capacity, may affect the achievement of VNet contracted requirements. A VNet Radio Resource Allocation (VRRA) mechanism is proposed to address the provision of requested capacity (data rate) for virtual connectivity over wireless heterogeneous networks, maintaining the isolation among the virtual networks. VRRA allocates radio resources adaptively and cooperatively, from different Radio Access Technologies, to the virtual resources in order to achieve the VNet requirements. The mechanism is able to instantiate or (re)configure itself upon receiving new connectivity requirements, e.g., QoS type for the virtual resource, capacity or delay. VRRA relies on certain hierarchical structure, since it is integrated within a cluster manager, which is responsible to manage a given set of Base Stations (BSs), which, in addition, perform some particular mechanisms and instruct the allocation of resources to the corresponding link schedulers. This is very well handling by the control introduced by OConS architectural framework concerning the possibility of easily (re)configure the mechanism and the usage of the involved communication capabilities of the OConS nodes. The two algorithms proposed for VRRA are described in Annex A.16. A first one performing a pre-allocation of radio resources according to the capacity request and second one with a more on demand approach doing the allocation as a function of the capacity utilisation.

4.4.2 Radio Resource Management Mechanism for Multi-Radio Wireless Mesh Networks

WMN are an efficient and low-cost solution for providing last-mile broadband Internet access in areas without fixed infrastructure. They face numerous challenges related to intrinsic characteristics of the multi-hop environment and flow of traffic. A novel OConS mechanism is proposed for multi-radio WMNs, for the unified management of tightly interdependent radio resources, such as channels, bit rates and transmission power levels of multi-radio Mesh Access Points (MAPs). It is a hierarchical-distributed strategy, combining rate adaptation, power control, and channel assignment mechanisms to efficiently guarantee max-min fair capacity to every node. The typical fat-tree distribution of traffic in mesh networks, where traffic flows between a gateway and aggregating access points, reached through mesh nodes multi-hop ramifications, is explored. A maximum capacity is allocated to gateway nodes, being reduced as links ramify. This enables to efficiently minimise transmitted power levels, reducing interference ranges and making possible channel reutilisation. A flow control mechanism is also integrated to guarantee max-min fair share of capacity to all nodes. A radio agnostic abstraction-layer is proposed, between the Network and Data-Link layers, enabling to control and operate multiple radios on a multi-radio MAP, supported by the OConS architectural framework. It is detailed in Annex A.17.

4.4.3 Lower layer/physical connectivity services

Spectrum sensing techniques are quite promising; recent work has proven that soft-decision techniques might outperform traditional binary approaches, by linearly combining the unprocessed spectrum energy measurements, captured by cognitive radio nodes, using a set of configurable (heuristically-optimized) coefficients. In this sense, the main objective of the OConS mechanism described in Annex A.18 has been to assess the benefits of optimized linear collaborative multi-band spectrum sensing in cognitive radio networks with respect to its non-optimized counterpart. Such an optimization hinges on maximizing the aggregate throughput while keeping the interference at each sub-band below a certain threshold.

Similarly, we have also looked at channel allocation schemes for advanced wireless networks (see Annex A.19); in particular we have concentrated on Orthogonal Frequency-Division Multiple Access (OFDMA) networks, in which the MCS is adjusted for every transmitted frame, according to the wireless channel condition of the intended receiver. When the channel condition is good, a more efficient MCS can be used. However, when the channel condition deteriorates, a more robust and less efficient MCS is appropriate.

To help the BS determine the appropriate MCS, every Mobile Station (MS) measures and sends Channel Quality Information (CQI) to the BS. The BS allocates a CQI channel for every active MS. The CQI bandwidth is a scarce resource, whose allocation must be adjusted to the actual needs of the MSs. However, allocations and de-allocations of CQI channels require expensive signalling messages between the BS and each of the MSs, and therefore should be minimised. The goal is to improve efficient allocation and bandwidth utilisation of the CQI channel, for each active MS in an OFDMA network. The details of our approach and some preliminary results are published in [18].

Some of the problems addressed are the reallocation of a released channel bandwidth and the modification of the CQI channel bandwidth as a consequence of new mobility patterns.

4.5 Benchmarking Algorithms

The benchmarking algorithms are optimisation or experimental studies that are executed over a simulated or experimental network, in order to gain better understanding, guidelines, and benchmarks for the specific problems researched. The results of the benchmarking algorithms are used as meta data by the OConS orchestration logic, to better select and configure the OConS mechanisms for optimized performance, and thus better fulfil the needs of the applications/users and the status of the network. Within the scope of research of OConS, the following benchmarking evaluations have been done:

- **Benchmarking of distributed mobility management schemes**: evaluation of advantages and drawbacks of the different distributed mobility management approaches, compared to the use of the well known Mobile IP protocol, with or without route optimization option (see Annex A.11).
- **Multi-path benchmarking: trade-off between control plane load and data plane efficiency**: A multi-path benchmarking supports the decision taking in whether a multi-path transport is relevant, weighting the trade-off between control-plane load and data-plane efficiency, thus offering an increased bandwidth and throughput to end-users and applications (see Annex A.20).
- **Multi-Path and Multi-Protocol Transport for Enhanced Congestion Control**: study of Concurrent Multi-path Transfer extension of the Stream Control Transport Protocol (CMT-SCTP) which highlights the scenarios and parameters allowing for higher performance and those which do not (see Annex A.2). This provides guidelines for the orchestration functions as to when such a multi-path transport is relevant to use, or when it is dispensable.
- **Resource Management within Heterogeneous Access Networks**. This study (see Annex A.21) uses *Game Theory* to establish the optimum strategies for resource allocation and pricing over heterogeneous networks.
- **Policy-based routing benchmarking**: Benchmarking for policy-based routing overlay in Core Network with overlay routing nodes, enabling shortest path routing and valid path concatenation (see Annex A.10).

4.6 Combining Several OConS Mechanisms

The evaluation work carried out so far for the OConS mechanisms shows that every mechanism is (individually) able to bring about improvements as compared to the legacy approaches. It is sensible to think that an appropriate combination of them would leverage a yet better operation. In this sense, we advocate that an appropriate mixture of OConS mechanisms in a coordinated manner shall be extremely powerful, since they should be able to benefit from each other. Thanks to the orchestration functionality (see Section 3.7), the availability of one mechanism brings advantages to the others, thereby providing connectivity services that are significantly stronger than just the sum of the involved mechanisms involved. We refer to a combination of OConS mechanisms as an OConS service. Below we provide some illustrative examples of potential combinations which might be of interest.

- The availability of multi-path mechanisms opens the opportunity for multi-homing, thereby significantly enhancing the access selection mechanisms. The outcome of this combination is better resource utilization and improved performance (higher throughput, lower latencies, improved QoE). Likewise, if multi-path is available, the access selection procedures would also use it so as to promote a better performance.
- A connection to a particular access network has certainly clear implications on the related mobility management procedures. In many cases the solutions do not properly work together.

OConS utilizes the advanced dynamic mobility scheme, enabling a closer cooperation between these two mechanisms (enhanced access selection and dynamic mobility management).

- The access selection mechanism would also embrace situations in which the end-user might want to use non-conventional networks, such as Wireless Mesh Networks or DTN. Thanks to the OConS functionality, the availability of enhanced routing and/or coding mechanisms would also be leveraged, so as to improve the overall decision procedure
- The OConS data-centre interconnection use case shows how OConS is able to integrate *legacy* network infrastructures with new generation networking technologies like OpenFlow. A further example of innovative mechanisms' combination for the OConS data-centre interconnection use case is the additional use of the Policy-based Routing Enhancement mechanism to reduce the number of hops and thus the end-to-end delay within the network. The combination of these OConS mechanisms facilitates the dynamic resource allocation for virtual resources and provides the elasticity needed in future cloud networks.
- Wireless mesh and opportunistic networks commonly lack human interaction and awareness, which could help the routing decisions in such challenged networks. That is the key benefit that DTN routing based on social routines brings to wireless networks, and it could complement other mechanisms such as the radio resource management, or the enhanced radio access selection in wireless environments
- The OConS DTN routing is based on the same social information as the NC mechanism applied to M-to-N communication. Information about social interactions among mobile nodes could be exploited by NC and combined with opportunistic routing in order to improve reliability and avoid packet retransmissions in intermittently disconnected networks. Preliminary results shown in A.14 suggest that the history of contacts could enhance the delivery ratio of multicast transmissions in challenged environments
- The Spectrum Sensing Techniques guarantee a more comprehensive and reliable radio channel/band selection, based on collaborative spectrum sensing information collected. These techniques are further beneficial when coordinated with access selection, CQI channel allocation in OFDMA networks, or radio resource management for mesh networks, resulting in an optimal radio frequency allocation and management, reducing interference, packet loss and errors that are derived from low Signal-to-Noise Ratio (SNR) at the reception terminal

As can be seen, the rich set of OConS coordinated mechanisms that benefit from each other and are customized for specific types of network or applications, makes OConS a strong proposition for enhanced connectivity services.

5 OConS for CloNe: Data-Centre Interconnection and Seamless Access for Mobile Users

In this chapter, we describe in detail the Data-Centre Interconnection and Seamless Access for Mobile Users use-case. With its two sub use-cases we show how the OConS orchestration works for CloNe and what the benefits are that OConS brings for CloNe. The first sub use-case is focused on the Data-Centre Interconnection, while the second one deals with the Seamless Access for Mobile Users. In both of them we show how the OConS connectivity services are used to improve the networking-related operation and performance in a cloud network environment. The two sub use-cases are further motivated in the next section 5.1; more specifically, the Data-Centre Interconnection sub use-case is detailed in section 5.2 and the Seamless Access for Mobile Users sub use-case in section 5.3.

5.1 Use-Case Story and Motivation

Nowadays current and emerging Internet applications are hosted in large data-centres on geographically dispersed locations. This however requires synchronization of the different data-centres, and therefore an efficient connection between the different locations is necessary.

It has been identified that in current cloud networks it is a very difficult manual task to securely support dynamic network infrastructures inside a cloud, from a cloud to a customer, and in multi-cloud contexts. SAIL deals with developing a complete and flexible architecture with Flash Network Slice (FNS) capabilities, which operates as a reference model for deploying complex applications over heterogeneous virtualized networks providing flexible and elastic cloud resources and services provisioning, access, networking, control and management.

The Data-Centre Interconnection and Seamless Access for Mobile Users use-case deals with two main issues which are the interconnection of the core data-centre and the enhanced wireless access in a heterogeneous environment. In order to achieve that, OConS provides enhanced connectivity services for CloNe.

The Data-Centre Interconnection and Seamless Access for Mobile Users use-case is selected in order to demonstrate the interaction between OConS and CloNe within SAIL. These sub use-cases detail a set of mobile wireless interfaces, access to an overcrowded community and also content distribution services which require a close management of the available resources between both the end-users and the services.

The OConS framework provides information of the involved domains, specifically OConS information elements are used to monitor and collect data of the available paths, between the servers and clients, and information of the load on these paths. Hereby the OConS framework can orchestrate appropriate OConS services to provide the requested connectivity.

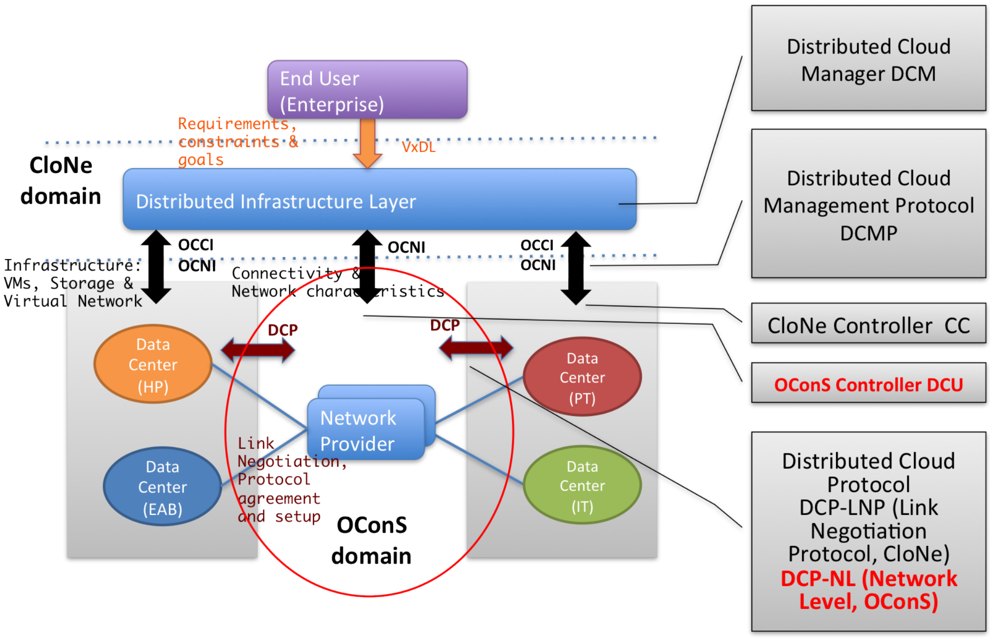


Figure 5.1: CloNe/OConS architecture interfaces

5.2 Data-Centre Interconnect

The Data-Centre Interconnection sub use-case for connecting data-centres with specialized processing and for managing connectivity between these data-centres shows the support of dynamic cloud aspects. The main goal on this sub use-case is to show load-dependent flow and processing control in a multi-processor network, demonstrated by means of multiple video streams. Thereby the OConS framework as an open and adaptative framework is able to model this sub-use-case.

This chapter introduces different connectivity services which are integrated within the OConS framework. Moreover the techniques to achieve these connectivity services and the means to improve their performance and flexibility are depicted.

Data-centre operators use several geographically dispersed locations for performance, load sharing and resilience reasons and thus not only require synchronisation of their different data-centres, but also an efficient connection between the different locations.

The focus here is the management and control on multi-domain data-centre interconnections with multi-path optimisation capabilities over multiple layers.

In order to enable multi-path transport services on the servers in the data-centres in distributed domains, the proposed solution is based on an OConS node, the so-called Domain Control Unit (DCU), which, in a centralized manner, manages the domain. In particular the DCU controls a set of Domain Control Clients (DCCs) OConS nodes, which are in charge of client switching, monitoring and also handle forwarding and routing tables. Furthermore both OConS node types and their mechanims follow the OConS architecture previously depicted (see Chapter 3). DCCs do not contain a SOP but are finding the DCU either by preconfiguration or by a kind of broadcasting to search for the DCU, e.g. by enhancing the Dynamic Host Configuration Protocol (DHCP) to let it tell the DCC nodes about which DCU to contact and register with. The DCU contains a SOP that is responsible for orchestrating the Data-Centre Interconnect. Furthermore the DCU contains an OR storing the OConS entities (IE,EE and DE) and OConS mechanisms available to be orchestrated by the SOP.

The embedding of the OConS domain into the CloNe domain is shown in figure 5.1. There also the OConS domain interfaces with the CloNe domain by using the Distributed Cloud Protocol (DCP) are shown. For more information the reader is referred to [19, chapter 3.4.2]. Shortly said the DCP Link Negotiation Protocol (DCP-LNP) (introduced and used in CloNe, [20], sect. 4.3.1.2) organizes the adjacent data plane connectivity and configures the data-centre - network interface (CE-Provider Edge (PE)) at link level. The DCP Network Level (DCP-NL) is introduced here for purposes of OConS-related network aspects between the data-centre (CloNe) domains and the networking (OConS) domains and dynamically instantiates the application flows within OConS. The OSAP is not shown in figure 5.1 but is always the first entry point to request services from the OConS and it is therefore used by CloNe to access the services provided by OConS, cf. Figure 5.4.

In order to set up edge-to-edge paths across multiple domains, both fully distributed peer-to-peer approaches, as well as hierarchically-centralised architectural alternatives, are considered. In the centralised hierarchical solution a DCU parent entity, which coordinates the path computation across its underlying domains is responsible for the inter-domain path computation between the domain border nodes. In a distributed peer-to-peer approach the DCUs of all domains gather external connectivity information and configure optimum paths. With this the optimization of link and path load within and between domains can be covered.

In the following subsections a description of how OConS orchestration mechanisms are used to support the Data-Centre Interconnect is presented. Also the OConS mechanism and basic signalling flow for CloNe-OConS bootstrapping and orchestration and for creating new paths are described. More detailed signalling flows are depicted in Annex A.7.

5.2.1 Basic OConS mechanisms used in the Data-Centre Interconnect sub use-case

Now a brief description follows indicating which basic OConS mechanisms are used in the Data-Centre Interconnect sub use-case to offer a service.

The overall reference model and control architecture for the Data-Centre Interconnect sub use-case is depicted in [1, Figure 5.3]. It also shows the placement of the OConS entities on the controller-side DCU and within the distributed DCCs.

DCU is used as the name of an OConS node with control and orchestration function typically for a complete single OConS domain. In particular, it integrates the SOP and the OR to select appropriate mechanisms and store their state information, and the main DE for the mechanisms control on a domain basis. It also provides an OSAP as entry point for the service requests from its CloNe user, the Distributed Cloud Manager (DCM) entity.

DCCs denote OConS nodes that comprise the three OConS entities (DE, IE, EE) for the involved OConS mechanisms, like flow/path establishment and modification, flow/path and resource monitoring, and the local forwarding decisions and executions. In addition, the DCC at the border of the OConS domain (’PE1’ and ’PE2’ in Figures 5.2 and 5.3) acts as the provider edge (data-plane) interface corresponding to the customer edge of the data-centre.

The DCU interacts closely with the DCCs via OConS integrated INC functionality and the node-external OConS interfaces to collect intra domain information, which is sent from the DCC IEs to the corresponding DE at the DCU.

If necessary, the DCU also interacts with neighbouring OConS domains’ DCUs to exchange abstract inter-domain information, e.g. on processing resources available remotely or network load between such processing resources and the edge nodes.

After orchestration, the DCU works as follows: the appropriate information is collected by the corresponding IEs of the intra-domain DCCs or relayed from the neighbouring DCUs; they are sent to the DE of the DCU, which establishes and acquires the appropriate and available processing and networking resources. Several attribute values, e.g. CPU resources and link resources currently available or network QoS, energy consumption and price can be used by the resource allocation algorithm.

When the network load state changes, signalled e.g. by sending an 'alarm' from a preconfigured IE, the DCU (in particular, the DE) reacts accordingly, thereby offloading processing in overload or flash crowd scenarios and providing a better data-path efficiency, service, QoS or other target the operator has specified. This ensures an uninterrupted service for the users and also an optimal usage of the operators network resources.

This way, the network link load is influencing the instantiation of processing resources within a data-centre and, at the same time, is assuring service stability.

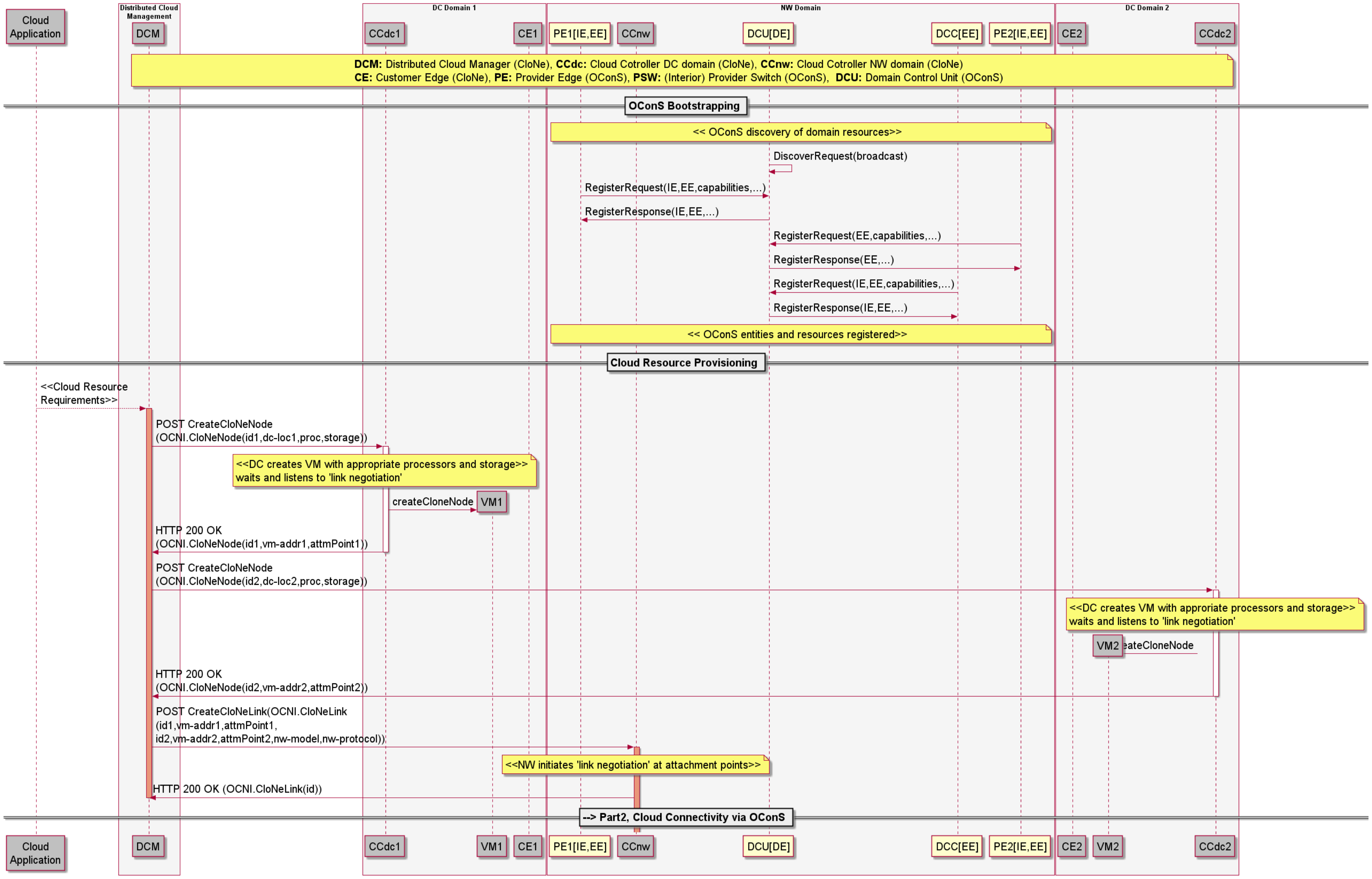

Cloud Application
Distributed Cloud Management
DCM
DC Domain 1
CCdc1
CE1
NW Domain
PE1[IE,EE]
CCnw
DCU[DE]
DCC[EE]
PE2[IE,EE]
DC Domain 2
CE2
CCdc2
DCM: Distributed Cloud Manager (CloNe), CCdc: Cloud Cotroller DC domain (CloNe), CCnw: Cloud Cotroller NW domain (CloNe)
CE: Customer Edge (CloNe), PE: Provider Edge (OConS), PSW: (Interior) Provider Switch (OConS), DCU: Domain Control Unit (OConS)
OConS Bootstrapping
<< OConS discovery of domain resources>>
DiscoverRequest(broadcast)
RegisterRequest(IE,EE,capabilities,...)
RegisterResponse(IE,EE,...)
RegisterRequest(EE,capabilities,...)
RegisterResponse(EE,...)
RegisterRequest(IE,EE,capabilities,...)
RegisterResponse(IE,EE,...)
<< OConS entities and resources registered>>
Cloud Resource Provisioning
<<Cloud Resource Requirements>>
POST CreateCloNeNode (OCNI.CloNeNode(id1,dc-loc1,proc,storage))
<<DC creates VM with appropriate processors and storage>> waits and listens to 'link negotiation'
createCloneNode
VM1
HTTP 200 OK (OCNI.CloNeNode(id1,vm-addr1,attmPoint1))
POST CreateCloNeNode (OCNI.CloNeNode(id2,dc-loc2,proc,storage))
<<DC creates VM with approriate processors and storage>> waits and listens to 'link negotiation'
VM2
eateCloneNode
HTTP 200 OK (OCNI.CloNeNode(id2,vm-addr2,attmPoint2))
POST CreateCloNeLink(OCNI.CloNeLink (id1,vm-addr1,attmPoint1, id2,vm-addr2,attmPoint2,nw-model,nw-protocol))
<<NW initiates 'link negotiation' at attachment points>>
HTTP 200 OK (OCNI.CloNeLink(id))
--> Part2, Cloud Connectivity via OConS
Cloud Application
DCM
CCdc1
VM1
CE1
PE1[IE,EE]
CCnw
DCU[DE]
DCC[EE]
PE2[IE,EE]
CE2
VM2
CCdc2

Figure 5.2: CloNe-OConS Bootstrapping and Orchestration Phase (scenario with 2 DC domains), Part1

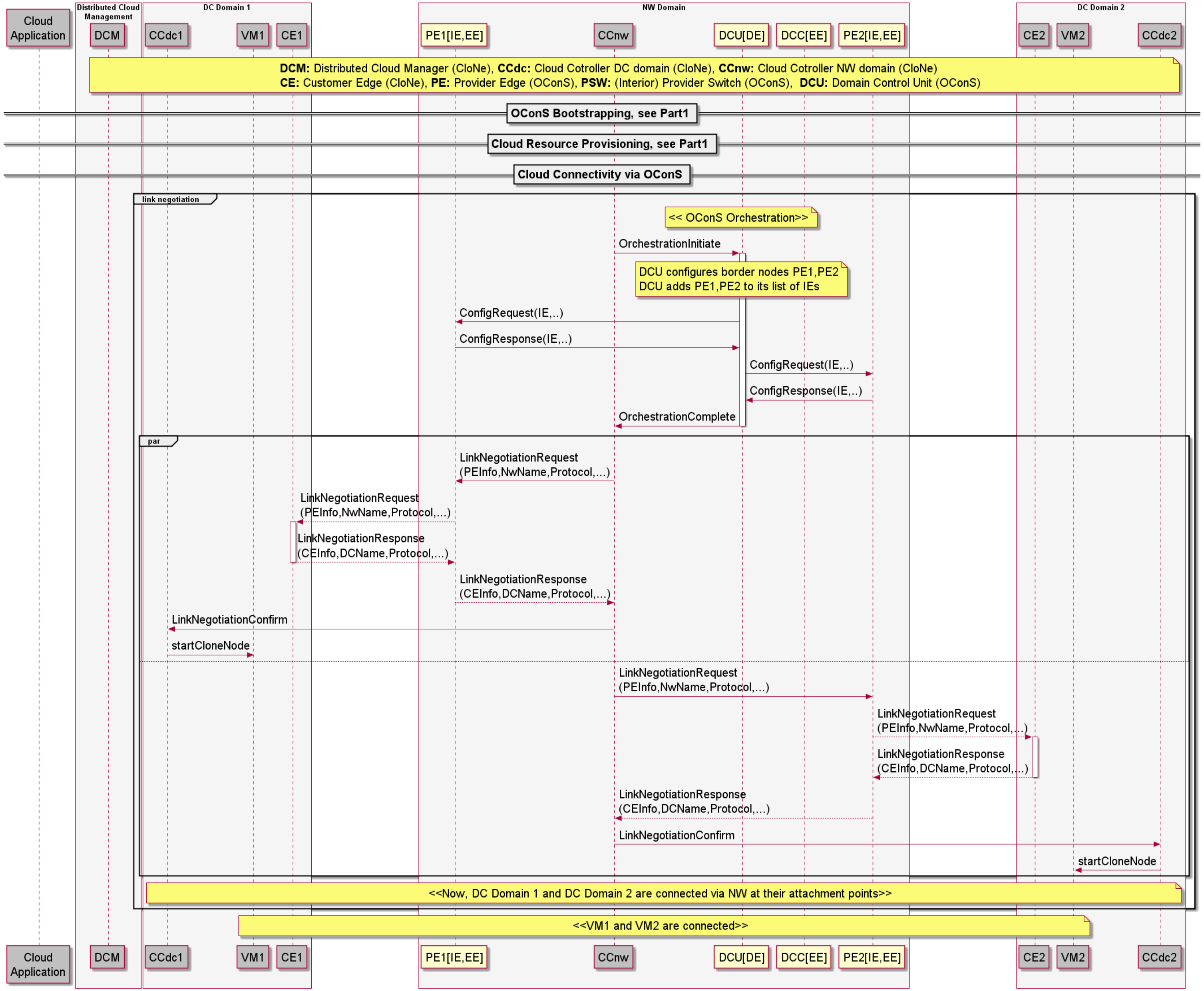

Cloud Application
Distributed Cloud Management
DCM
DC Domain 1
CCdc1
VM1
CE1
NW Domain
PE1[IE,EE]
CCnw
DCU[DE]
DCC[EE]
PE2[IE,EE]
DC Domain 2
CE2
VM2
CCdc2
DCM: Distributed Cloud Manager (CloNe), CCdc: Cloud Cotroller DC domain (CloNe), CCnw: Cloud Cotroller NW domain (CloNe)
CE: Customer Edge (CloNe), PE: Provider Edge (OConS), PSW: (Interior) Provider Switch (OConS), DCU: Domain Control Unit (OConS)
OConS Bootstrapping, see Part1
Cloud Resource Provisioning, see Part1
Cloud Connectivity via OConS
link negotiation
<< OConS Orchestration>>
OrchestrationInitiate
DCU configures border nodes PE1,PE2
DCU adds PE1,PE2 to its list of IEs
ConfigRequest(IE,..)
ConfigResponse(IE,..)
ConfigRequest(IE,..)
ConfigResponse(IE,..)
OrchestrationComplete
par
LinkNegotiationRequest
(PEInfo,NwName,Protocol,...)
LinkNegotiationRequest
(PEInfo,NwName,Protocol,...)
LinkNegotiationResponse
(CEInfo,DCName,Protocol,...)
LinkNegotiationResponse
(CEInfo,DCName,Protocol,...)
LinkNegotiationConfirm
startCloneNode
LinkNegotiationRequest
(PEInfo,NwName,Protocol,...)
LinkNegotiationRequest
(PEInfo,NwName,Protocol,...)
LinkNegotiationResponse
(CEInfo,DCName,Protocol,...)
LinkNegotiationResponse
(CEInfo,DCName,Protocol,...)
LinkNegotiationConfirm
startCloneNode
<<Now, DC Domain 1 and DC Domain 2 are connected via NW at their attachment points>>
<<VM1 and VM2 are connected>>
Cloud Application
DCM
CCdc1
VM1
CE1
PE1[IE,EE]
CCnw
DCU[DE]
DCC[EE]
PE2[IE,EE]
CE2
VM2
CCdc2

Figure 5.3: CloNe-OConS Bootstrapping and Orchestration Phase (scenario with 2 DC domains), Part2

Now the details are given how the OConS orchestration is used to support the Data-Centre Interconnect sub use-case. The OConS orchestration functionalities (cf. Section 3.3) using the Orchestration Registry (OR) and the Orchestration Monitoring service of the Service Orchestration Process (SOP) support the Data-Centre Interconnection sub use-case as follows. After the OConS Node internal entities have been coordinated (see Fig. 3.10) the SOPs of other OConS nodes must be discovered. Thereby for scalability reasons not all OConS nodes have to be aware of all mechanisms of all other OConS nodes, but only those nodes that have a specific mechanism to offer for a certain service register themselves with the SOP. After that, all available network resources and capabilities have to be discovered and identified and then have to be made available to the DCUs node internal SOP. The discovery of a single resource, e.g. an accelerator board or a general purpose Central Processing Unit (CPU) can run in parallel. The DCU can then start the mechanisms (in the DEs) that control the used resources and assign them to incoming service requests accordingly. Then the DCC IE monitors the resources for their usage, e.g. for link, storage and CPU usage. Based on this monitoring in a future system thus, for example, a smart access control mechanism and an intelligent load mechanism could be used to improve the system response and resource utilization.

The effort and the data needed to run the OConS orchestration for the DDC-WIM mechanism is as follows. The data-centre DCU collects and identifies the available resources in its domain. The DCU also monitors current resource usage data using the OConS entities described above. The resource data can be the CPU usage for each processing board, the storage usage on each board, the link usage between boards and even between different data-centre locations. Thereby the current bandwidth used, the delay, jitter and the current error rate on the link can be of interest for assigning optimal resources. In addition, the DCU can react accordingly when application traffic changes, i.e. when user processes enter or leave the data-centre, thanks to the configuration which is carried out at the corresponding IE. Also the network operator can prepare the data-centre in advance in case of an expected flash crowd, e.g. a soccer game or the like. This, of course, not only needs a control interface between the DCU and the resources in the network, but also a network management interface between the data-centre and the operations and maintenance centre.

Up to now by using OR, SOP and OSAP we have depicted how the OConS orchestration supports the DDC-WIM mechanism, which in turm is responsible to find the processing resources and connectivity resources that are available to the DCU. Also after having collected the required data, the DCU has to continuously update its current state of the processing and connectivity resources according to their usage. Thus we can argue that in average less processing resources will be used from the data-centre as in today networks. Furthermore, the OConS orchestration functionalities (cf. Section 3.3) allow us the handling of a big flash crowd of users and thus accounts for the elasticity of a data-centre application.

5.2.2 Basic signalling flow for setting up new paths

Before paths can be set up and handled, the OConS bootstrapping and OConS orchestration procedure - cf. Figure 5.2 and Figure 5.3 - must be completed. Note that in both Figures the DCU comprises a DE, an OR and a SOP, but for sake of brevitiy in those fictures these details are omitted.

In this procedure first the OConS bootstrapping within the network domain takes place. Then the cloud resources within the data-centre domains are provisioned via an interface called by the DCM from within CloNe. After that, the cloud connectivity is configured via OConS, which means that the data-centres and their Virtual Machine (VM)s are connected via the network at their attachment points but no explicit paths resources are allocated inside the network at that moment. This is done either implicitly (by sending a data packet) or explicitly (by a management system) (cf. Figure 5.4) when data transport is needed. Note that OConS nodes capable of monitoring the whole network domain or parts of it, like e.g. the IEs of the DCC are already registered at the

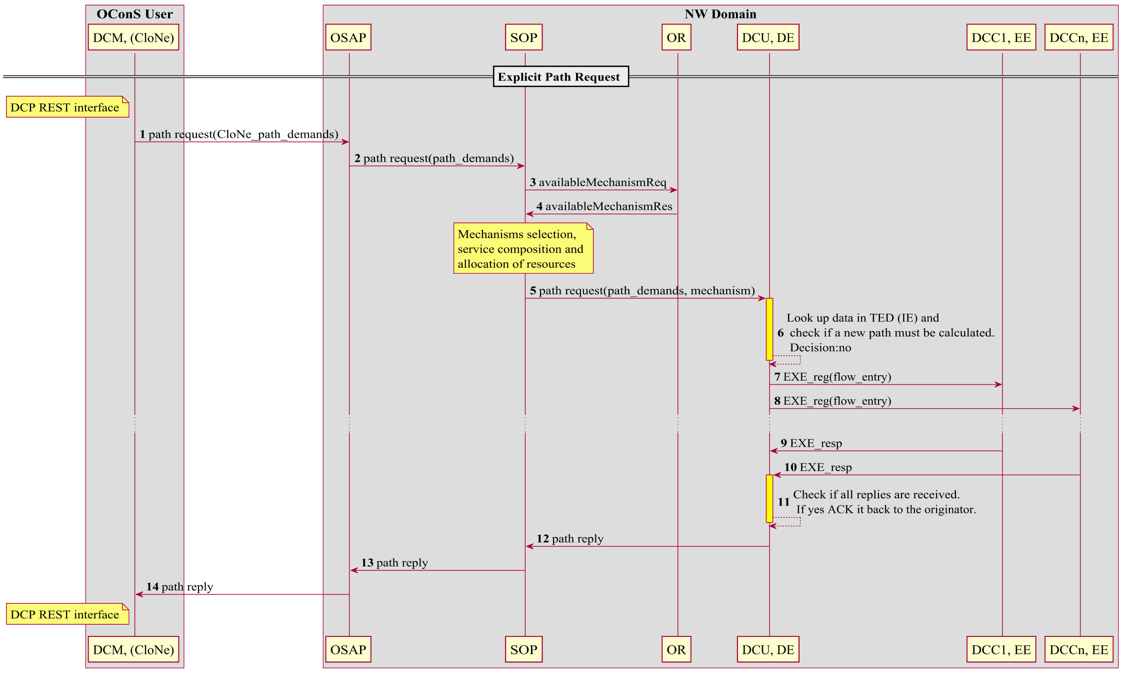

OConS User
NW Domain
DCM, (CloNe)
OSAP
SOP
OR
DCU, DE
DCC1, EE
DCCn, EE
Explicit Path Request
DCP REST interface
1 path request(CloNe_path_demands)
2 path request(path_demands)
3 availableMechanismReq
4 availableMechanismRes
Mechanisms selection, service composition and allocation of resources
5 path request(path_demands, mechanism)
6 Look up data in TED (IE) and check if a new path must be calculated. Decision:no
7 EXE_reg(flow_entry)
8 EXE_reg(flow_entry)
9 EXE_resp
10 EXE_resp
11 Check if all replies are received. If yes ACK it back to the originator.
12 path reply
13 path reply
14 path reply
DCP REST interface
DCM, (CloNe)
OSAP
SOP
OR
DCU, DE
DCC1, EE
DCCn, EE

Figure 5.4: Explicit Path Request Message flow in OConS

respective OR and can thus deliver the needed information about the state of the network domain at any time.

When the OConS bootstrapping and OConS orchestration procedure in Figure 5.2 and 5.3 has been successful and all OConS nodes and their entities, as well as their functions are available, this also means the DCU DE has already collected certain data inside its Traffic Engineering Database (TED) IE and also the OConS monitoring function of the network is activated. Thus the DCU DE can react promptly when new paths are requested or when the network monitoring detects an overload situation on a used link.

As an example the process of setting up an explicit path is depicted in Fig. 5.4. The explicit path request comes from CloNes DCM via a RESTful interface. The conveyed CloNe_path_demands contain the source and destination node address together with path requirements like maximum delay, throughput, jitter and cost. The CloNe_path_demands first arrive at the OSAP in form of a Demand Profile (see 3.5.3), and OSAP translates them into an OConS specific connectivity request that is forwarded to the SOP.

According to the registered mechanisms in the OR, the SOP selects the most appropriate mechanism to be used and initializes them in and from the DCU. For example if the current network state does not allow to use a single path to fulfil the request then a multipath mechanism may be selected. In addition e.g. the Policy-based Routing Enhancement mechanism can be used to reduce the number of hops and thus the delay. According to the path demands, the DCU DE decides whether or not a new path must be calculated. The DCU DE in the end sends the path to its DCCs that establish the needed forwarding entries. The procedure finishes with a response and if everything went right, the DCM in CloNe gets an acknowledge.

Moreover when the path request from the CloNe DCM has been carried out and the corresponding nodes have been instructed to install that path, then also the monitoring function inside these nodes is instructed to monitor the path and inform the DCU if the requirements, such as maximum delay or other service demands, cannot be kept any more.

5.2.3 The Benefits of Orchestration

In the Data-Centre Interconnect sub use-case dynamic cloud aspects are supported by the open and adaptative OConS framework as follows:

- The DDC-WIM decides whether or not a new path must be calculated by dynamically selecting the best paths according to the current network state, i.e. the state of the links and the state of the processing resources the overall cloud service is improved.
- In addition the Policy-based Routing Enhancement can be used to reduce the number of hops and thus the delay.
- By use of the DDC-ARM the signalling traffic between data-centres can be reduced considerably.
- If the current network state does not allow to use a single path to fulfill a connection request then a multipath mechanism may be added.

All in all the combination of these OConS mechanisms allows the dynamic resource allocation for virtual resources and thus provides the elasticity needed in future cloud networks.

5.3 Seamless Access for Mobile Users

The OConS access service can be used to support seamless access to CloNe services for mobile users. The most appropriate access alternative can first be selected in a flexible manner. Then, appropriate flow management mechanisms can be activated on-the-fly to support the user's data flows, taking into account various criteria such as dynamic network conditions, user preferences, required network performance (QoS), acceptable application quality (QoE), resource usage and resultant network costs.

Multiple access selection and flow management mechanisms have been proposed and defined within the OConS framework [1, sec. 6.1.1–6.1.4 and 5.2.7]. Each of them addresses different aspects and results in different decision functions. A user-centric network selection and flow scheduling mechanism which considers user-relevant criteria such as application quality, battery lifetime of the device, price of using the connected networks is proposed in [1, sec. 6.1.3]. A network-based Multi-P transmission mechanism aimed at meeting network-related criteria is presented in[1, sec. 5.2.7]. Its goal is to increase the transmission reliability and capacity and, in turn, to achieve optimized network resource utilization by splitting traffic flows among multiple selected wireless networks. Rather than only considering a single side (user or network), [1, sec. 6.1.1] presents a more generic and flexible access selection mechanism, which allows the decision to be made either at the end-user or at the network-side, by means of combining different user and network related parameters (*e.g.*, policies, preferred operator, service requirements, load of the network) with the possibility of prioritizing some above the others. Finally, the mechanism addressed in [1, sec. 6.1.2] focuses on providing dynamic distributed mobility management including per-flow handover-decisions, and per-flow anchor selection and activation.

Any of these mechanisms can be used, through OConS to provide enhanced connectivity to the CloNe user. The goal of this section is to illustrate how OConS enables dynamic selection and combination of the most appropriate ones, and orchestrate them in order to satisfy the current requirements.

5.3.1 General Scenario

Figure 5.5 presents the sub use-case in which a mobile user requests access to a cloud service in the CloNe data-centres via one or multiple access networks. The mobile user has multiple interfaces to connect to multiple wireless access technologies (*e.g.*, Wi-Fi, 3G, LTE or WiMAX). Various such access networks are available with overlapping coverage areas and thus result in heterogeneous

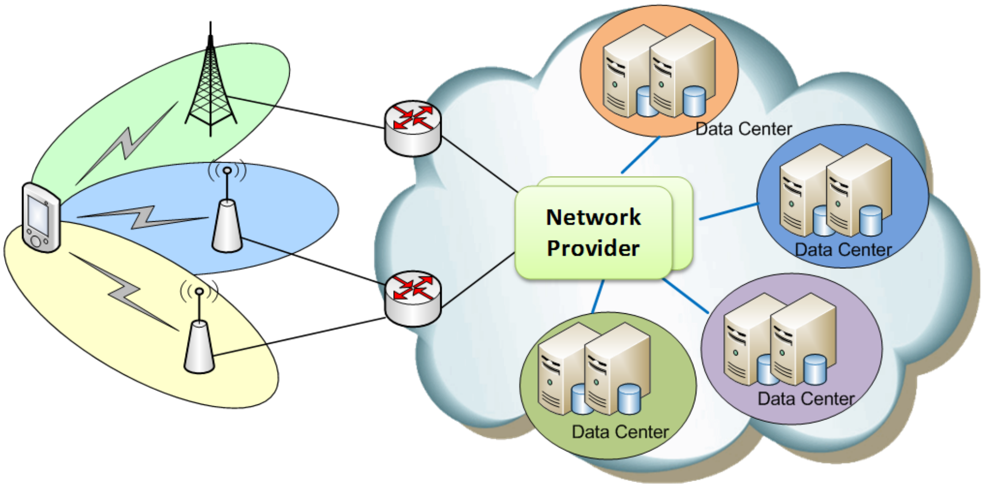


Figure 5.5: CloNe/OConS Seamless Access for Mobile Users

network environment. No assumption is made on whether the access network and data-centre are within the same OConS domain or belong to different stakeholders.

Each user may have a number of different application flows (*e.g.*, video, VoIP or web), each with different QoS requirements or expected perceived (QoE). The availability and network quality of each access network changes over time, depending on the network load (in terms of number of users, traffic load), link quality (*e.g.*, delay or loss) and capacity.

5.3.2 Orchestration Process

In this exemplified sub use-case, the user sends an explicit request for connectivity to the cloud data-centres. The following presents the corresponding orchestration process to dynamically select the most appropriate access selection and flow management mechanisms and combine them to provide the CloNe user a seamless access to the cloud service.

We consider the situation after the bootstrapping phase has successfully concluded. Therefore, the SOPs on all OConS nodes are aware of the available entities. All of those which might be needed to orchestrate the presented mechanisms are assumed to be available. This means, amongst other aspects, that the DE of the access selection mechanism is already aware of the information it might use for taking decisions, and as well the different mobility management protocols/procedures which are present (and their corresponding EEs). Furthermore, the Multi-P mechanism is also available (this can be of relevance, for instance, to decide whether more than one access networks can be used for the same flow).

As known from section 4.6, some of the mechanisms can be combined together to provide an overall improved OConS service. In this particular case, the access selection and Multi-P mechanisms can be naturally combined so as to enhance the utilization of the available network resources and thereby improve the transmission capacity per flow. This mapping can be validated for example by simulation, experience, machine learning based on measurement in the network or on the user equipment. In this example, we take this combination as a given configuration of mechanisms for the OConS, as the fourth case mentioned in section 3.7.4.

The orchestration process is shown in Figure 5.6. As mentioned earlier, it is assumed that the bootstrapping phase has successfully completed, which is also shown in the figure. The OSAP is therefore aware of the possibility of using both an enhanced access selection and the Multi-P procedure. Both procedures are complementary and can thus be activated together in this case.

Upon receiving a connectivity request from the CloNe user, the SOP of the mobile node (MN)

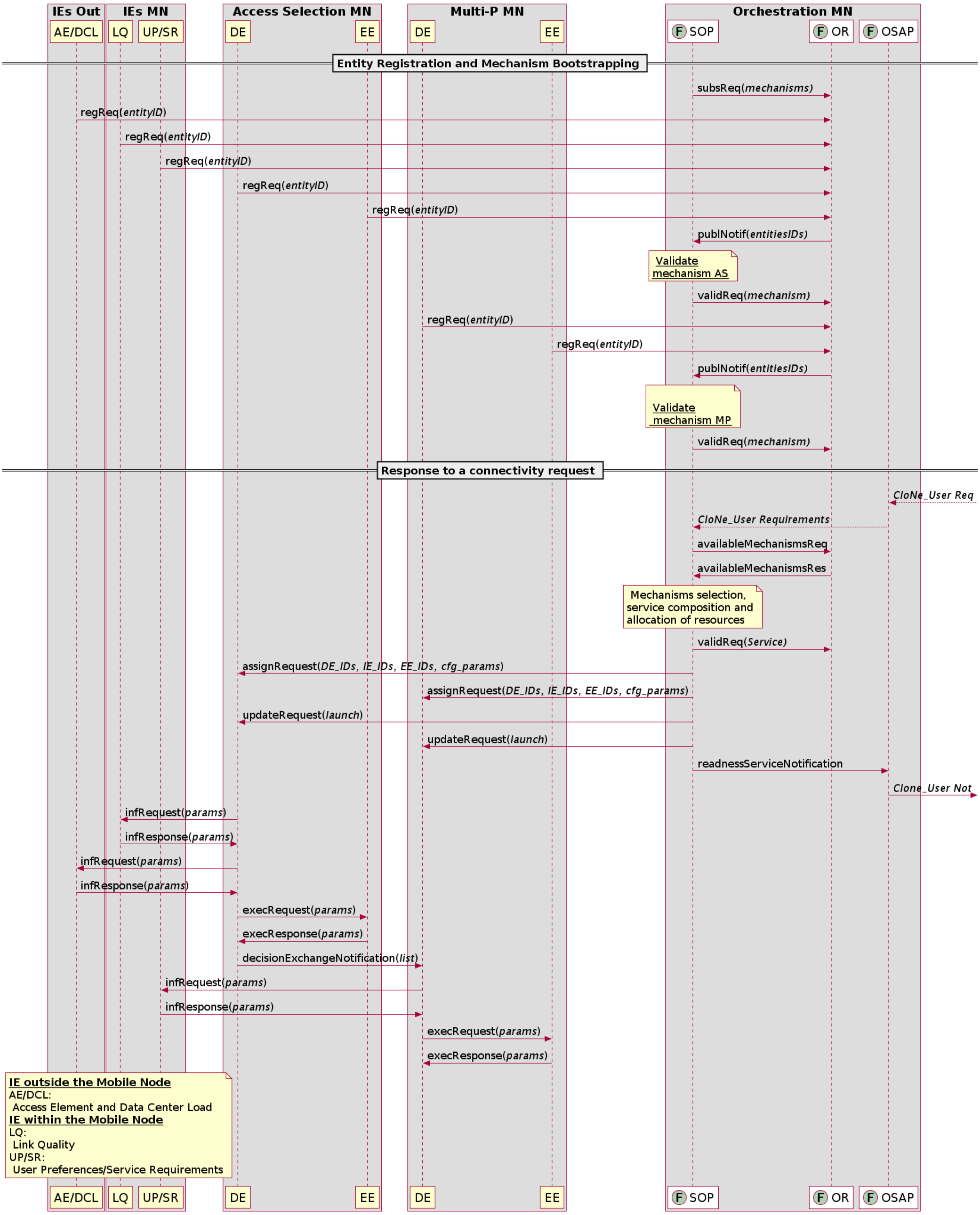


Figure 5.6: Message flow of the interaction procedure to orchestrate a seamless access OConS service.

obtains a set of connectivity requirements from the user demand profile (see 3.5.3). For instance, the CloNe user wants to receive a specific video stream from the cloud, which requires a network supporting a 1 Mbit/s data rate with low delay. The user also prefers low access networks costs. These requirements define the QoS (such as capacity, delay, losses) which shall be provided through the selected access network(s), and their associated costs.

As described in section section 3.7.4, based on the given connection requirements a set of candidate mechanisms need to be firstly selected out of all available ones from the OR. At first, the SOP in the mobile node checks the available OConS access selection mechanisms registered in the OR. Then the SOP decides which access selection mechanism is best suited for the needs and instantiates it. This procedure considers a number of different aspects such as the CloNe user's requirements (*e.g.*, whether any user-centric criteria were given), the current network state and the availability of the mechanisms (*i.e.*, the availability of all the required components as OConS elements). Some other criteria can come under consideration such as whether the user prefers to make the decision on their side or have it off-loaded to the infrastructure. After the selected mechanisms have been launched properly, a notification is also sent to the CloNe user to notify that the OConS service is ready.

To perform the access selection, the chosen access selection mechanism is orchestrated as a combination of the relevant entities. As shown in Figure 5.6, the SOP sends a request to the DE of the chosen access selection mechanism, which in turn requests information from the relevant IEs:

- IEs within the mobile node, which provide the user information, e.g. link quality, user preferences, and service requirements;
- and IEs outside of the mobile node, which offer the information about the current network state, e.g. access network and data-centre loads.

Based on this information, the DE performs the selection and enforces the decision to the corresponding EEs (this might imply, *e.g.*, the related mobility-management operations, locator handling or rendez-vous). Moreover, the decision entity of the access selection mechanism collects a list of selected access networks.

In the next step, the access selection and Multi-P mechanisms will be combined to orchestrate the service following the given configured combination of mechanisms. Then, the list of selected networks made by the access selection DE is then sent to the DE of the Multi-P mechanisms and processed by the Multi-P DE to decide whether the flow should be split or how it should be distributed over the selected networks; in addition, the Multi-P DE also uses information about the user profile and the service requirements. If needed, it could also retrieve some information that has been obtained by the access selection mechanism about the current status of the data centres, or about the end-to-end paths to be used to reach them.

In this case, the Multi-P DE shall also be aware of the characteristics of the potential paths. Once the two DEs have finally made a decision, the user can start receiving the flow through the established paths. Monitoring of these paths for changing conditions can also be instantiated by the DE instructing the corresponding IEs to periodically report performance and operation metrics for the selected configuration. In Figure 5.6, we can see how the Access Selection DE, after successfully enforcing its decision informs the Multi-P DE of the list of accesses which can be used, and triggers this second level round of decision.

It is worth mentioning that the previous example is meant to show the possibilities which are brought about by the Orchestration. As such, it is an illustrative operation of the various entities. The OConS framework is flexible enough so as to enable more combinations or possibilities. We have seen that the decision was taken by the end-user in the above example; we could also have situations in which either the access selection or the Multi-P operation is established by the network entities, by collecting different pieces of information.

Another example can be a combination of the access selection and mobility management mechanisms for the CloNe user. In this case, once the access selection mechanism decides which access alternative has been chosen for the current connectivity request, mobility management mechanisms can be used to select and activate the network anchor points. This combination can therefore provide seamless mobility and session continuity to the cloud data-centre throughout network access changes.

The corresponding message flow to orchestrate a seamless handover is shown in Figure 5.7. The entities registration and bootstrapping procedure is same as the above example (Figure 5.6). In this example, the orchestration is triggered by the change of network state, which is monitored by the group of IEs in charge of network monitoring. When the IEs detects that the user's link quality (e.g., SNR) reaches a defined threshold for making a handover, they launch a notification message to the SOP to trigger the handover event which requires re-selection of the new AP with seamless mobility. Thus the SOP will check the available OConS access selection and mobility management (handover) mechanisms that have been registered in the OR.

Then the SOP selects access selection and mobility management mechanisms to be launched, and map the selected two mechanisms with the connectivity requirements, network state into service composition and allocate required resources. Finally the created OConS service is also registered in the OR.

After the above procedure, the DE of the chosen access selection mechanism requests information from the old AP, the new AP and its IEs within the mobile node about "Load, Link Quality, Delay, User Preferences and Service Requirements".

Based on the given information from the different IEs, the DE of Access Selection makes the selection and then sends its decision to the Mobility Management (MM) DE. This MM DE sends a *decisionExchangeNotif* message to the DE of the old and new AP to inform them about it is going to be a disconnection from the former and a connection to the latter. After that, the handover is carried out by the MM DE at the mobile node sending an *exeqRequest* message to the MM EEs of the New AP. At the end, the EEs of the New AP sends an *execRequest* message to the EE of the old AP in case a tunnel is necessary.

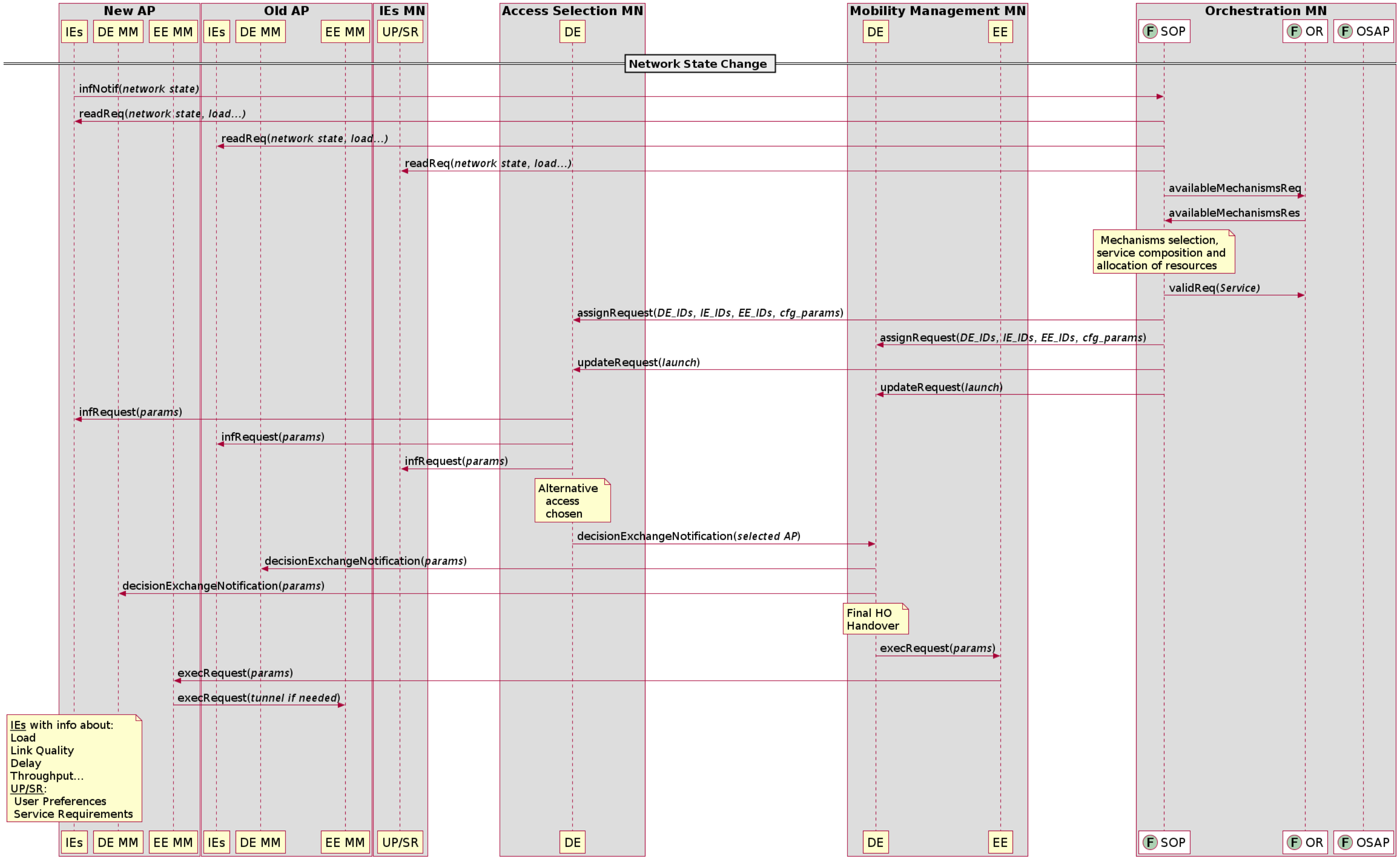


Figure 5.7: Message flow of the interaction procedure to orchestrate a seamless handover(for the sake of brevity, not all the entities registration have been done and the responses have been removed).

5.3.3 The Benefits of Orchestration

In the Seamless Access for Mobile Users sub use-case, a CloNe user can benefit from OConS facilities, i.e., by using different OConS access selection and flow management mechanisms via orchestration. As seen from the above two examples, the main benefits achieved with orchestration are summarized as follows:

- By dynamically selecting the best networks according to the network state changes, the user's connectivity to access to the cloud service is enhanced, e.g. with lowest delay, loss.
- By combining access selection and Multi-P mechanisms through orchestration, the user can use its available network resources much more efficiently and achieve higher capacity by being able to use multiple access networks simultaneously, and thus the user can improve its QoE.
- By integration of enhanced access selection and dynamic mobility management mechanisms through orchestration, the user can achieve better connectivity, and seamless mobility and session continuity to the cloud service.
- By optimizing the network resource usage, the overall network costs can be reduced, which is a gain desired by the network operators.

6 OConS for NetInf: Wireless and Multi-P Support for Information Centric Networks

This section describes in detail the benefits that OConS brings for NetInf, in particular when applying wireless and Multi-P mechanisms, where 'Multi-P' stands for the combination of multi-point, multi-path and multi-protocol. As a common scenario, the Event with Large Crowd (EwLC) scenario has been specifically described in [19] and serves as an illustration of how the orchestration process of OConS can support the requirements from NetInf as a client and accommodate to network conditions dynamically. The EwLC scenario consists of mobile nodes that are connected to the network infrastructure (via a base station) and that have local communication capabilities. Depending on radio link properties, node position, etc. a node can communicate to one or multiple other nodes at the same time. Communication sessions can be short-lived, as nodes can lose contact etc., so that, in general, the crowd provides communication opportunities, but with unpredictable performance. Each node is also connected to a cellular network (infrastructure), but we assume (due to large crowds) that the link to the corresponding base station is overloaded, i.e. heavily congested and not always usable.

6.1 Use-Case Story and Motivation

In this use-case we illustrate how OConS provides connectivity services to Information Centric Networks (in particular to NetInf), creating and sustaining the connectivity in challenged wireless networks while using multi-path enhancements. Imagine that there is an unexpected event happening (e.g. a street performer) and many people stop by, spread the word through social networking, and a flash crowd is spontaneously gathered. Some people record and upload multimedia content to a social network (e.g. Facebook), send around photos and videos, so that people in the crowd or other followers of the social network start to download the content files, as well as background information about the event.

Users employ NetInf nodes caching, and forward the produced content based on their name and locator. In this flash crowd some users have good connectivity, while others experience poor or intermittent connections. The combination and orchestration of several OConS services can improve the connectivity of the flash crowd users, who require a minimum reliability, a minimum QoE, and therefore need optimized resource utilization. More specifically, the message forwarding functionality in each NetInf node needs to be enhanced by OConS. Hence, by collecting network information via the OConS IEs, the DE in the enhanced NetInf message forwarding function can decide to activate and configure the OConS mechanisms using the appropriate EEs.

In this use case the following OConS functions are considered:

- OConS multi-path connectivity services can select the best multi-path strategy (distribution, splitting, and replication) to retrieve content from the mobile devices. The selection and enforcement of these strategies are performed at the participant devices as well as in the network.
- For poorly or intermittently connected users, OConS DTN routing, OConS Mesh Networking, and OConS Network Coding can further improve the NetInf nodes connectivity.

6.2 Connectivity Services for the EwLC Use-Case

According to the description motivated in the previous subsection, we can assume that a NetInf-OConS device is present in the flash crowd event, so that the connectivity support provided by the OConS Orchestration can be shown. The NetInf functionality is supported by one or several OConS services implemented in this end device. The overall goal is to have different connectivity services (Figure 6.1) and different and/or complementary mechanisms available in a same device, so that OConS Orchestration is able to decide which one to select and activate, depending on specific requirements (e.g. NetInf application requirements) or network conditions (e.g. information provided by OConS IEs and processed by Orchestration DE).

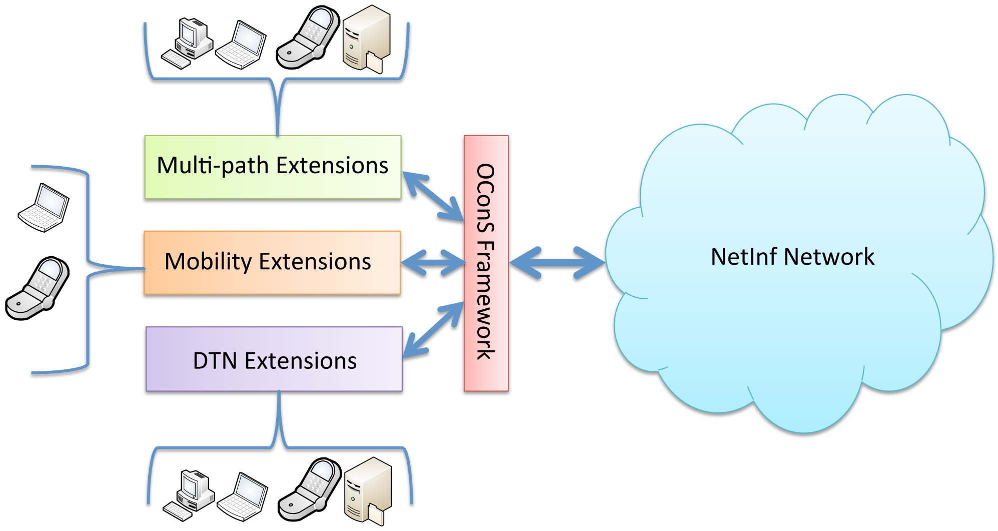


Figure 6.1: NetInf/OConS architecture interfaces

An illustrative story line could be depicted as the following sequence:

- A person present in the flash crowd records a video and shares it via one of his/her profiles in a social network. Some friends see it and like it, and some of them who are near the spot spontaneously decide to join the flash event. One of these friends downloads the video and also shares information of the event (e.g. local pictures) via Bluetooth with surrounding fellows.
- More and more people start sharing content and videos. The NetInf devices employ caches and produce associated names and locators. Eventually, and because the intermediate caches might experiment high volume of demands, the network status could require some traffic balancing and/or flow management mechanism, which OConS Orchestration is able to detect and act upon. In this case, multi-path could be triggered in a seamless way so that NetInf service performance is enhanced and the user experience maintains a good quality.
- If some people do not have Internet access on their mobile devices, and still want to share and retrieve content regarding this flash event, they could benefit from hop-by-hop interactions via DTN connectivity. The OConS Orchestration is aware of the DTN routing mechanism available and can request the contact related information from the IEs, which are collecting data from the history of encounters. Whenever an opportunistic path is available via DTN connectivity, OConS will enforce the content retrieval, benefiting from previous interactions among present devices (maybe friends from a social network who are also present in the spot, or spontaneous intermittent connections with surrounding devices).

We can even think of different transmission requirements if for instance the event organisers would want to distribute some information or official multimedia content to all present people in the crowd. That would also raise an optimization issue for the broadcasting or multicasting communication. OConS Network Coding applied over DTN for M-to-N transmissions is a mechanism aiming at avoiding packet replication, minimizing transmission delays and maximizing throughput as much as possible. This mechanism works in conjunction with the DTN routing, exploiting the history of people's social interactions for the encoding decision taking. Based on the OConS architecture, the NetInf user might benefit from an enhanced access selection, in which different strategies (centralized vs. distributed, network-based vs. user-based) can be used. In the wireless access the user might have different possibilities to connect to. OConS provides support to such process, so that it can be improved by the monitoring and management functionalities of the Orchestration. Briefly described, a full OConS support would comprehend the provision of the specific requested connectivity service by a NetInf application. OConS takes care of the whole connectivity set-up by applying similar functionalities as in the seamless access case described in section 5.3 of the previous chapter.

The orchestration is needed so as to be able to identify, structure, and manage an appropriate collection of information elements which might be valuable for NetInf purposes. In addition orchestration should also span and manage any other mechanisms which might be of interest, like DTN routing, multi-path, access selection or network coding, for instance.

6.2.1 Multi-path Content Delivery in Information Centric Networks

The mechanism on multi-path content delivery for ICNs focuses on enabling the use of multiple devices that the participants of the flash crowd use to download content. These devices use NetInf as the ICN architecture. The users request some content, available at different sources, and the NetInf nodes will consist of an OConS based forwarding and convergence layers that retrieve content from the identified sources. These OConS components in NetInf establish and select the best multipath strategy to retrieve the content. The identification of this strategy is done using functionality provided by the OConS framework. This functionality includes the use of IEs that feed information to make decisions by DEs which are then implemented by the EEs located in the different NetInf based devices in the network.

These new extensions in NetInf to support the OConS multi-path content delivery is referred to as OConS Multi-path Network of Information (OMPNetInf). The main characteristics of the OMPNetInf mechanism are as follows:

- Mechanism operation - The operation of this mechanism consists of a number of phases in which the environment for retrieving content is set up and, at the end, the content is finally retrieved.
    - Discovery - This phase results in the discovery of the DE for the 'multi-path content retrieval for NetInf' mechanism (see Annex A.1).
    - Registration - The phase results in the discovery of the different entities of OConS that support the operation of this mechanism and in the registration of the corresponding capabilities
    - Forwarding Strategy Enforcement - This phase is where the multi-path strategy is selected and operated to retrieve the required content
- Data Model - The different IEs located in NetInf nodes, feed information to evaluate the rules that select the appropriate multi-path strategy to request and forward content.
- Interfaces - The INC which is part of the OMPNetInf handles the messages that are received and forwards them to the appropriate functionality to be processed. These messages are carried over the NetInf protocol or the underlying transport protocol used by NetInf.

OMPNetInf performs segmented retrieval of content where the NetInf GET messages are used to request for and receive content (Named Data Object (NDO)), [21]. The operation of OMPNetInf involves the controlling of the multi-path capable Convergence Layer (CL) to perform the retrieval of content using the multiple paths that the NetInf node has to the network. Figure 6.2 shows the messages that are passed between the different components in the client node when operating the multi-path mechanism (strategy).

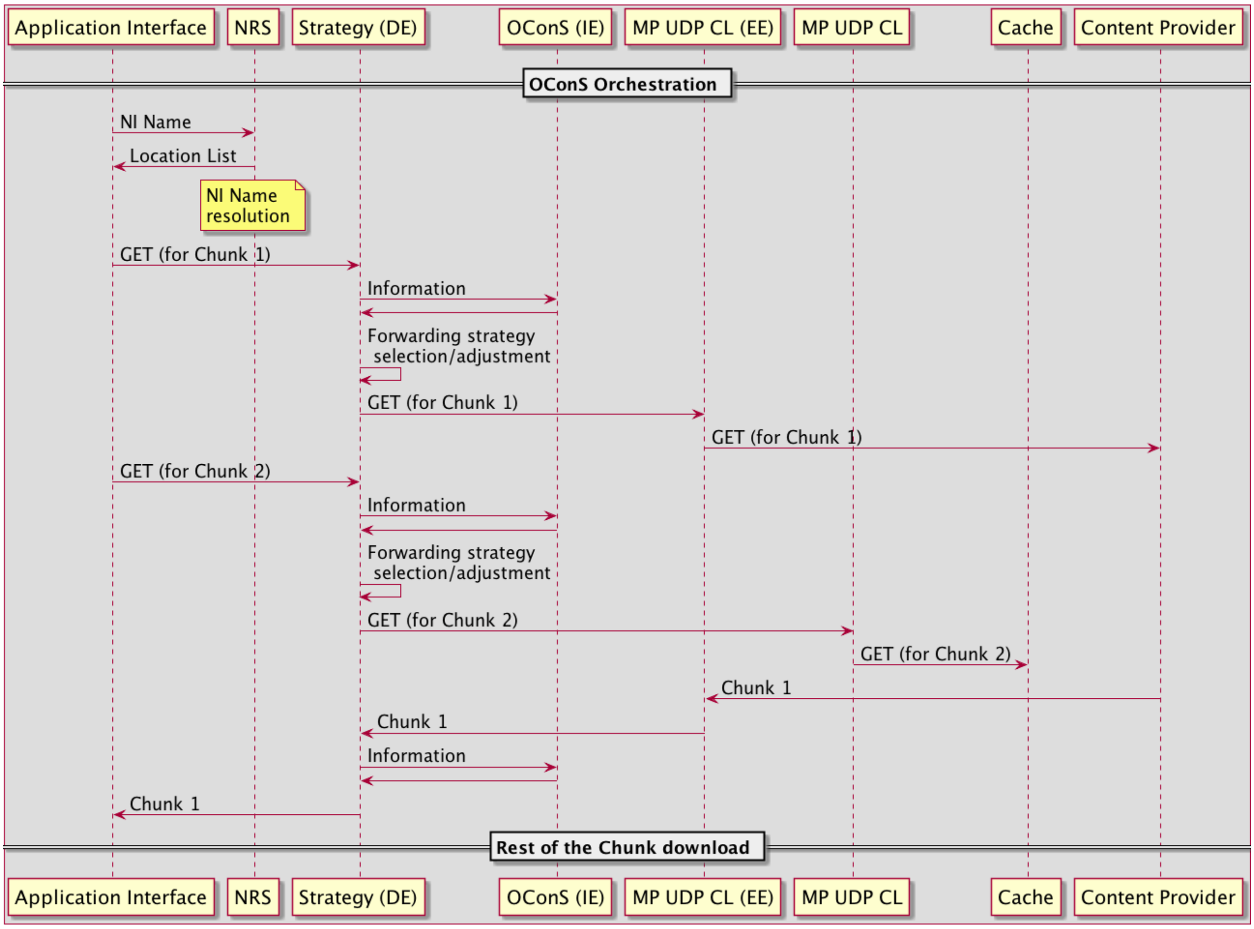


Figure 6.2: Operation of OMPNetInf

The NetInf enabled application requests for a content providing the NetInf Identifier (NI) name. This is resolved by the Name Resolution Service (NRS) to a list of possible locations from which the content can be retrieved. This information is used by the OConS CL to retrieve the content NDO by NDO (referred here as chunks). The OConS CL consists of the application interface module, strategy module, OConS module and the number of Multi-Path (MP) UDP CL instances that are created to handle each of the paths. The application interface module is responsible for generating the NetInf GET messages for the NDOs of the content required. The strategy module performs the selection of the MP UDP CLs to forward the GET messages. It also evaluates the forwarding strategy to use regularly, triggered by the receipts of NDOs and also based on other information provided by the OConS elements in the network.

The basic procedure followed by any NetInf node in the flash crowd, deployed with this mechanism is as follows.

- Applications in the NetInf based device request content
- The NetInf node performs the name resolution to get the locators of the content (resolving the NI name to locators)
- The orchestration mechanisms in OMPNetInf initiates the DEs to identify the multi-path strategy to be adopted

- The convergence layer in OMPNetInf downloads the content based on the adopted strategy. The DEs continually re-evaluates the rules to modify or select a new forwarding strategy.

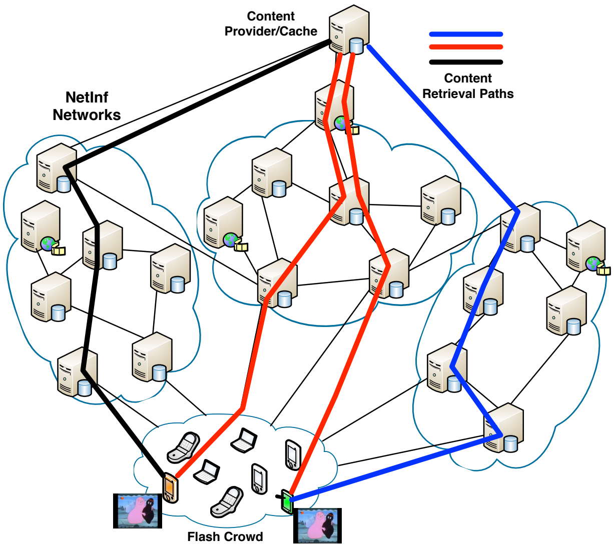


Figure 6.3: Flash crowd participants retrieving content over multiple paths with OMPNetInf

Some participants in the flash crowd are expected to have multiple paths to the NetInf networks. Since these participants attempt their content retrievals over common paths, the quality of the retrievals may degrade. But the operation of OMPNetInf results in the selection of the best multi-path strategy that provides the best performance to each flash crowd participant (Figure 6.3).

6.2.2 Support NetInf performance by OConS DTN service

OConS, thanks to the flexibility brought about by the orchestration process, allows the monitoring and usage of different information to enhance the connectivity of both legacy and innovative mechanisms. In this case, for the enhancement of opportunistic routing strategies, the awareness of people's social routines is introduced, but there might be more examples. This illustrative OConS DTN routing is based on the recent history of social encounters occurred between nodes of the flash crowd. As mentioned before, we assume that some of the people gathered in the crowd have downloaded (or locally produced) some content during the event: information about the event, videos, etc. Within this crowd, some nodes might have not direct access to the Internet, so they could get these pieces of information from neighbouring nodes instead. Also, not every node is permanently connected to every other node in the group, and so there are no permanent routes established among all possible destinations altogether forming a DTN environment. If several people start moving away from the group, they could still spread the content to new neighbours, not locally present in the flash event. DTN routing would resolve the opportunistic path available to retrieve the requested content through hop-by-hop successive interactions.

Figure 6.4 shows a typical scenario, where several nodes are part of a mobile DTN topology and eventually, one user terminal (Node C) might have access to the outside world (i.e. the Internet) through a wireless technology interface (apart from its available DTN physical interface, which might be based on the same or different wireless technology).

Each DTN node is responsible for exchanging information regarding previous encounters and estimated path-ratings with its neighbouring nodes. A probabilistic algorithm will then come up with quantitative rating values for all possible routes. The best next hop (node with the highest rating value, or probability) to reach a certain destination is decided according to a mathematical

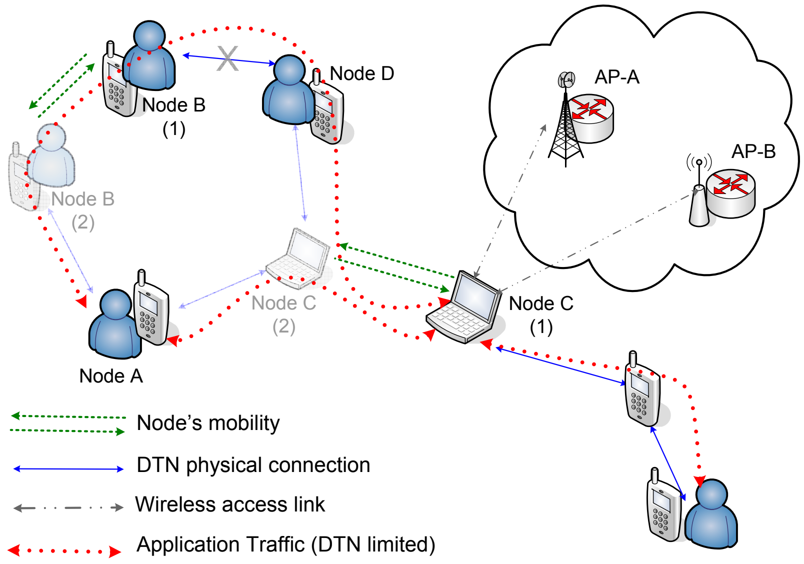


Figure 6.4: Dynamic DTN scenario with diverse wireless connectivity

equation where the accumulated mean values of inter-contact and contact-duration times in historical encounters are considered (see A.13). As an illustrative example, suppose, in Figure 6.4, that Node A must find a route towards Node C to gain access to the outside world. Let us assume that every DTN node in the scenario is aware of the capability of Node C for accessing the Internet. If Node A demands a specific content (e.g. a video from Youtube) from the Internet, it will need to decide how to reach Node C (i.e. decide which intermediate neighbour would most probably contact Node C, or in a more reliable way). DTN nodes are mobile, and they register information about who contacts whom, with which frequency and for how long. In Figure 6.4 nodes B and C are moving back and forth from position (1) to (2), which induces the establishment of new connections (with nodes A and D), and the intermittent disruption of the link between B and D. Once Node A contacts Node B, they exchange information about the probability with which they expect to reach Node C (this value is merely based on their history of contacts with D and C). The nature of these encounters regarding frequency or duration will vary depending on specific features of the scenario considered. Outdoor and indoor topologies might result in very different contacting routines, for instance. In the same way, people do not show the same social behaviour with colleagues during labour days, as with friends during the weekend. Human routines are affected by the surrounding environment and conditions, but they also affect the resulting connections established in a mobile DTN.

In this particular situation, the routing mechanism could be aware of some nodes acting as NetInf caches to spread a popular content related to flash event, and consider that parameter when deciding which is the best next hop for a certain route. To support NetInf operation OConS has developed a DTN service which combines the implementation of the Bundle Protocol Query (BPQ) extension over the DTN protocol suite for Android smart phones. In this way, the local Orchestration SOP would be able to combine both mechanisms to serve NetInf requests in a *GET_req-GET_rsp* manner and using each mobile phone as an intermediate cache.

The Figure 6.5 represents the message sequence chart performed within an OConS node whenever the local SOP receives a connectivity request. In this case, the request is triggered by the NetInf Application Interface, through which a user is demanding a certain content from a public location. After translating the demand profile into connectivity requirements, the local SOP needs to verify that there is a connection available that permits the retrieving of that content. At first, the SOP tries to validate that there is a Third Generation (3G) access available, but there is not such access (it can be due to unavailable interface in the physical device, or because the access to the AP is congested and there is no possibility of gaining attachment at that moment). Secondly, the SOP verifies that the implementation of NetInf-BPQ requests over DTN is available, and then it launches the specific service, combining the required mechanisms, and allocates the associated resources.

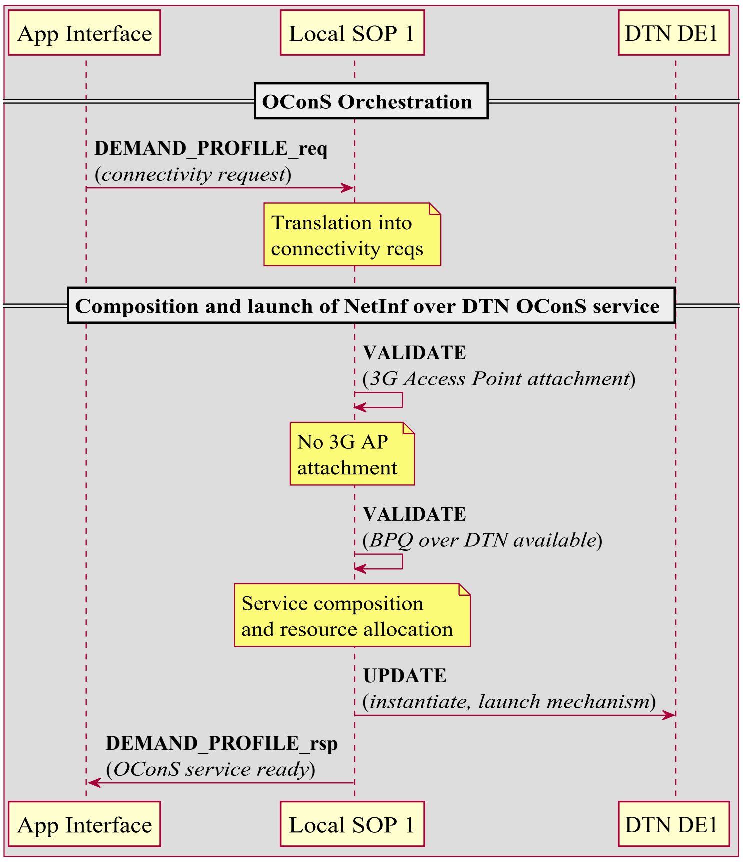


Figure 6.5: Local instantiation of OConS service BPQ over DTN

Accordingly, once the OConS service is ready, the Application Interface sends the message $GET(content, location)$ to the node's DE, starting the common service operation during runtime. The key steps of this sequence and the messages among entities involved is represented in Figure 6.6. The sequence chart represents the message exchange between two neighbour nodes: $Node1$ and $Node2$. $Node1$ receives a GET_req and checks that the destination address is not $localhost$. $Node1$ does not possess the requested content in its local cache, so it forwards the GET_req message to its neighbour $Node2$ (OConS routing mechanism estimates probabilities for best next hop according to configurable policies). $Node2$ possess the requested content in its local cache, so DE2 decides not to forward the GET_req towards the $location$ addressed, since it is not necessary. DE2 enforces the transmission of the requested content to $Node1$ generating a GET_rsp message with the content attached.

In the described scenario NetInf benefits from OConS service for a DTN topology where permanent connectivity cannot be assumed and the attachment of nodes to the network might be heterogeneous and, in some cases, not possible. Using the OConS DTN routing, a NetInf node could improve its QoE in the flash crowd scenario. The request of a specific user to get a certain content (i.e. multimedia archive) would not be trunked or dismissed only because there is no physical path to the destination addressed. The most relevant gain here would be the pos-

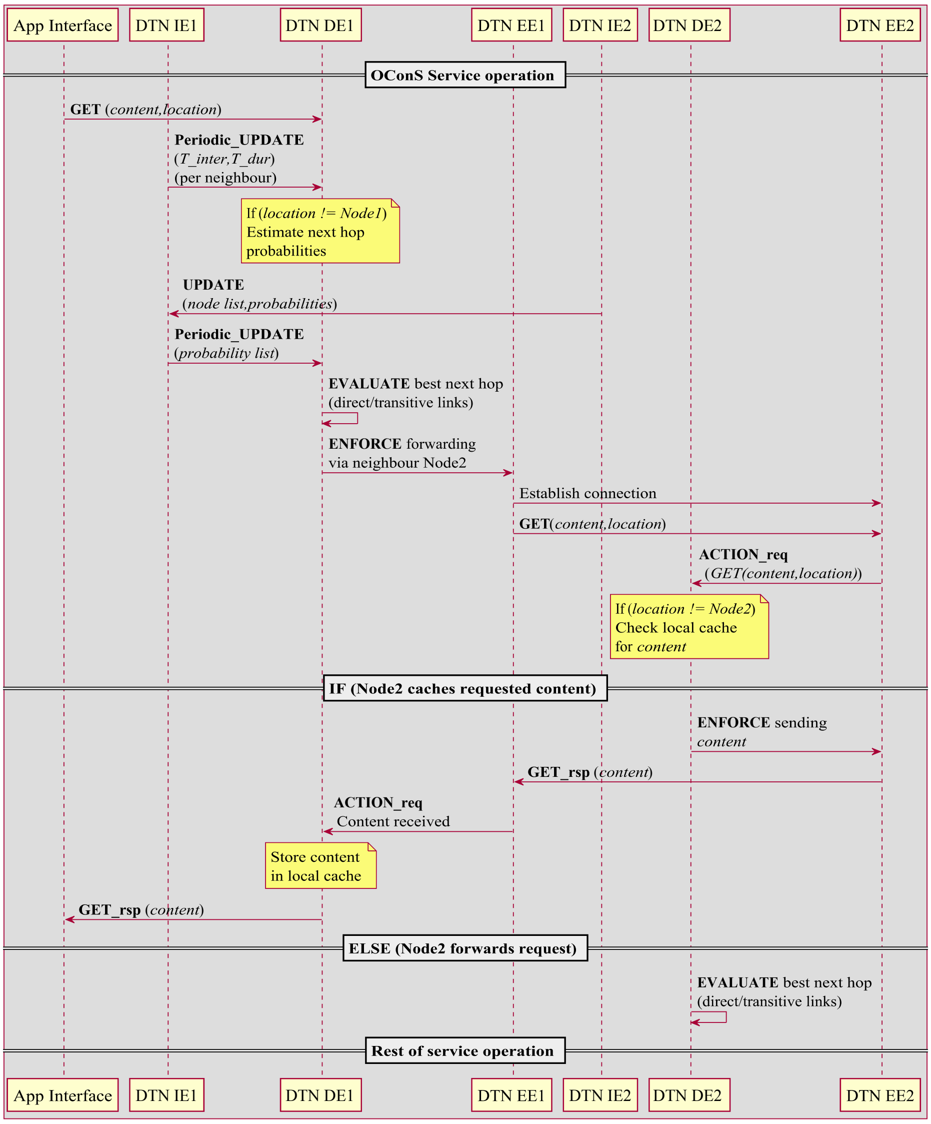

App Interface
DTN IE1
DTN DE1
DTN EE1
DTN IE2
DTN DE2
DTN EE2
OConS Service operation
GET (content,location)
Periodic_UPDATE (T_inter,T_dur) (per neighbour)
If (location != Node1) Estimate next hop probabilities
UPDATE (node list,probabilities)
Periodic_UPDATE (probability list)
EVALUATE best next hop (direct/transitive links)
ENFORCE forwarding via neighbour Node2
Establish connection
GET(content,location)
ACTION_req (GET(content,location))
If (location != Node2) Check local cache for content
IF (Node2 caches requested content)
ENFORCE sending content
GET_rsp (content)
ACTION_req Content received
Store content in local cache
GET_rsp (content)
ELSE (Node2 forwards request)
EVALUATE best next hop (direct/transitive links)
Rest of service operation
App Interface
DTN IE1
DTN DE1
DTN EE1
DTN IE2
DTN DE2
DTN EE2

Figure 6.6: Runtime operation of BPQ over DTN: process *GET_req* via local caching

sibility of combining and instantiating several available mechanisms by the Orchestration SOP: an application specific request (from NetInf application interface) could be served either using a 3G wireless attachment, or a DTN hop-by-hop connection. An optimal ad-hoc solution could be selected according to current network state and each node's available interfaces. Moreover, OConS routing mechanism for DTN provides the benefit of using the history of connectivity patterns among nodes to better predict a best next hop and increase the probability of delivery success for each transmission attempt.

7 Validation and Assessment Planning

The deliverable has up to this point presented the results of the OConS research, namely the framework and its architectural concepts as well as a set of OConS mechanisms and how they are applied. This section discusses the validation and assessment of these results, what aspects are addressed by the evaluation and where results are documented.

The achievements are considered in the context of the project, in relation to CloNe and NetInf, and with respect to the derived scientific contributions. Due to the different nature of the results, the validation will consequently consider different aspects too. While the performance evaluation for the proposed OConS mechanisms indicate qualitative improvements, other criteria are relevant for the OConS architectural framework. Whenever possible, the reader will be directed to further, concrete evaluation results, which are discussed and further elaborated in Deliverables D.C.4 [3] and D.C.5 [5].

7.1 OConS Architectural Framework Evaluation

We view the OConS orchestration as the process to operationally control mechanisms, the OConS architecture as an approach to support the design of OConS orchestration and the OConS architectural framework as the combination of elements that are engaged to define the architecture and corresponding orchestration processes.

There were no indications that OConS entities are not sufficient to design and support the implementation of orchestration processes and their interactions. So far, the OConS entities have been understood as a means to describe a wide range of different connectivity functionalities and as a suitable way to depict the orchestration of the OConS mechanisms. Further validation requires to identify and to assess the ability or limits in the way theses models are helpful to design orchestration solutions. A proof of concept has been already made, showing, as a first step, the implementability and feasibility of the orchestration OConS proposes (subsection 7.2).

The quality of the OConS architectural framework can be evaluated by the capability provided, which translates demand profiles into connectivity requirements. A solution that can dynamically support during execution time the selection of mechanisms, that until now were static, can be seen as an improvement. If such a support is understood as a basic building block, ready made to express (parts of) mechanisms semantics and to support combinations that ease the design of orchestrated mechanisms, then the building blocks would in a sense be comparable to design patterns as used in object-oriented and process improvement design.

An additional topic in the architecture, however, is the support of security that can be already provided in the beginning of the design of new orchestration procedures. Two activities were followed in this field. First, a threat model has been elicited to identify and understand potential vulnerabilities that may appear due to the misuse of the orchestration process (see Appendix E). From an attacker's perspective the opportunities to misuse the orchestration of mechanisms were looked at. Besides, potential privacy violations have been identified. One consequence of this security analysis was to protect the messages and the message exchange of the interaction protocol used between OConS nodes (see also Annex C.4). Further consequence of the findings of the security analysis are left for future work, which will certainly influence the way the OConS architectural framework will be enriched with security functionality.

Overall it would be desirable to evaluate whether the extra effort of creating and using components that implement the orchestration is beneficial, considering the additional control of exiting mechanisms for the sake of re-using or combining parts of existing mechanisms. Some of the saving may actually take effect as the designed and existing solutions can benefit from re-use. This is certainly true for only a subset of existing solutions. However, designers and network engineers see replications of mechanisms and recognize patterns that happen to be developed again and again, such as the support of context information. A meaningful evaluation, however, is beyond the scope of the SAIL Project; such effort was not aimed at, although it seems to be enabled with OConS.

7.2 OConS Orchestration and Proof of Concept Implementation

Focussing on the assessment of OConS orchestration solely, i.e. without building on specific mechanisms that have been elaborated, we have some good evidence that the orchestration can actually happen. To sustain this, the current results of a proof of concept are briefly presented below. Once the benefits of the orchestration become more relevant to the developers, we reckon that the OConS Entities, the OConS SOP, and the OConS Architecture will become of practical use.

So as to assure the feasibility of the proposed orchestration, a proof of concept focused on the entities bootstrapping and mechanism recognition is shown in [22] (more details to be provided in upcoming prototyping-related D.C.5 deliverable [5]). In particular, this test-bed has been used for the access selection mechanism introduced in Section 4.1. The mechanism is composed by five functional entities, which allow a user centric, context aware access selection over a scenario of one user node and two access nodes. The distribution and behaviour of the entities are as follow:

- A DE, set in the user node, which carries out the decision making according to the information gathered by the corresponding IEs.
- An IE, also in the user node, able get the current quality of the available access nodes according to the perceived SNR.
- An IE that provides the user profile information, which enables the decision to be made according to the current user.
- Another IE to provide network state information has been implemented; it offers information about the current traffic load of the different routers behind the access points.
- Finally, the mechanism includes an EE able to perform the connection towards selected access node.

The mechanism has been defined as the aggregation of the aforementioned entities as compulsory, except the IE concerning the traffic load, which has been considered as optional. Through an experimental evaluation the bootstrapping process (as described in Figure 3.10) was proved: first, all the entities locally register towards the SOP, enabling it to perform the discovery of capabilities; next, the mechanism is validated, provided that the compulsory entities are available, to finally enable the DE to make the decision as configured.

Once the basic bootstrapping is succeeded, which means that the user node gets a connection, the orchestration has the ability to extend the mechanism according to the new entities discovered in other nodes, the access elements in this case. At that moment the DE is updated so as to take into account the information provided by the network.

In addition, the proof-of-concept also showed the feasibility of the orchestration of various mechanisms, since it also featured a *Dynamic Mobility Management* module, which was combined with the enhanced access selection mechanism previously depicted (see [22, 5] for further details).

7.3 OConS Mechanisms

Different OConS mechanisms have been developed and validated subsequently. Many of them can be applied in one of the use-cases, OConS for CloNe or for NetInf. Furthermore, the OConS mechanisms can further be applied to other use cases, enhancing several networking fields. To evaluate these individual OConS mechanisms e.g. in terms of performance, different steps were taken. As a first step, each mechanism on its own is evaluated and compared against state of the art in its respective research area. The results and details of this assessment effort per mechanisms is summarized in Deliverable D.C.4 [3]. A major part of such discussion is based on recent publications referenced there (the interested reader might get additional details therein); in some parts, it encompasses performance evaluations which are not yet published.

Secondly, because most of these mechanisms are OConS-compatible i.e., they have commonly agreed interfaces, they can be combined by the orchestration and by this combination we can potentially achieve even better performances. The expected performance gains are briefly introduced here, while a comprehensive description will be further detailed in Deliverable D.C.4 [3].

As a third step, we reckon that when designing new mechanisms, synergies can be achieved by re-using common OConS components such as the IEs to monitor and predict the network states, or by capitalising on the DEs to solve multi-criteria cost functions when optimising the network states; this third step is however beyond the scope of the OConS work within the SAIL project. Therefore, in this section, the performances and advantages of the individual mechanisms are summarized, independently of the use cases described above. Furthermore, the potential benefit of the simultaneous usage of these mechanisms is addressed.

The mechanisms investigated are addressing three different areas: wireless access, data-centre interconnection and mesh network services. OConS mechanisms pertaining to these different areas can of course also be combined using the OConS orchestration procedure.

OConS Mechanisms beneficial in Heterogeneous Wireless Access Environments

If the user is connected via a wireless interface, different mechanisms can improve the user connectivity. The most important is the selection and management of the most suitable access network and technology. Generally, there is no best mechanism, as the user can be in different context and with several access networks available, she potentially can connect to different operators, some might be OConS-enhanced while others might not.

Accordingly, if the network nodes are not OConS-aware, we expect that a user centric access selection is possibly an advantage; yet, if the network side information is available to the OConS mechanisms, this information can be utilized and the decision can be taken either on the user device or on the network. Decisions on the network side can consider the set of users connected or ready to connect and thus they can improve the overall network quality.

For access selection and flow management, different mechanisms have been proposed in the OConS framework. Each of them addresses different aspects and results in different decision functions.

In Annex A.4, a user-centric network selection and flow scheduling mechanism is investigated, considering user-relevant criteria such as application quality, battery lifetime of the device, and the price of using the connected networks.

In Annex A.5, a network-based Multi-P* transmission mechanism is developed for future wireless networks consisting of a mixed heterogeneous 3GPP, *e.g.* LTE, and non-3GPP access technologies, *e.g.* WLAN. The goal is to increase the transmission reliability and capacity and, in turn, to achieve optimized network resource utilization, by splitting traffic flows between multiple selected wireless networks.

Rather than only considering a single side (user or network), a more generic and flexible access selection mechanism is also available (see Annex A.3), which allows the decision to be made either at the end-user or at the network-side, by means of combining and possibly prioritizing different user and network related parameters.

Any of these mechanisms can be used through OConS to provide enhanced connectivity to wireless users. Moreover, taking into consideration the functions and benefits provided by different mechanisms, a number of them can also be combined together to provide an overall improved and more flexible service. For instance, the decision of access selection mechanism can be used by the Multi-P mechanism (*e.g.* to distribute the traffic of the same flow between the selected access networks), and their procedures can collaborate through the OConS orchestration, such as to increase the utilization of the available access networks and thereby improving the transmission capacity.

For the CloNe use case, we have illustrated in Section 5.3 how OConS enables the dynamic selection of the most appropriate access selection mechanism, combining it with flow management or mobility management mechanisms, and orchestrating them in order to provide the CloNe user a seamless access to the cloud service, as well as seamless mobility and session continuity.

Besides selecting and managing the appropriate access, further mechanisms can be activated to improve the quality of service for the user. Firstly, in case of error-prone channels, network coding can be applied, as summarized in Annex A.15; this can be activated by OConS only when needed, using the OConS orchestration function.

In some cases more spectrum can be used if cognitive radio systems are applied, as summarized in Annex A.18. If policies allow, and the users and the access points or base stations are in a suitable area, OConS can activate this mechanism.

Benefits for Data Centre Interconnect with OConS Mechanisms

The management of the connectivity between distributed data-centres is a rather difficult and complex process. The OConS DDC-WIM allows the management of the connectivity and processing resources within one domain by the DCUs, see A.7. Furthermore the OConS architecture allows different OConS mechanisms to be combined, to form a comprehensive and complete service, to control and manage the connectivity between data-centres; this relies on the OConS functional elements, interfaces and information model as specified in this document. Accordingly, as a first step, we showed the advantages of path selection and multipath communication for distributed data-centres. In the activities being demonstrated and prototyped in OConS so far, the mechanisms are mainly restricted to one domain only.

For Interdomain-Routing cases, the Border Gateway Protocol (BGP) protocol is predominantly used today. Using the DCUs within each domain as proposed above, the inter-domain routing could be improved with respect to the BGP protocols by using the OConS mechanisms and protocols to exchange minimum information or to negotiate transport options; findings from the activities summarized in Annex A.10 could be applied here.

The multipath communication for data-centres is also backed by the investigations on efficient multipath for IPTV services as summarized in Annex A.20. Furthermore the overlay mechanisms for data-centres interconnection, see Annex A.9, can thus be applied here.

Improved mesh connectivity by means of OConS Mechanisms

Improving connectivity services in challenged ad-hoc networks is difficult and typically implies handling diverse dynamic requirements, dealing with different physical interfaces and resource optimization (which are expected to be feasible in a few cases only). In OConS, several mechanisms have been investigated and efforts are undertaken to develop a holistic OConS wireless mesh approach. In some cases only one wireless interface can be used, whereas in other cases, several interfaces can be simultaneously used. In the latter case the most appropriate strategy has to be applied; hence, we have investigated the strategies for the NetInf use case as summarized in Annex A.1. The mechanism addressed in Annex A.11 focuses on providing dynamic distributed mobility management including per-session handover-decisions, and per-session anchor selection and activation. In the same line, the mechanism reported in Annex A.17 proposes a radio agnostic abstraction-layer, between the Network and Data-Link layers, enabling the control and operation of multiple radios on a multi-radio MAP. Furthermore, in delay tolerant networks, several improvements can be provided by OConS; firstly, using a history of contacts, e.g. see Annex A.13, a better prediction of whom to forward a message to can be obtained; secondly, also network coding can be utilized in delay tolerant networks, as summarized in Annex A.14.

7.4 OConS in the context of SAIL

OConS has been investigated in search of new answers for solutions that can cope with connectivity demand in ever changing future communication environments. Individual mechanisms that improve connectivity solutions have been addressed. In addition, an approach is followed that ties together such mechanisms and hence orchestrates existing solutions to improve connectivity solutions.

To that end we acknowledge that the connectivity solutions that have been already deployed are legion and that in some situations it would be better to build on what is existing and deployed, as we believe. This leads to wiring simultaneously on both mechanisms and orchestration. The concept of orchestration provides an additional control over existing investments, which we think makes a positive impact on the operation of future communication environments.

Since we took a bottom up approach, lots of mechanisms were developed in parallel. For these we needed an incremental framework specification approach, where successive refinements can be applied. At this stage we are presenting the orchestration in this deliverable in combination with new or improved OConS Mechanisms.

One way to assess OConS is to make use of it within SAIL to support CloNe and NetInf. The use cases that are presented in Chapter 5 and Chapter 6 address this support. Apart from the fact that the use cases describe the effort for the integration of results from the three technical barebones of SAIL, we see a clear benefit in both use cases.

The “Data-Centre Interconnection and Seamless Access for Mobile Users” use case shows the benefits of the combination of several OConS mechanisms like multipath and optimal access selection. Thus, for example, if the current network state does not allow to allocate a single path to fulfil the request, a multipath mechanism may be selected. In addition, the Policy-based Routing Enhancement mechanism can be used to reduce the number of hops and thus the end-to-end delay. As a result, an OConS solution can enhance the current Internet connectivity solutions, which is transparent for the users of the Data-Centres, while improving performance or throughput.

The “Wireless and Multi-P* support for ICN” use case validates the beneficial synergies obtained from the OConS multipath content delivery mechanism together with the available DTN routing mechanism based on the history of social interactions. OConS services improve the connectivity of the flash crowd users, which require a minimum reliability, a minimum QoE, and therefore, need optimized resource utilization. More specifically, the message forwarding functionality in each NetInf node is enhanced by OConS. By collecting network information via the OConS IEs, the DE in the OConS-enhanced NetInf message forwarding function can decide to activate and configure the OConS mechanisms using the appropriate EEs.

7.5 Discussion Summary

To sum up the validation discussion, the results of the OConS research activities require assessments that are different by nature: OConS mechanism validation allow to assess the performance or resource consumption improvements after deployment and during operation. These are aspects that do not express the quality of an architectural framework. Such a framework shall improve the design and development of new solutions, e.g. to better address openness and flexibility or to involve new technical enablers, as in our case, the separation of the control and the forwarding solution spaces. Evaluation criteria to adequately evaluate the properties of the OConS architectural framework and reasoning about the evaluation results are included in Deliverable D.C.4 [3].

8 Conclusion

Summary and Concluding Remarks

In this deliverable we have continued the research work on the specific OConS mechanisms (see Chapter 4) which were designed and assessed within the scope of the SAIL project; these mechanisms have been grouped according to the particular connectivity level they are applied to (i.e. flow, network or link). The results already available show that these OConS mechanisms provide (already just by themselves) several improvements compared to legacy solutions. In order to go beyond these monolithic solutions, the OConS approach also facilitates the combination of the individual mechanisms, so that they benefit from each other, bringing about a better operation. It is important to remark that this combination of mechanisms is not just a simple addition of their individual features, but a smart integration of them, so that they really take advantage from the capabilities provided by the others. This is, for instance, illustrated by an enhanced access selection mechanism, which takes decisions being aware that multi-path and advanced flow management procedures are available.

Accordingly, we have investigated and specified the OConS Orchestration functionalities (see Section 3.3 and Section 3.7) needed when combining several mechanisms in order to provide the optimal OConS service. Thus, we have defined a comprehensive OConS architecture with all the necessary Orchestration components (such as OSAP, OR, and SOP); then we have described the different Orchestration phases following their life span and applicable scope, such as OConS node and topology configuration, OConS nodes and mechanisms bootstrapping, and finally the OConS service orchestration (either triggered by an OConS user request or by a change in the subscribed network state).

Likewise, we have explored the relevant OConS interfaces and the related messages, and we have also analysed the security threats for the open connectivity services. Moreover we have developed an Information Model to capture the OConS concepts, the semantics of information, and the information processing in the OConS system.

Last but not least, we have provided a concrete Orchestration example, and also presented the OConS mechanisms benefits in two use cases from the overall SAIL flash crowd scenario, Data-Centre Interconnection and Seamless Access for Mobile Users in Chapter 5 and, respectively Wireless and Multi-P Support for Information Centric Networks in Chapter 6.

Next Steps towards the OConS Vision

When it comes to the remaining OConS efforts, we will focus on the upcoming D.C.4 deliverable on the applications for the open Connectivity Services (such as CloNe and NetInf), providing even more technical results coming with the OConS mechanisms, together with their concrete assessment and evaluation.

Furthermore, the final results from the experimentation and prototyping activities will be provided in D.C.5 deliverable (which is due at the end of the project), demonstrating these applications for Connectivity Services. We will also implement and report a proof-of-concept of the orchestration concept, using the access selection mechanism and dynamic distributing mobility anchoring mechanism.

More broadly speaking, we are aware that, although individual mechanisms improvements are still needed and can be made in the future, the main open research challenge remains the specification of a complete orchestration process, flexible, scalable and powerful enough to encompass current and new networking mechanisms.

Moreover, we reckon that the open and modular approach OConS proposes, not only eases the migration from the currently ossified layers, but it also prepares us for the upcoming SDN world where most, if not all, of the networking control functions and mechanisms can be seen as modules and implemented mainly in software (e.g., see [23]), thus benefiting from this powerful paradigm (i.e., they can be programmed, launched and updated, as we do today in the operating systems). Nonetheless, to reach this ultimate goal, the majority of the actors from the Networking and IT ecosystem needs to take pragmatic and decisive steps, notably by working on open and interoperable standards in the related fora such as ONF, IETF, BBF, and 3GPP.

A OConS Mechanisms in Detail

This Annex describes in more detail the different mechanisms referenced in this deliverable. In addition, the description of mechanisms' capabilities are also included, together with some hints on the orchestration rules applicable to them. Please note that the actual results, and the evaluation of them, is reported in the companion deliverable [3].

A.1 Multi-Path Extensions for Information-Centric Networks

Multi-path extensions for Information Centric Networks focusses on enabling the simultaneous use of multiple paths that an ICN node has to the networks. These extensions utilize the OConS framework to determine and operate the different multi-path strategies following a content request and delivery over the multiple paths.

Every ICN node deployed with the multi-path mechanism is setup using the processes described in Orchestration. There are a number of multi-path strategies that an ICN node can adopt based on the information supplied by the OConS framework. These strategies are defined as rules. These rules are evaluated continually to select the appropriate multi-path strategy to at any given time. There are 3 possible high level strategies (rules) that are defined. These strategies focus on either aggregating the bandwidth of the attachments or on enabling reliable content delivery.

- Distribution Strategy - With this strategy, the rules are set to transfer multiple content downloads over multiple attachments that the ICN has.
- Splitting Strategy - The strategy used to distribute the retrieval of one content stream into the multiple attachments.
- Replication Strategy - The strategy used to replicate the requests for the retrieval of content into the multiple attachments so at least one copy of the content is received.

The rules in these strategies consist of a set of conditions and actions. These conditions (i.e., rules, policies, preferences) which are part of the information model are held, for example within the distributed IEs. Each of these strategies have further sub strategies that enable them to be used in a particular context. For example, what kind of content retrieval is extremely important such that it is done at any cost, so that these content streams can utilize the attachment over the satellite connection.

The main focus of this work has been to contribute to the overall demonstrator of SAIL that centers around the "Event with Large Crowd" (EwLC) use case. The prototype is based on NEC NetInf Router Platform (NNRP), a NetInf prototype developed in the SAIL project. The OConS enabled NNRP prototype for supporting multi-path content delivery consists of a number of NNRP modules. These extensions have the following features.

- Segment based content retrieval where content is split into chunks and request pipelining is used to retrieve content OConS Multi-path Content Delivery
- Use of IP networks as the underlying networking technology to request for and retrieve content
- Network attachments of NetInf nodes are considered as individual paths
- Use of the GET and CHUNK messages of NNRP for content retrieval
- Implements the Splitting strategy

The prototype uses UDP as the underlying transport network (Figure A.1-(1)). It supports a number of different NetInf enabled applications including the NetInf enabled Video LAN Client

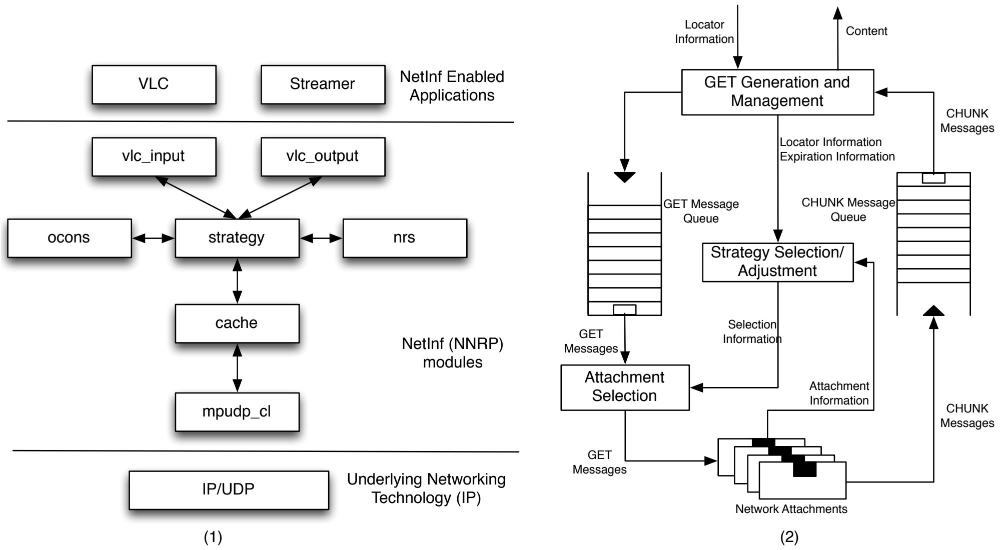


Figure A.1: OConS Multi-path NetInf Implementation (1) and the Multi-path Splitting Strategy Operation (2)

(VLC) video streaming tool [24]. The NNRP modules for multi-path content delivery are as follows:

- vlc input - handles the requests for content from NetInf enabled VLC appli- cations (VLC)
- vlc output - handles the serving of content requests of NetInf enabled applications
- strategy - handles the Orchestration and DE functionality of the OConS framework to select the multi-path strategy
- ocons - handles the interactions with the IEs to obtain information
- nrs - handles the name resolution functionality
- mpudp cl - handles the EE functionality of OConS where the selected strategy is implemented
- cache - handles the caching of content

The Splitting strategy implemented in the NetInf nodes of the prototype utilises the multiple paths by distributing the NetInf GET requests for content chunks into the multiple paths. This strategy uses the Additive Increase/Multiplicative Decrease (AIMD) mechanism [25] to adjust the distribution of GET requests (Figure A.1-(2)).

This prototype has been demonstrated at the MONAMI 2012 conference held at Hamburg University of Technology, Hamburg in Germany from the 24th to the 26th of September 2012.

A.2 Multi-Path and Multi-Protocol Transport for Enhanced Congestion Control

Mobile handheld devices have gained tremendous popularity and acceptance in the recent years. Most of the modern network enabled mobile devices have multihoming capability. 3G and Wi-Fi are the most common combination in this regard which exists in the vast majority of all smart-phones and tablets today. Research has shown that that overall performance of these networked devices can be significantly improved by striping the capacity spread over multiple network interfaces. Aggregating capacity spread over multiple network interfaces has been attempted in all layers of OSI network model. Traditional transport layer protocols such as TCP, UDP, DCCP and SCTP does not inherently support multipath data transmission features. Multipath TCP (MP-TCP) [26] and Concurrent Multipath Transfer extension [27] of the Stream Control Transport Protocol [28] (CMT-SCTP) are the two major transport protocols supporting multipath networking at transport layer today.

Performance of multipath transport protocols have known to be sensitive to path asymmetry in terms of capacity, Round-trip time (RTT) and packet loss [29]. The characteristics difference among each of the paths participating in a single data flow over multiple paths, significantly decreases the overall performance. We therefore investigate various causes of performance degradation resulting from the asymmetry of network paths with CMT-SCTP for reliable transmission of single data flow over multiple paths [30]. A simple multipath network topology of two path is simulated in *ns*-2 [31] with heterogeneous capacity and connectivity, and we observe the evolution of both directly observable behaviours, such as per-path throughputs, and internal congestion control parameters, such as windows, in order to identify and analyse the cause of performance discripancies.

A.3 Access selection and decision algorithms

Nowadays end-users might be able to connect to a wide range of access alternatives, based on different technologies. An important aspect to analyze is the large number of parameters which can be modified to bring about a better network performance. They can be classified in a twofold way: static parameters (different policies and preferences from either the user or the network) and dynamic ones (requirements of the applications, network load, etc). In this work we analyze the combination of different parameters to decide the access elements to connect to. Thanks to the OConS framework, we aim at combining different points of view (which could be contradictory between each other) when taking the decision of which base station to connect to. The following are some illustrative examples of the figures of merit which we could consider: capacity, security levels, service requirements, different policies (for instance pricing), cooperation strategies between networks, particular characteristics of the scenario... The OConS framework eases the process of combining all these elements, as well as distributing the decision between the end-user and the network elements. The following works present some of the results which have been obtained for this line of research, which is based on both analytical techniques (linear programming and game theory) and a proprietary event-driven simulator: [32], [33] and [34].

- What it does: enhanced access selection in heterogeneous networks.
- OConS service level: main focus is done at a flow level (a decision is taken upon a service request a connectivity service for a flow); network level can be also considered when connectivity is required even without ongoing services.
- Mechanism category: access and path selection.
- What it guarantees: improved QoS/QoE, considering current context and service requirements, possibility to interact with other mechanisms (which might be activated if need be).
- Mechanism constraints: there is not any particular requirement.
- What can be configured: access selection algorithms and configuration (distribution between entities).
- Needed IEs: to provide various pieces of information about links and conditions (load, quality), service requirements (mapped onto a uniform set of criteria), user and operator policies, etc.
- Needed EEs: an EE to perform the connection.
- Needed Runtime Resources: not particulary high. Some overhead expected due to signaling protocols.
- Which other mechanism(s) it complements/works with: path selection, dynamic mobility management, flow scheduling, multi-path. It is worth mentioning that this procedure could be the initial part of any other mechanism.

We work over a highly heterogeneous deployment, where a MT having different interfaces to connect to access elements which employ various technologies. In this access selection framework, the decision can be handled both at the network level as well as at the end-user. After a connectivity request, the DE will decide, using all the information it is aware of (provided by the various IEs), the access alternative to be used. Once the DE has taken the decision it initiates the corresponding connectivity configuration (mobility setup, etc), with a request to the corresponding EE.

Besides the evaluation by means of simulation and analytical studies, a real implementation of the different entities has also been done. The main goal is to assess the feasibility of the enhanced access selection procedure, based on the OConS framework and explore the potential integration with other demonstrations.

This mechanism is rather orthogonal, so it can be applied to the two use cases.

A.4 Mobile-driven QoE-aware Multihomed Flow Management

By facilitating information monitoring about remote elements, the OConS framework allows to support a more informed decision (IE–DE interaction) on which network accesses and paths to select. This allows to address the *Multihomed Flow Management* (MFM) problem, first introduced in [35], much more adequately. This problem consists, in the presence of multiple flows to several destinations, and the availability of different access networks and technologies, in selecting the most relevant access network(s), and distributing the flows in order to optimise some metric.

One particularly relevant criterion in the case of a mobile user, is the quality that user perceives from its network use. The QoE [36] models defined by the International Telecommunication Union (ITU) [37, 38, 39] can be used to map user expectations to network QoS requirements. To cover all user-relevant criteria, network access costs and battery consumptions are also taken into account.

A first evaluation of the approach using constrained optimisation [35] has provided optimal solution bounds. It can be summarised as a maximisation over possible links $\vec{A}$, then-possible flow distributions $\vec{D}$ and application parameters $\vec{p}$ of the applications qualities $Q(\cdot)$ and the opposite of the price $Pr(\cdot)$ and power use $Pw(\cdot)$,

$$\max_{\vec{A},\vec{D},\vec{p}} \left(\sum_{f\in F} W_f Q(f, p_f, q_{\text{req}}(f, p_f)) - W_b \sum_{i\in I} Pw(l_i) - W_p \sum_{i\in I} Pr(l_i) \right) \tag{A.1}$$

(a detailled formulation, and the constraints applied, can be found in [35]), which concludes that the QoE-aware Multihomed Flow Management (MFM) can support much better QoE while limiting the access costs and battery consumption.

While encouraging, this approach cannot be used as an on-line decision method, and other approaches were investigated. Based on experience learned from [34], it was decided to investigate Binary integer programming (BIP) methods to solve the problem in real-time. Due to the non-linearity of $Q(\cdot)$ [35], (A.1) had to be converted to a pre-computed utility function,

$$u_{fcin} = \alpha Q(f, c, C_{fc}, D_{in}) - (\beta E'_{in} + \gamma M'_{in}) C_{fc}, \tag{A.2}$$

which was then used in a linear objective function,

$$\max \sum_{f,c,i,n} u_{fcin} x_{fcin} - \sum_{i,n} (\beta E_{in} + \gamma M_{in}) a_{in}. \tag{A.3}$$

Optimising binary variables x_{fcin} (and auxiliary variables a_{in}) is similar to (A.1). Details of this formulation are available in [40].

The QA-MFM mechanism has then been further refined by generalizing the binary integer programming (BIP) formulation to allow for the management of both real-time and elastic flows elastic (*i.e.*, TCP-based) traffic in parallel. This relies on a two-step decision mechanism based on the assumption that real-time traffic has a higher priority. It is therefore allocated first, with a specific new utility term q_{in} quantifying the capacity occupancy ratio by the real-time flows on each interface.

$$q_{in} = \sum_{\vec{f}_{rt}} x_{fcin} \frac{C_{fc}}{C_{in}}. \tag{A.4}$$

The objective of q_{in} is to maximise the remaining capacity for the elastic traffic, scaled by δ, so as to avoid having no capacity left for the elastic traffic in case of high load (overload) situations.

$$\max \sum_{f \in \vec{f}_{\mathrm{rt}}, c, i, n} x_{fcin} u_{fcin} - \sum_{i,n} a_{in}(\beta E_{in} + \gamma M_{in}) - \sum_{i,n} \delta q_{in}. \tag{A.5}$$

For the mixed traffic scenarios, the decision process is therefore done as follows. At first, decide the network associations for real-time flows as well as their parameters and distribution. After this step, the remaining capacity which can be used by the elastic traffic is known on each interface. Then perform the optimization for the elastic flows, to decide their flow distributions and capacity sharing on each link. For pure elastic traffic or pure real-time traffic scenarios, the same method can be used, only skipping the step for the missing type of traffic.

A.5 Multi-P Decision and Transmission for the Interconnection of 3GPP and non-3GPP Systems

Future wireless networks will consist of a mixed heterogeneous 3GPP and non-3GPP access technologies. The 3rd Generation Partnership Project 3GPP has already facilitated the integration of non-3GPP access by standardizing the System Architecture Evolution (SAE) where non-3GPP access technologies can co-exist with 3GPP access networks. In such heterogeneous networks though the seamless vertical handovers can be performed between the available access networks. The question still remains whether the Quality of Service (QoS) demands of user applications can be satisfied by QoS-unaware non-3GPP access technologies. Within the context of SAIL Open Connectivity Services (OConS), this work investigates the effects of the integration of two network types on user Quality of Experience (QoE) in both downlink and uplink directions.

In order to guarantee the required QoS/QoE level also on the non-3GPP access technologies, this work proposes two novel resource estimation and management algorithms. With the help of simulation it is shown that integration of non-3GPP technologies in the existing 4G networks extends the network capacity without compromising the user QoE when the proposed schemes are deployed.

Nowadays end-user equipments are more powerful and normally have more than one interfaces which can connect to different wireless networks (e.g. mobile systems or WLAN). On the other hand, the network resources are always scarce while supporting a huge number of users running different applications. The Multi-P algorithm, enables the interconnection of LTE and WLAN, so that end users can exploit all of the available wireless resources. In addition, mobile operators can also flexibly balance the traffic load among multiple access networks.

In this work, the focus is put on developing a simulation model that can be used to realize the Multi-P transmissions of 3GPP LTE and WLAN. This involves development of simulation model according to 3GPP specifications, implementation of MIPv6 extensions to realize multi-homing and management techniques, as well as, the integration of user QoE evaluation tools. With the help of the intelligent resource management schemes proposed by this work, it is shown that network capacity improvements and user QoE enhancement can be achieved.

3GPP SAE architecture specifies how non-3GPP access technologies can be integrated in 3GPP networks, enabling a seamless handover between these access technologies. This work proposes an extension to the specifications, allowing a user to take advantage from all available access technologies by connecting to them simultaneously. The legacy WLAN does not provide QoS guarantee and therefore not suitable for real time interactive applications. However through the use of suggested algorithms for resource management and accurate bandwidth capacity estimation, WLAN bandwidth resources can be utilized for multi-homed users running QoS sensitive applications. In order to validate the proposed algorithm and its related procedures, an implementation of integrated network of LTE and legacy WLAN access technologies in OPNET simulator has been carried out. The simulation results provide proof of the concept, where the proposed scheme succeeds not only in providing QoS-aware service to multi-homed users but also improves the network bandwidth resource utilization. By outperforming the current 3GPP proposal, the new scheme assures a win-win situation for network operators as well as for users in future wireless networks.

Detailed information regarding the algorithms, their performance and the simulation model used can be found in [41] [42].

A.6 Handover with Forward Admission Control for Adaptive TCP Streaming in LTE-Advanced with Small Cells

The introduction of small cells in cellular networks promises increased capacity, due to the popular smartphone streaming and interactive video applications. Small cells (eNBs in LTE terminology) might not have terrestrial backhauling, but rather will be relayed via air interface to an adjacent terrestrially-connected small cell, termed, a donor. Consequently, handovers will be needed more frequently, while the bandwidth between the relayed and the donor cells becomes a scarce resource.

This clearly affects one of the fast-growing cellular applications, the adaptive TCP video streaming services. TCP transport protocol is becoming increasingly popular for media streaming applications, thanks to features such as congestion control, flow control and traversal through NAT gateways, thus ensuring network stability and reliability. In adaptive TCP streaming, the video is segmented into chunks that are each requested by a different HTTP GET command. The server encodes each chunk at a bit rate that matches the connection's most recent throughput, obtaining better video quality for higher throughput. However, the high handover rate introduced by small cells, increases packet loss, while solutions that forward packets from the old eNB to new one (the donor) must use expensive wireless bandwidth.

Initially, our research was targeted toward improved efficiency of Handover procedures in Multi-hop Wireless Networks. The problem and the mapping into the OConS architecture were discussed in Deliverable D-4.1 [1] (Section 6.1.5). Recently, we decided to focus on a more specific scenario of the same problem, optimizing the increasingly popular adaptive TCP streaming services with frequent handovers in LTE networks with small cells.

The details of our approach are published in [43]. In summary, our research ensures effective handover mechanisms, maintaining transport-layer sessions during the movement of the User Equipment (UE) from eNB1 to eNB2. The eNB1 decides whether to forward or to silently drop the packets received for the UE after the handover decision has been made, depending on the type of the traffic and the availability of resources. The forwarded packets might be received at the UE out of order, since new packets are routed from the sender directly via eNB2. Dropping packets during handover lets TCP recover from the packet loss, while forwarding packets from eNB1 to eNB2 significantly improves TCP performance and reduce variance, granted that wireless resources are available for such forwarding. Wireless bandwidth between eNB1 and eNB2 is a scarce resource, and thus, eNB1 must be very selective when forwarding packets. In our research we define this optimization problem and seek for good solutions.

In this dynamic forwarding decision, we consider two different optimization criteria: minimum forwarding cost, and maximum throughput. The former is more appropriate for heavily-loaded networks, where a limit on the throughput of every connection is attributed to the lack of network resources, even if every packet is forwarded. The maximum throughput criterion is more appropriate when the ability of a connection to expand its window before the next handover is mostly limited by losses of the wireless channel. Our solutions for throughput maximization also reduce throughput variance, as they minimize the periods during which TCP congestion window (cwnd) is unnecessarily reduced.

Generally speaking, our findings are that, as long as wireless resources are available for the forwarding process, it is adventurous to forward sufficient number of packets in order to avoid a connection timeout and slow start. Otherwise, it is better to not forward any packets at all. As any other OConS mechanisms, our proposed mechanism is orchestrated by the orchestration functionality. During the bootstrapping phase, it is discovered, registered in the OR, and launched (if configured to be launched automatically). At runtime, the orchestration function utilizes the OSAP interface for communicating with the user (request, status). It operates at the TCP protocol level, over an LTE network with small cells. As an infrastructure mechanism, it is transparent to any of the proposed use cases (OConS for CloNe, OConS for NetInf), and can be activated for improved performance. The efficient handover mechanism does not collide with or substitute any other OConS mechanism. Its operation, however, needs to be coordinated with the following mechanisms:

- Centralised optimisation of mobility management with a self-adapting network, A.12. Same as ours, this mechanisms is enhancing an LTE network, but addressing low-mobility users, while our mechanism optimises high-mobility users. The mobility patterns of users need to be monitored (or reported), in order to determine which of the two mechanisms is more appropriate
- Dynamic Distributed Mobility Management, A.11. When operated over an LTE or LTE advanced network, both mechanisms address session continuity, and thus, might need to be coordinated.

A.7 Distributed Data Center WAN Interconnectivity Mechanism (DDC-WIM)

Today the interconnection of data centres is a challenging problem, since the data centre operator and the network operator do not expose their deployed technologies and capabilities easily to one another. Also in case of flash crowd scenarios there can be unexpected network and processing load impacting the intended service.

Thus the OConS framework has to react to the sudden increase of users requesting a service, both in terms of providing the connectivity and the suitable processing capabilities. The OConS services and the framework used to facilitate this elasticity requirements over a wide area network (WAN) are illustrated hereafter.

The novel DDC-WIM described below was first presented in deliverable D-4.1 (D-C.1) [1], Section 8.4. The information exchanged between the OConS entities for data centre interconnection is listed in chapter B of this deliverable D-C.2. Briefly stated, the DCU (as the responsible control entity of a single OConS domain, i.e., a domain centralized SOP) collects the measurements like current link loads and load of CPUs of a local data centre and stores it inside its OR. The DCUs DE then performs the path computation from a data centre to the CN or to another data centre either upon request or on the fly. Also the DCU in its DE enforces the path by establishment of forwarding entries in the switches and routers (DCCs) along a path.

Starting with the time of instantiation the DCU monitors the utilization of the networking and

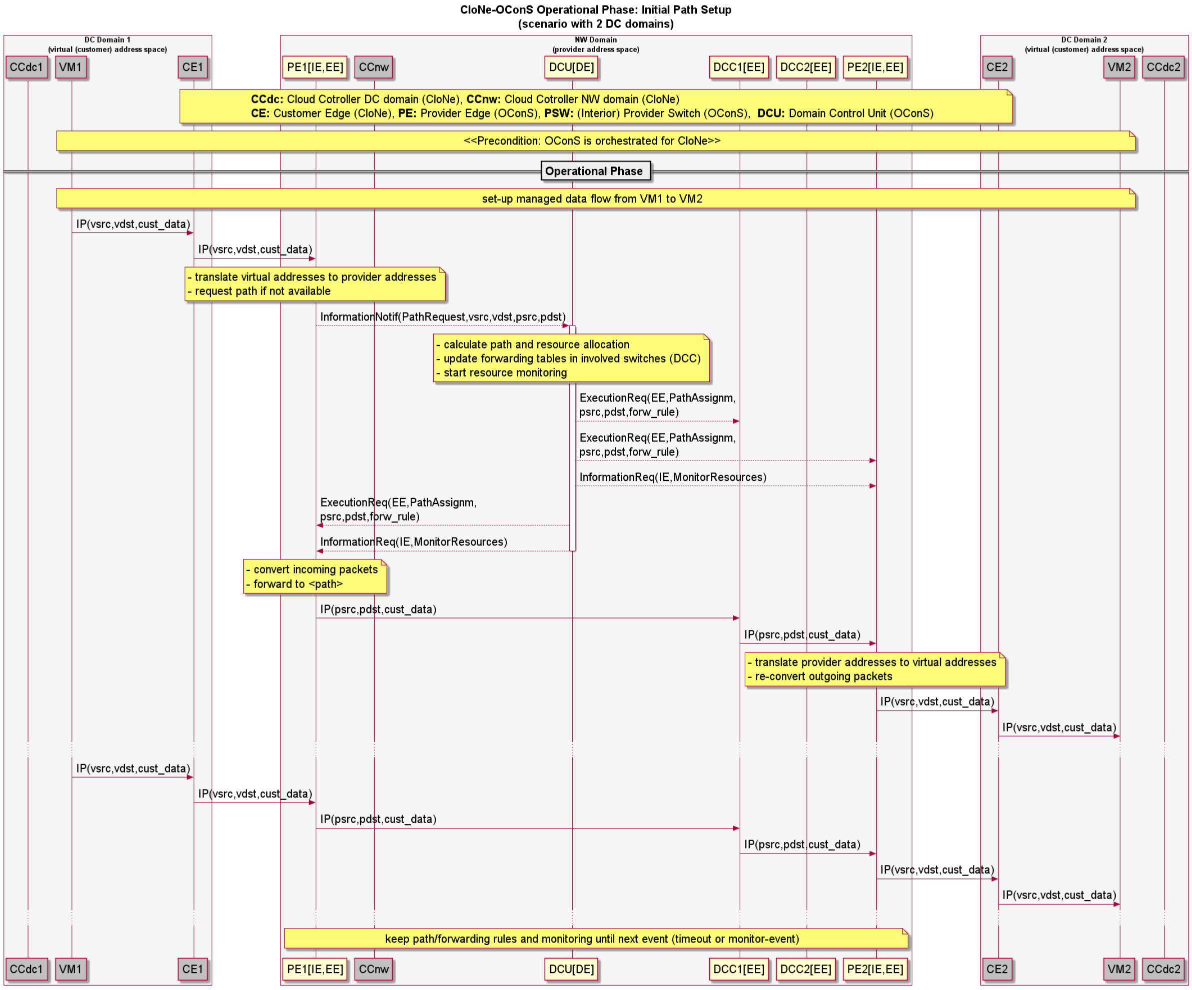

CloNe-OConS Operational Phase: Initial Path Setup
(scenario with 2 DC domains)
DC Domain 1
(virtual (customer) address space)
NW Domain
(provider address space)
DC Domain 2
(virtual (customer) address space)
CCdc1
VM1
CE1
PE1[IE,EE]
CCnw
DCU[DE]
DCC1[EE]
DCC2[EE]
PE2[IE,EE]
CE2
VM2
CCdc2
CCdc: Cloud Cotroller DC domain (CloNe), CCnw: Cloud Cotroller NW domain (CloNe)
CE: Customer Edge (CloNe), PE: Provider Edge (OConS), PSW: (Interior) Provider Switch (OConS), DCU: Domain Control Unit (OConS)
<<Precondition: OConS is orchestrated for CloNe>>
Operational Phase
set-up managed data flow from VM1 to VM2
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
- translate virtual addresses to provider addresses
- request path if not available
InformationNotif(PathRequest,vsrc,vdst,psrc,pdst)
- calculate path and resource allocation
- update forwarding tables in involved switches (DCC)
- start resource monitoring
ExecutionReq(EE,PathAssignm,
psrc,pdst,forw_rule)
ExecutionReq(EE,PathAssignm,
psrc,pdst,forw_rule)
InformationReq(IE,MonitorResources)
ExecutionReq(EE,PathAssignm,
psrc,pdst,forw_rule)
InformationReq(IE,MonitorResources)
- convert incoming packets
- forward to <path>
IP(psrc,pdst,cust_data)
IP(psrc,pdst,cust_data)
- translate provider addresses to virtual addresses
- re-convert outgoing packets
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(psrc,pdst,cust_data)
IP(psrc,pdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
keep path/forwarding rules and monitoring until next event (timeout or monitor-event)
CCdc1
VM1
CE1
PE1[IE,EE]
CCnw
DCU[DE]
DCC1[EE]
DCC2[EE]
PE2[IE,EE]
CE2
VM2
CCdc2

Figure A.2: CloNe-OConS Operational Phase: Setup of Managed Path/Flow

processing resources along these paths. In case an overload situation in the processing path is detected, the DCU either initiates a redirection of the involved paths or it sets up an additional path over less loaded processing nodes in data centres that are served by the OConS domain. All this is set up and managed by the cooperating and distributed DCUs, as OConS entities that control the resources assigned to the respective mobile cloud data centre.

During the bootstrapping process the DCU is detecting the available DCCs. Then the IE inside the DCU can start to collect and monitor the available resources from the DCCs. The latter includes resource utilization so that the DCUs local resource management can react according to predefined policies and parameter settings. So if for example a link or CPU resource gets over utilized, the DCCs IE informs the DCUs IE so that the DCU DE can react appropriately.

Summarized the mechanism described above provides the following capabilities. It does the management of processor and path selection in Mobile Cloud data centres in form of processing resource selection, path selection and forwarding establishment. The targeted OConS service level is to instantiate flows/paths and to monitore them during their life time. Also for virtual resources the mechanism guarantees dynamic resources allocation and efficient resource usage concerning CPU load and transport delay and the ability to monitor these resources. The mechanism is always used if new processing requests that have to be processed in the Mobile Data centre come in. The processing selection policy like maximum resource usage level and the optimization strategy to be used can be configured. OConS provides the needed IEs, e.g. an IE to measure link load, an IE to

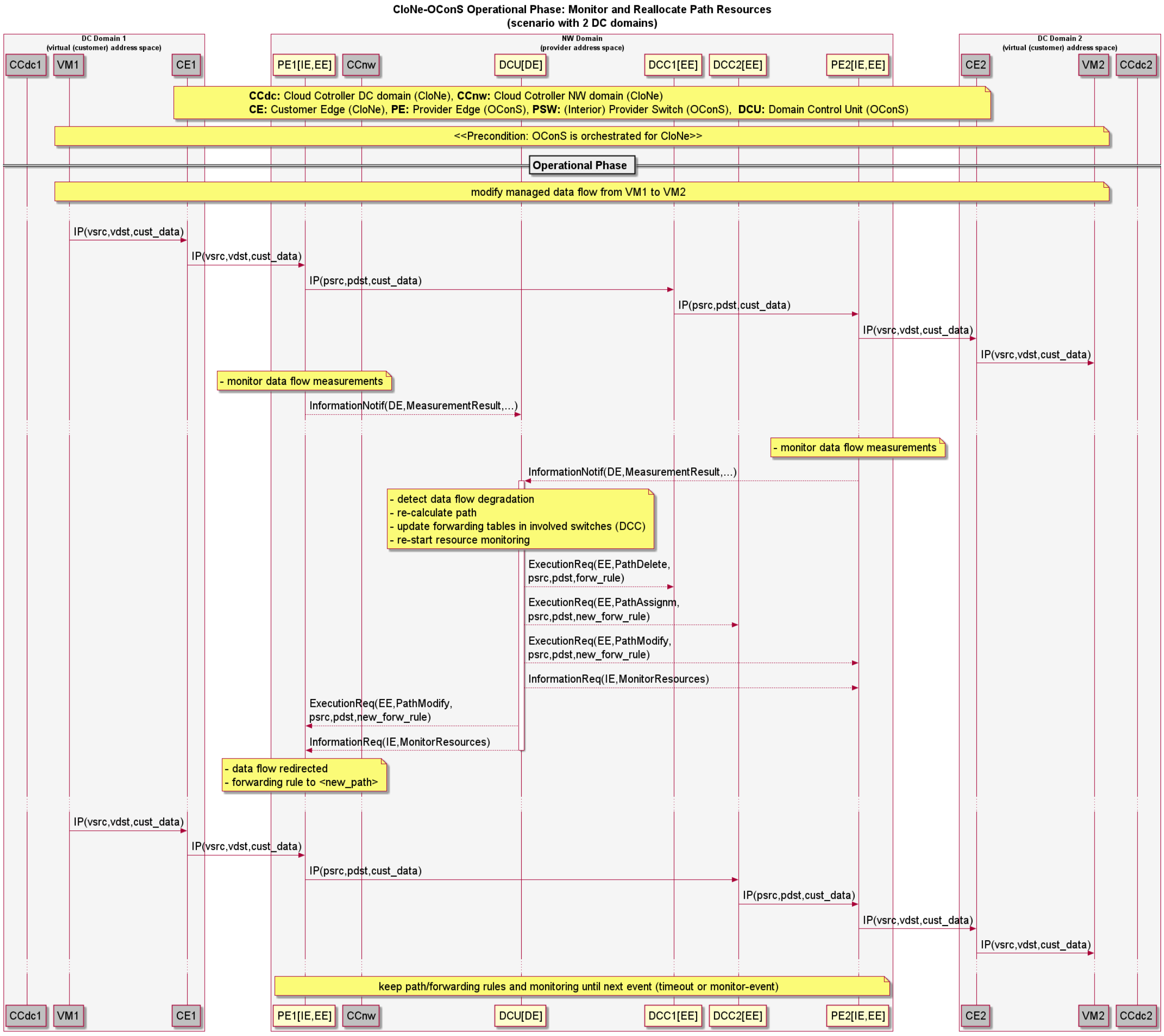

CloNe-OConS Operational Phase: Monitor and Reallocate Path Resources
(scenario with 2 DC domains)
DC Domain 1
(virtual (customer) address space)
NW Domain
(provider address space)
DC Domain 2
(virtual (customer) address space)
CCdc1
VM1
CE1
PE1[IE,EE]
CCnw
DCU[DE]
DCC1[EE]
DCC2[EE]
PE2[IE,EE]
CE2
VM2
CCdc2
CCdc: Cloud Cotroller DC domain (CloNe), CCnw: Cloud Cotroller NW domain (CloNe)
CE: Customer Edge (CloNe), PE: Provider Edge (OConS), PSW: (Interior) Provider Switch (OConS), DCU: Domain Control Unit (OConS)
<<Precondition: OConS is orchestrated for CloNe>>
Operational Phase
modify managed data flow from VM1 to VM2
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(psrc,pdst,cust_data)
IP(psrc,pdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
- monitor data flow measurements
InformationNotif(DE,MeasurementResult,...)
- monitor data flow measurements
InformationNotif(DE,MeasurementResult,...)
- detect data flow degradation
- re-calculate path
- update forwarding tables in involved switches (DCC)
- re-start resource monitoring
ExecutionReq(EE,PathDelete, psrc,pdst,forw_rule)
ExecutionReq(EE,PathAssignm, psrc,pdst,new_forw_rule)
ExecutionReq(EE,PathModify, psrc,pdst,new_forw_rule)
InformationReq(IE,MonitorResources)
ExecutionReq(EE,PathModify, psrc,pdst,new_forw_rule)
InformationReq(IE,MonitorResources)
- data flow redirected
- forwarding rule to <new_path>
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(psrc,pdst,cust_data)
IP(psrc,pdst,cust_data)
IP(vsrc,vdst,cust_data)
IP(vsrc,vdst,cust_data)
keep path/forwarding rules and monitoring until next event (timeout or monitor-event)
CCdc1
VM1
CE1
PE1[IE,EE]
CCnw
DCU[DE]
DCC1[EE]
DCC2[EE]
PE2[IE,EE]
CE2
VM2
CCdc2

Figure A.3: CloNe-OConS Operational Phase: Modification of Managed Path/Flow

measure processing load, an IE to collect the network topology. The needed runtime resources, e.g. the overhead of measuring bandwidth utilization of links or CPU resource utilization of processing nodes are rather low. For setting up the connectivity the EE in the involved DCCs is needed to execute the forwarding rules.

A first mechanism evalution is performed in [44], where also a comparison of similar concepts is carried out and a first experimental study of the feasibility of the proposed DCU concept is presented.

The procedure for setting up a path between two OConS domains and for the modification of a path are depicted in figures A.2 and A.3 respectively.

The DCU mechanism integrated with the OConS framework works as follows: when an path request for connectivity is received by the DCU DE, it instantiates the appropriate path and its monitoring mechanism. Thereby the operator policies concerning processing resource placement and path selection are taken into consideration. The DCU DE is also in charge of reacting to changes it utilization levels of its monitored resources. For achieving this goal the related IEs offer load information to the DCU so that the DCU DE can react accordingly. This is how the DCU can exploit the functionalities of the IEs in the OConS framework and gain additional benefit. Once the DCU DE has established a path and its monitoring mechanism the network will take care and reroute the path if congestion is signaled by DCC IEs concerned by the communication.

A.8 Address resolution mechanism for distributed data centers (DDC-ARM)

Rationale. More and more services are provided by large data centers with a potentially very large number of physical or virtual hosts. As the number of hosted services and service consumers increases, also the number of hosts inside a data center raises to cope with the increasing end-user demand. Current data center networks are usually based on Ethernet and mechanisms like load balancing or redundancy between data centers require a transparent connection of these Ethernet networks over a WAN. Due to the large number of hosts, these interconnected data center networks face scalability problems on different protocol layers. One such issue, which has also been discussed within the Internet Engineering Task Force (IETF), is the scalability of the link layer ARP.

Introduction. The OConS mechanism ”DDC-ARM” presented here manages the control traffic caused by address resolution procedure between interconnected data centers. It provides a network level service for the OConS user (the data center as CE) that continues to use usual Layer2 ARP procedures to find the physical MAC address for a given (private, data center/cloud internal) IP address.

DDC-ARM is based and dependent on the basic DDC-WIM mechanism as described in Section 5.2 and Annex A.7 ”WAN Interconnectivity of Distributed Data-Centres for Virtual Networks”. During orchestration the mechanism will be provided with the information of available attachment points to adjacent domains (data centers as PEs), and about the principle connectivity between the edge nodes as a result of DDC-ARM orchestration.

In addition, it is shown how an ARP proxy as an extra architectural element at the PE switches can improve the overall scalability, and that a proxy significantly reduces the ARP traffic across Virtual Private LAN Services (VPLS) switches.

The details of the proposed mechanism and of its performance evaluation model have been given in [45].

Service Guarantees. The DDC-ARM guarantees that the usual ARP procedure, used in a single Local Area Network (LAN), can be extended without protocol changes, when ’bridging’ those distributed LAN islands across WANs based on OConS domains. It prevents flooding of interconnecting WANs with unsolicited broadcast messages by introducing an intelligent address translation, and therefore improves the scalability of distributed data centers in terms of involved (private) IP hosts, and number of involved cloud components/sites (degree of distribution).

Problem statement. Current and emerging Internet applications are hosted in large data centers. Data center operators use several geographically dispersed locations for load sharing and resilience reasons. This, however, requires synchronization of the different data centers, and an efficient connection between the different locations is necessary.

Data centers often employ Ethernet as networking technology, and use server virtualization to run several virtual machines on one physical host. If a large number of nodes is attached to a data center network, the broadcast traffic caused by the ARP can result in scalability issues [46]. Although the scalability of address resolution in Ethernet networks can be improved by partitioning the network, e.g., into smaller Virtual Local Area Networks (VLANs), mechanisms like redundancy, load sharing, or virtual machine mobility require that a large subset of nodes is in the same VLAN ([46],[47]). Due to virtualization, more than 10000 nodes (either physical or virtual) may be connected to the same VLAN, which may also span more than one location. This requires a transparent interconnection of different data center sites, i.e., the transport of Ethernet frames over a WAN.

The address resolution scalability problem for large data center networks is currently discussed within the IETF [48]. For data center interconnect solutions it is important to quantify the impact of broadcast traffic due to link layer address resolution. Hence, for the evaluation of the proposed mechanism, we model the address resolution traffic and then study the signalling load caused by this traffic on data center interconnect solutions. There are many different solutions to tunnel Ethernet frames over a WAN [49] and also ongoing standardization activities on further possibilities to exchange MAC address reachability information. We concentrate on VPLS [50] as an example and quantify the amount of ARP traffic at a VPLS edge switch. In addition, we show how an ARP proxy can improve the overall scalability at a VPLS edge switch.

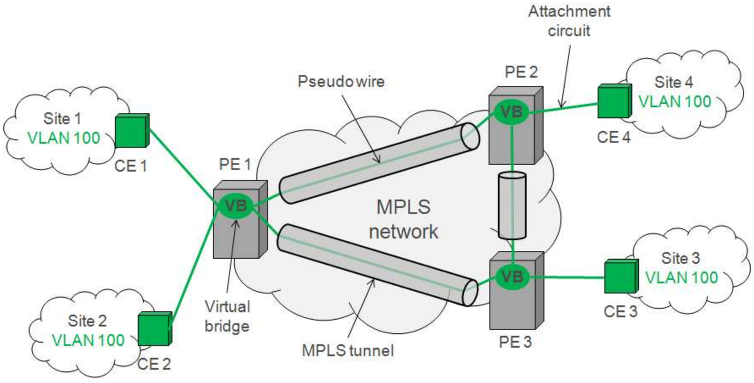


Figure A.4: VPLS architecture scenario as OConS domain

Architectural Solution. VPLS is a standardized mechanism to connect Ethernet domains over a Multi-Protocol Label Switching (MPLS) core. VPLS is implemented in PEs and the PEs are connected via a full mesh of MPLS tunnels among each other. This VPLS architecture can be considered as one possible example of a OConS architecture realization.

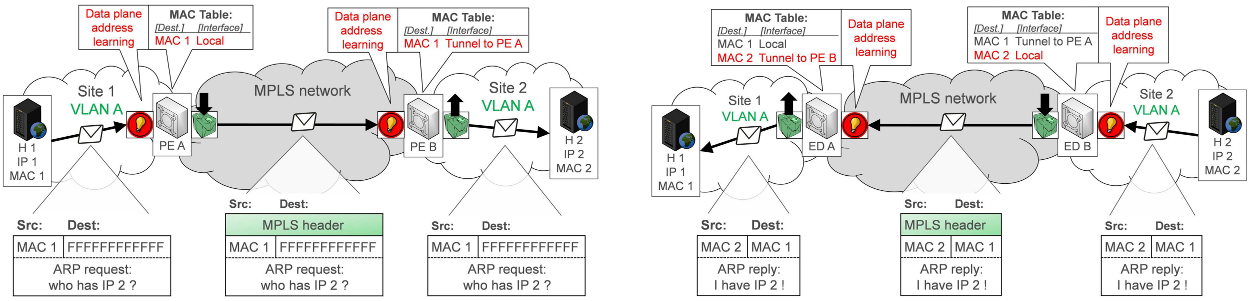
The provider edges apply data plane learning on all interfaces to learn the mapping from destination MAC address to outgoing interface. If the destination of an outgoing packet is not known, the packet is flooded via the full mesh to all other connected provider edges. To avoid loops in the full mesh of MPLS tunnels, a provider edge does not forward incoming packets from one MPLS tunnel to another MPLS tunnel. This is called the "split horizon" rule.

In the OConS architecture, the mechanism will be realized in OConS edge nodes and consists of:

- IEs that monitor incoming traffic from data center CEs and filter the ARP broadcast requests at OConS ingress nodes,
- Cooperating DEs or a DE that know about the connectivity of connected data centers, and how to forward packets between the involved OConS edge nodes,
- EEs that execute the forwarding of encapsulated ARP broadcast to the connected corresponding OConS egress nodes,
- EEs at OConS egress nodes that receive the encapsulated ARP broadcasts from the ingress nodes, remove the encapsulation and forwarding the ARP as broadcast to the connected data center/ CE.
- Optional DEs that keep track of received and transmitted requests and can decide to immediately issue a local ARP response based on the local or global ARP cache.

The initialization of the involved (edge) nodes and possible modifications of underlying topology during operation will be done by the SOP for the DDC-ARM.

Description of Mechanism. In the following, we explain the forwarding procedure for unicast traffic at provider edges. We distinguish between outgoing and incoming Ethernet frames.



(a)

(b)

Figure A.5: OconS - VPLS scenario: (a) flooding and address learning, (b) packet forwarding

1) Data Plane Learning: If a provider edge receives an Ethernet frame on one of its interfaces, it first adds or updates the entry for the source MAC address in its local MAC table. The table stores the mappings from MAC address to outgoing interface for a specific VPLS instance. The frame is then further processed depending on the communication direction.

2) Outgoing frames from internal nodes: For outgoing frames received via one of the local interfaces, the provider edge checks in its internal MAC table, whether there is an entry for the destination MAC address. If there is an entry in the table, the provider edge forwards the Ethernet frame on the associated MPLS tunnel. In case there is no entry, the Ethernet frame is broadcast on all MPLS tunnels belonging to this VPLS instance. Therefore, the provider edge replicates the packet and forwards it on the appropriate MPLS tunnels.

3) Incoming frames from external nodes: For incoming frames received via one of the MPLS tunnels, the provider edge also checks in its internal MAC table, whether there is an entry for the destination MAC address. If the entry points to another MPLS tunnel, the Ethernet frame is discarded to avoid a possible loop in the full mesh of MPLS tunnels. If the entry points to an internal interface, the MPLS header is removed and the frame is forwarded on the internal interface. In case there is no entry for the destination MAC address, the packet is broadcast on all interfaces except the interfaces pointing to an MPLS tunnel. This again avoids loops in the full mesh of MPLS tunnels.

Further in this scenario, the ARP traffic is handled by VPLS as shown in a communication example between two endhosts (H1 and H2) located in different customer sites, see Figure A.5(a). Endhost H1 has IP address IP1 and a MAC address MAC1. Endhost H2 has IP address IP2 and a MAC address MAC2. H1 knows the IP address of H2 and the first step of the well-known ARP resolution is to get the MAC address for H2. Therefore, H1 sends an ARP request to discover the MAC address of H2. The ARP request is sent to the broadcast MAC address and the source address is the MAC address of H1 (MAC1).

The frame arrives at PE A which then performs the VPLS forwarding process. First, PE A learns that MAC1 can be reached locally and stores the appropriate entry in its MAC table. The destination MAC address is the broadcast MAC address, hence PE A floods the packet to all other PEs (PE B). PE B learns that MAC 1 can be reached via the MPLS tunnel to PE A and stores this information along with the MAC address in its MAC table. PE B then removes the MPLS label and floods the frame on its site-faced local interfaces. It does not flood the packet over MPLS tunnels because of the split horizon rule. Eventually, the broadcast frame arrives at H2.

H2 now responds to the ARP request with an ARP reply, see Figure A.5 (b). Therefore, H2 sends a frame addressed to MAC1 and uses its own MAC address MAC2 as source address. The frame arrives at PE B, which learns that MAC2 can be reached locally and stores this information in its MAC table. PE B already knows that MAC1 can be reached via the MPLS tunnel to PE A and adds the appropriate MPLS header. The frame is then only forwarded to PE A, which receives the frame and learns that MAC2 can be reached via the MPLS tunnel to PE B. PE A removes the MPLS header and forwards the frame according to the entry in its MAC table. Eventually, H1 receives the frame.

A further suggested improvement of the mechanism is to introduce a new architectural element at any PE switch, an ARP proxy.

An ARP proxy is a well-known solution to improve the scalability of Ethernet address resolution. We extend our model and introduce an ARP proxy in the PE. We show how such a proxy in a VPLS switch reduces the number of ARP broadcast requests between data center sites.

ARP proxies are usually implemented in edge switches. They snoop ARP traffic and cache the mappings from IP to MAC address seen in the ARP reply packets. The ARP proxy sees the ARP replies from nodes outside its own domain as these ARP replies pass through its interfaces. The cache inside the ARP proxy can thus be seen as an aggregate of the ARP caches of the nodes in the local domain. If a node in that domain asks for an already cached IP address, the ARP proxy generates an ARP reply locally rather than broadcasting the ARP request to other domains. As a result, the ARP proxy reduces the number of ARP broadcast requests between the different domains. A more detailed description of an ARP proxy can be found for example in [51].

Summary. We introduced an OConS mechanism for the network layer address resolution in connected distributed data centers, and we described a scalable solution of this mechanism in an VPLS network architecture.

In addition, we studied how an ARP proxy can improve the overall scalability by reducing the ARP broadcast traffic. However, our proposed model is not specific to VPLS and can be adapted also for other data center interconnect solutions or for ARP traffic prediction within a single data center.

Detailed evaluation results for this mechanism will be reported in [3].

A.9 Overlay for Data-centres Interconnection

Rationale

OpenFlow (OF) has left the academic world and is starting to be deployed in Data Centres. In order to interconnect Data Centres (DCs), different vendors are proposing proprietary mechanisms built on top of OF to interconnect DCs through the WAN. In Annex A.7, we present an implementation of a DC interconnection mechanism using virtual networks that uses OConS principles. This work presents another solution for DC interconnection through an OpenFlow enabled infrastructure and is used in the OConS for CloNe use case in Chapter 5.

This work also shows the integration work between the OConS and CloNe work-packages. Our prototype integrates a PyOCNI and DCP server and interacts with the infrastructure described in Annex A.7. This mechanism has been implemented according to the description in Deliverable D-C.1 [1], Section 8.4.

Summary of the main results

The work has shown the synergies that can be mobilised within the SAIL project. The prototype is divided in for modules: 1. a Graphical User Interface (GUI) that conveys the paradigm of an OpenFlow-based Data Centre that drives 2. a Mininet-based implementation of an OpenFlow emulator and is integrated with 3. the PyOCNI and 4. the DCP servers. With this setup, we provide a proof-of-concept environment to validate the two protocols during the last phase of the project. The results of this validation will be presented in a future deliverable at the end of the project.

A.10 Policy-based Routing Enhancements

We aim at managing and controlling the Advanced Connectivity Services in an efficient and scalable manner, while specifically investigating Policy-based Routing enhancements. The mechanism studied here is not meant to be executed over the real-time OConS network infrastructure. Our mechanism is an optimisation study that is executed in a simulated or experimental network, in order to gain better understanding regarding the shortcoming of current provisioning, and in order to identify possible enhancements, new dimensioning, or better configuration for it. The results of our study can be used as benchmarking that guides network operators with regards to optimised setup of overlay routing. As such, the proposed mechanism does not need to be orchestrated, and is not directly part of the proposed use cases (although, indirectly, its guidance enable better provisioning for improved policy-based routing).

This research reported in Deliverable D-4.1 [1] (Section 7.3.1), and is published in [52]. The following text summarizes the problem and the results.

One of the main reasons that BGP is heavily used in current Internet is that it supports policy-based routing. Policy-based routing allows Autonomous Systemss (ASs) to deploy routing schemes that reflect the commercial agreements they have with peering ASs. However, when deploying policy based routing, other desired properties are not taken into account: for example, routing along shortest paths. The shortest path routing is important for efficient use of routing resources. It also contributes to reduced packet latency that is crucial for interactive voice and video services. Another undesired property of policy based routing is that the concatenation of two legal paths may be illegal due to policy constraints. For example, a direct path from A to B may be legal, a direct path from B to C may be legal, but the path from A via B to C may be illegal (if B is a customer of both A and C).

In our research, we consider the deployment of routing middle points or service gateways. These in-network devices can be used by the flows to enable shortest path or to validate path concatenation, so in the above example - if such a device is located in AS B, then one could realise the path from A via B to C. Overlay routing is a very attractive scheme that allows improving certain properties of the routing without the need to change the standards of the current underlying routing. However, deploying overlay routing requires the placement and maintenance of overlay infrastructure. This gives rise to the following optimisation problem: find a minimal set of overlay nodes such that the required routing properties are satisfied. In our research we rigorously studied this optimisation problem. We showed that it is NP hard and derive a non-trivial approximation algorithm for it, where the approximation ratio, the ratio between the result obtained by our approximation algorithm and the optimal cost, depends on specific properties of the problem at hand. We examined the practical aspects of the scheme by evaluating the gain one can get over the BGP routing problem, and showed, using up-to-date data reflecting the current BGP routing policy in the Internet, that a relatively small number of relay servers are sufficient to enable routing over the shortest paths from a single source to all ASs.

A.11 Dynamic Distributed Mobility Management

Currently, the mobile-capable terminals are mostly anchored to the same node, usually centralized and placed deeper within the core networks. In order to optimize the network behaviour (e.g., even for devices that does actually move), new paradigms for mobility anchors location and selection should be considered.

In our view, the optimal balance between host-centric and network-centric decision points can be dynamically obtained for each application flow and depending on a given communication context (i.e., resources, requirements, policies). We thus assessed in [13] two main approaches for distributing mobility management schemes, at network edges or at endhosts. Our simulations show advantages and drawbacks of the different approaches, in comparison to the use of the well known Mobile IP protocol with or without route optimization options. Our results confirm the advantages of evolving from classic "centralized" mobility management toward distributed and dynamic approaches. Moreover, they show the benefits of dynamically mixing network-based with end-to-end mobility management.

Likewise, for the execution part, we want to minimise the maintenance of unnecessary traffic encapsulation, mobility anchors and mobility-related context; thus, the anchoring node can be activated and changed only when a device moves, keeping the anchor closer to the terminals to enhance the performances for end-users and also to increase network efficiency. More specifically, the distribution and the dynamic activation of mobility management functions aims to overcome several issues, such as: the bottlenecks in centralized core networks mobility management entities (e.g., Mobile IP Home Agent (HA) or 3GPP Packet Data Network Gateway (PGW)), the maintenance of unnecessary traffic encapsulation and user's mobility context (e.g. when hosts/terminals are not in motion), or the additional end-to-end traffic delays caused by cascading hierarchical mobility anchors and/or traffic tunnelling functions (e.g. eNodeB, Serving Gateway (SGW)/PGW, HA/Local Mobility Anchor (LMA)).

Thus, a possible scheme to distribute the mobility anchors has been defined and submitted as an IETF draft, see [53]. Accordingly, we have proposed a solution based on Proxy Mobile IP [54] where the Proxy Mobile IP execution functions (i.e., LMA and Mobile Access Gateway (MAG)) were deployed in a flatter architecture (note that even in the case of Fast Proxy Mobile IP [55] we still need a centralized LMA functional entity).

As described in [53], we have devised a per-flow mobility scheme based on IPv6 network level mechanisms, i.e. we benefit from multiple IP addresses or prefixes that can be allocated by the different access routers to which the mobile terminal is attached during its movements. Therefore, our approach is both a distributed and a dynamic mobility scheme, i.e. the mobility execution functions can be distributed around a flat IP network, at the access routers level; moreover, the anchoring and the indirections functions during the handovers are dynamically activated only for active traffic flows.

At the communication set up time, any new applicative flow uses the IP address/prefix acquired locally from the mobile node's current access router; if the terminal does not move (i.e., it stayed "attached" with its current access router), the flow is routed like in any "fixed" IP network.

Mobility related contexts and encapsulation/de-capsulation operations are dynamically activated only when the terminal performs a handover to a new access router, thus ensuring service continuity for ongoing flows. In such situations, a direct tunnel is used between the flow anchor's router and the new access router to which the terminal is attached; tunneling redirections are maintained on a temporary basis, as long as they are useful for delivering ongoing traffic flows initiated with an "old" IP address; timer based and/or flow's ending events are used to refresh and clean up the tunneling and location contexts in the mobility anchors (i.e., access routers) [1].

[1] Please note that usually the corresponding nodes do not keep the track of the mobile terminals at the IP-address

Moreover, when several handover occurs and ongoing flows are active in parallel, each flow will use a direct tunnel between its initial mobility anchor (i.e. the access router on which the terminal was attached when a given flow was initiated) and the access router the terminal is currently attached to; therefore, in our scheme, there is only one level of traffic indirection per flow, avoiding inefficient chaining of several tunnels between the different routers.

Thus, our solution allows the dynamically distribution of the mobility functions among access routers for an optimal routing management. The goal is also to dynamically adapt the mobility support of the MN's needs by applying traffic redirection only to MNs' flows when an IP handover occurs.

In conclusion, we reckon that the OConS approach provides a highly scalable mobility approach, considering both mobility decision and execution functions in a distributed and flow based approach.

A.12 Centralised optimization of mobility management within a self-adapting network

Introduction and motivation

This section documents a new mechanism, named Mobility Parameters Optimization (MPO), for the dynamic optimization of the mobility management procedures currently used in the Evolved Packet Core (EPC), as standardized by 3GPP. A.6 depicts the EPC architecture and the integration of OConS entities in the MME and in a functional new entity: the Policy Manager (PM). The OConS MME groups both the legacy functionalities and the new OConS functionalities: it means that, on one side, the OConS entities in the MME (in particular the IE and EE) implement OConS interfaces, but on the other side they interwork with legacy processes running on MME. Thus, they take care of collect the information needed by MPO, locally on the MME and to transform the policy received by PM into local configuration rules.

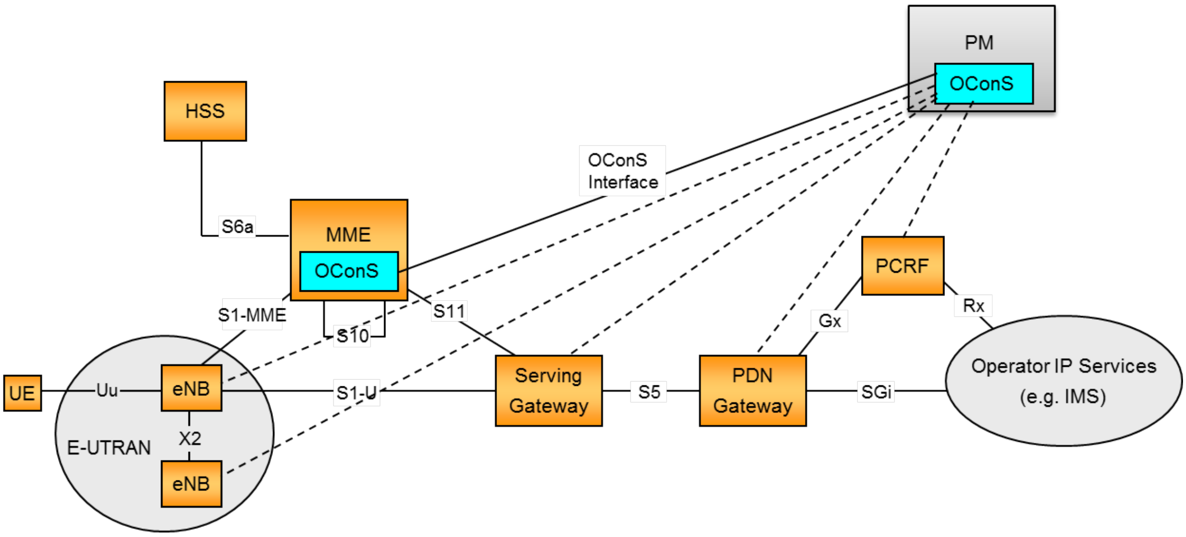


Figure A.6: MPO optimization: Architecture

In order to understand the benefits of MPO, we recall here some concepts defined by 3GPP in EPC (see [14], [15]):

level, e.g., they are using the application level registers/rendezvous; however, we are currently evaluating the usefulness of a “paging” scheme at the IP-level, to enrich our approach if needed so.

Tracking Area (TA) : a set of adjacent cells which share the same Tracking Area Identifier (TAI). When an UE (User Equipment, i.e. a mobile device) is in IDLE-mode (i.e. when it is not transmitting and it has released radio resources), the MME only knows his location at the TA level.

Tracking Area Identifiers List (TAI List) : a list of TAI (they can be adjacent), assigned to each UE by the network (i.e. the MME), every time a mobility management procedure is performed (e.g. Attach, Tracking Area Update). The TAI list is typically statically configured on MME: usually it contains the last locations visited by the UE and possibly other surrounding TAIs. TAI list are the same for all UEs.

Tracking Area Update (TAU) : this procedure is defined in [14], [15] and it's triggered by an IDLE-mode UE to inform the network about its current location, when he moves into a new TA. It is a simple Req/Resp procedure, with which the UE sends the TAI of his current cell to the MME. There are two types of TAU:

- Normal TAU (NTAU): a NTAU is executed every time an IDLE-mode UE enters a TA which is not in his current TAI List[2].
- Periodic TAU (PTAU): if the UE does not change his location, a TAU is executed periodically anyway. A parameter called Periodic TAU timer controls the time interval the device uses to trigger a PTAU. The value of periodic TAU timer is controlled by the network (MME) and communicated to the UE during mobility management procedures [3].

Paging : this procedure is triggered by the network (MME) to find the cell (eNB), which an IDLE-mode UE is currently attached to, in order to prepare him to receive traffic from the network and to allocate necessary radio resources. Since the MME knows the location of the UE on TAI-list-level, it must send a Paging Request to every eNB contained in the TAI list for that specific UE. Hence, the number of Paging messages strictly depends on the number of TAI in the TAI list of the UE.

Description of MPO

MPO focuses on the 'low mobility' devices: once MPO has identified 'low mobility' users, it works on the optimization of two parameters: Periodic TAU timer and TAI list composition. They both have a big impact on TAU frequency and Paging scope (i.e. the number of eNB to page): a short periodic TAU timer means an high number of TAUs to be performed, even if the user does not move, while a very long timer means that if the UE moves among the TAs in its TAI list, the network must page a potentially large number of cell to locate the user. Similarly, a long TAI list avoids frequent TAUs among the TAs in the list, but it means a larger Paging area. In any case, the wrong configuration of these two parameters brings a potentially large signalling overhead.

MPO assumes that low mobility users are stationary for most of the time and when they move, their target is one of the low mobility zone, e.g. from work to home. MPO dynamically builds TAI List with the TAIs where the mobile device is attached for the longest time. In addition, based on movement patterns analysis, MPO tries to predict the "next" TAIs where the user is going to move to, if he is moving and to restrict the TAI list to those TAIs. When MPO is handling users in a 'low mobility' state (they are stationary in their home/work zone), it performs the following operations:

[2] When an Idle UE changes TAI, the UE sends a Normal TAU Request to the MME. The MME replies to the UE with a Normal TAU Response, with a periodic TAU Timer and a TAI List.

[3] After a the periodic timer expires, the UE sends a Periodic TAU Request to the MME legacy. The MME replies to the UE with a Periodic TAU Response, with a periodic TAU Timer and a TAI List.

1. assign the users a longer periodic TAU timer
2. modify the TAI List, reducing the number of TAIs to 1 (the last visited TAI, i.e. home/work)

When a low mobility user move,

1. MPO fallbacks from a ’low mobility’ state to a ’normal’ mobility management transparently for the user and assigns him normal timers and TAI list.
2. Next, if the user seems to move with a known pattern (e.g. work-home), MPO assigns the moving user a new TAI list that contains the predicted ”next” TAIs for that user (e.g. all the TAIs from home to work)

Architecture

Figure A.7 shows an overview of the MPO message flow between the nodes, to optimize the mobility management parameters. The OConS DE in the PM is the responsible for running the MPO algorithm: it collects all the information that receives from the OConS IE located in the MME, by means of the OConS interfaces and enforces the new policies to the OConS EE (again in the MME), to apply the parameter optimization.

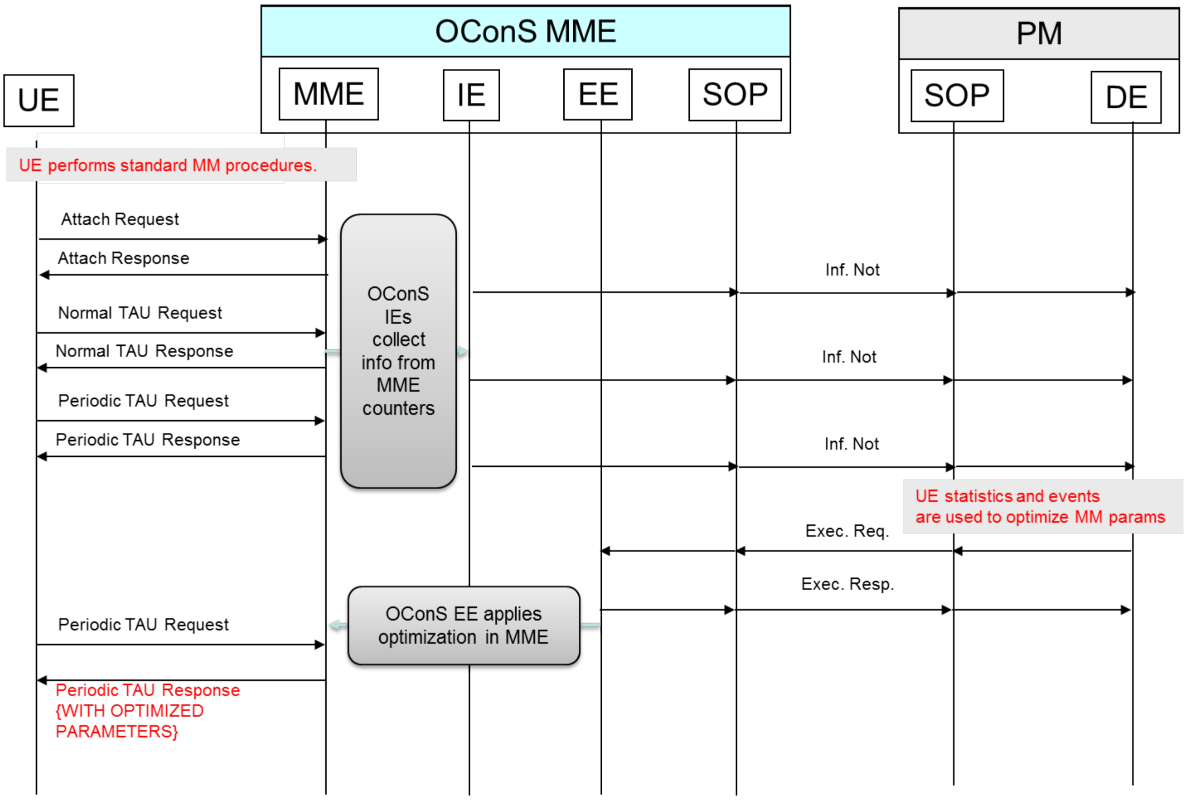

OConS MME
PM
UE
MME
IE
EE
SOP
SOP
DE
UE performs standard MM procedures.
Attach Request
Attach Response
Normal TAU Request
Normal TAU Response
Periodic TAU Request
Periodic TAU Response
OConS IEs collect info from MME counters
Inf. Not
Inf. Not
Inf. Not
UE statistics and events are used to optimize MM params
Exec. Req.
Exec. Resp.
OConS EE applies optimization in MME
Periodic TAU Request
Periodic TAU Response {WITH OPTIMIZED PARAMETERS}

Figure A.7: Overview of MPO message flow

In this document, OConS entities are only integrated in the MME, since the current MPO algorithm only needs information (about location of users), which are stored in the MME. However, in the future, MPO could be extended to use other kind of information, for example about network load conditions, coming from other nodes. Thus OConS could be also implemented on eNB (to get updated information about congestion on the radio access), Serving/PDN Gateway (to get information about user traffic, like frequent applications used) and PCRF (to get real-time information about the type of QoS treatment the user is subject to).

Detailed operations

For sake of simplicity, we'll focus here only on information provided by MME. Surely, MPO can use OConS to gather other information from other network devices, like eNB, which contain information about the congestion on the radio access network and S/PGW which contain information about user traffic, QoS profile and applications.

Phase 1: bootstrapping

When a node is started (e.g. MME, PM), it performs the discovery and the bootstrapping of OConS entities. Bootstrapping happens following the OConS Orchestration procedures. When the bootstrapping procedure is completed, any OConS node is aware of any other node and in particular any MME knows how to contact the PM that is responsible for it. After the discovery phase is successfully completed, the MME starts to periodically send notifications to the PM, containing information collected by the IE, carried into a Info_Notification message. Information include:

- Attach/Detach events of users
- TAU event of users

Additionally, they can contain information on the network load, extracted from counters on MME:

- number of connected users
- attach/detach rate
- mobility procedures rate
- MME CPU load

Phase 1b.: user profiling

Before being able to activate the actual parameters optimization, the PM must decide if a user is a 'low mobility' user or a 'normal' user, based on the information collected in the step above. As described above, if the subscription information of the users are available to the PM and if a user has been tagged as 'low mobility', the decision is straightforward and the PM has the necessary knowledge about the TAIs in which a user will be, most likely, stationary for most of his time. If this is not the case, the PM collects information on the users for a certain timeframe (e.g. one month of observations), and derives statistics for the users (e.g. most visited TAs). If, for any user, there are one/two TAs which are visited most of time (above a predefined percentage threshold), MPO label that user as 'low mobility'. This algorithm is not in the scope of the present document, since we concentrate here on how the actual optimization works, but it has to run before actual optimization can start. Two data sets are then associated to a 'low mobility' user:

- Frequent TAIs (e.g. TAI_home, TAI_work)
- Frequent TAI-path, i.e. a sequence of TAIs frequently traversed by the user, to move between TAs in the Frequent TAIs set. (e.g. all the TAIs traversed by the user when he moves from TAI_home to TAI_work)

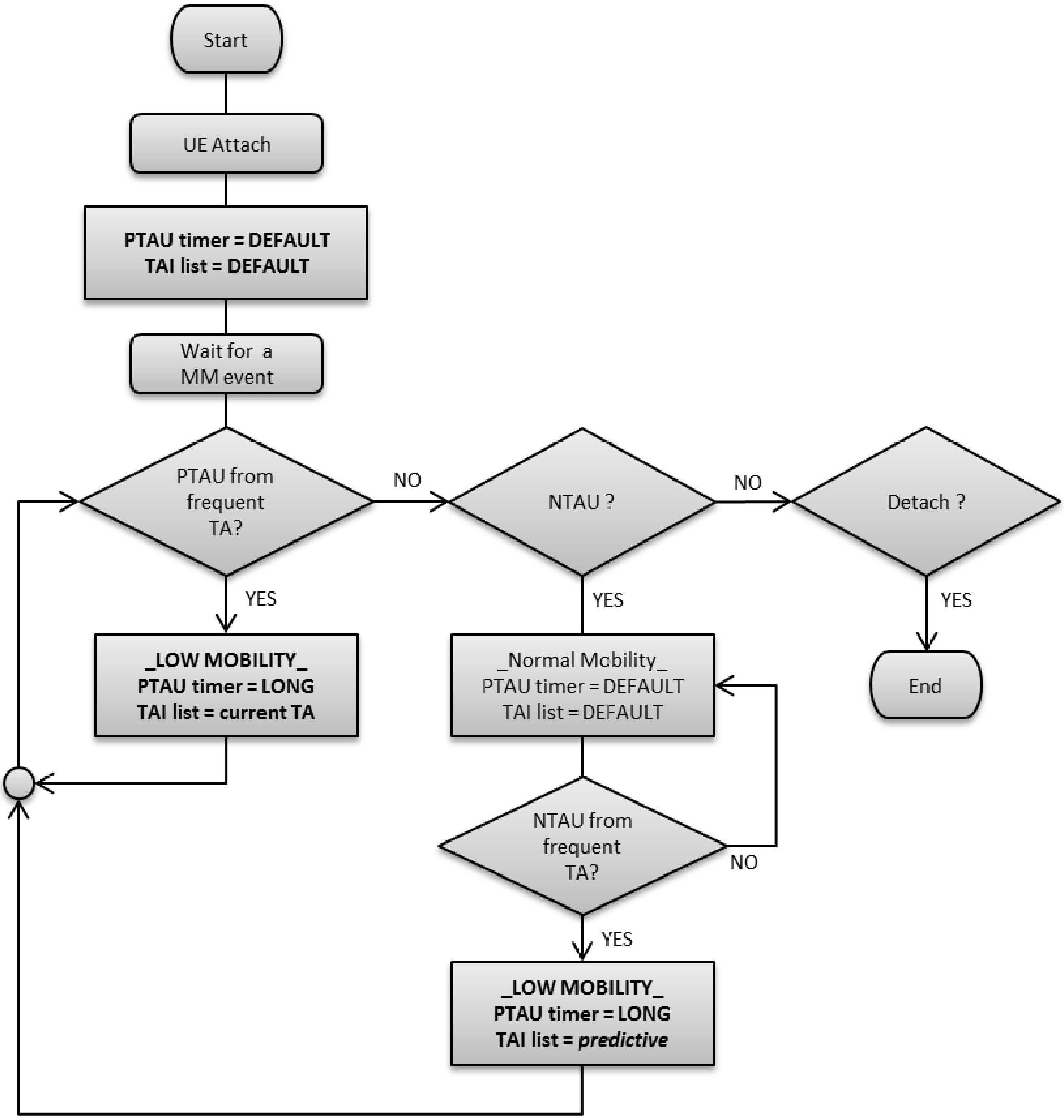


Figure A.8: MPO algorithm logical diagram

Phase 2: optimization

At high level, MPO first tries to understand if a user, previously tagged as 'low mobility', is actually in a stationary state (i.e. is located in one of the TA in the Frequent TAIs set), by looking at mobility events generated by the UE. If yes, it starts the actual optimization. An overall scheme of the MPO algorithm is depicted in Figure A.8. A detailed description of the optimization phases is given below. The MME is previously configured with a default TAI list and default timer values. When UE first attaches to the network, the MME gives every UE the set of parameters (P-TAU timer and TAI list) of the default users. The MME then notifies to the PM the UE attach, as depicted in A.9.

At this point, PM starts optimization for that user. When the first Periodic TAU Timer expires and the UE sends the TAU request to the MME (A.10), the MME still replies with the 'standard values' and notifies to the DE the 'Periodic TAU Request' Event.

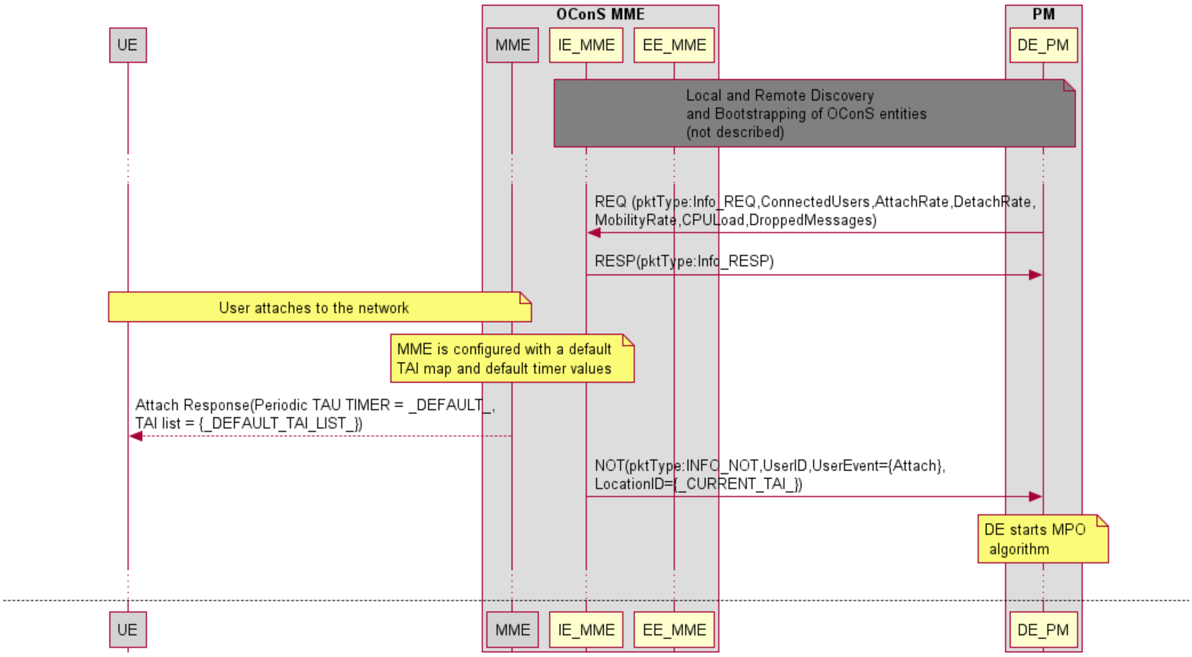

OConS MME
PM
UE
MME
IE_MME
EE_MME
DE_PM
Local and Remote Discovery
and Bootstrapping of OConS entities
(not described)
REQ (pktType:Info_REQ,ConnectedUsers,AttachRate,DetachRate,
MobilityRate,CPULoad,DroppedMessages)
RESP(pktType:Info_RESP)
User attaches to the network
MME is configured with a default
TAI map and default timer values
Attach Response(Periodic TAU TIMER = _DEFAULT_,
TAI list = {_DEFAULT_TAI_LIST_})
NOT(pktType:INFO_NOT,UserID,UserEvent={Attach},
LocationID={_CURRENT_TAI_})
DE starts MPO
algorithm
UE
MME
IE_MME
EE_MME
DE_PM

Figure A.9: MPO optimization: UE attaches

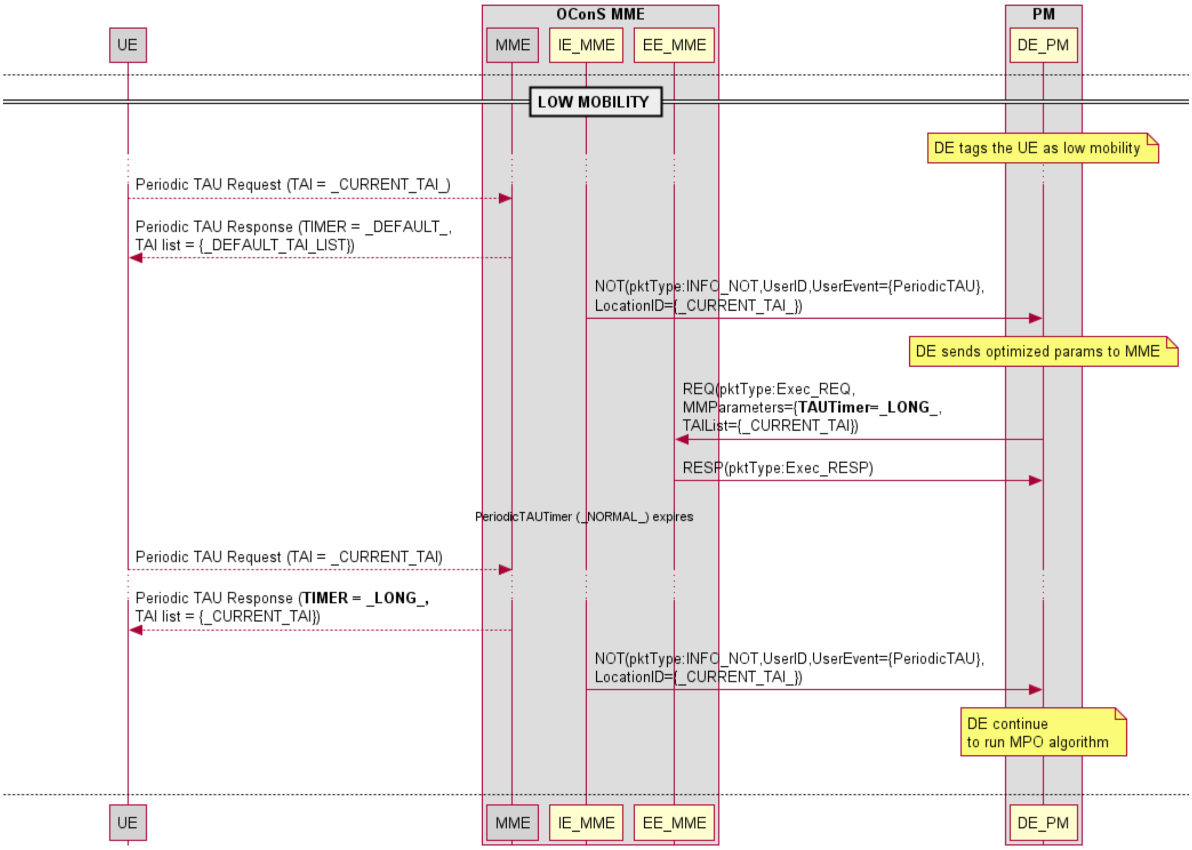

OConS MME
PM
UE
MME
IE_MME
EE_MME
DE_PM
LOW MOBILITY
DE tags the UE as low mobility
Periodic TAU Request (TAI = _CURRENT_TAI_)
Periodic TAU Response (TIMER = _DEFAULT_,
TAI list = {_DEFAULT_TAI_LIST})
NOT(pktType:INFO_NOT,UserID,UserEvent={PeriodicTAU},
LocationID={_CURRENT_TAI_})
DE sends optimized params to MME
REQ(pktType:Exec_REQ,
MMParameters={TAUTimer=_LONG_,
TAIList={_CURRENT_TAI})
RESP(pktType:Exec_RESP)
PeriodicTAUTimer (_NORMAL_) expires
Periodic TAU Request (TAI = _CURRENT_TAI)
Periodic TAU Response (TIMER = _LONG_,
TAI list = {_CURRENT_TAI})
NOT(pktType:INFO_NOT,UserID,UserEvent={PeriodicTAU},
LocationID={_CURRENT_TAI_})
DE continue
to run MPO algorithm
UE
MME
IE_MME
EE_MME
DE_PM

Figure A.10: MPO optimization: first PTAU

If the user has been tagged as 'low mobility' and he is actually in one of his TA contained in his Frequent TAI set, PM performs the following,

- periodic TAU timer is increased
- a 'custom' TAI List is filled only with the TAI where the mobile device is attached (since a long time)[4]
- communicates to the EE of the MME the parameters for the low mobility

When the standard periodic TAU Timer expires, during the next periodic TAUs, the MME is now able to assign the new TA list and the new (longer) periodic TAU timer to the UE. and informs the DE regarding the "Periodic TAU Request" Event. If the user stays in the low mobility location, PM continue to assign him the short TAI list and the long periodic TAU timer. And These messages exchange between MME and the DE_PM continues. When the 'low mobility' UE moves, towards a TA that is not included in the 'short' TAI list, it performs a TAU procedure and this triggers the MME to handle the UE with the normal mobility management:

- the TAI List is populated according to the standard rules
- the periodic TAU timer is set to the default value.

The MME notifies this event to the PM, which already has the mobility pattern of the users, and it can foresee the new locations where the mobile device will probably move, by looking at the Frequent TAI path set. The PM sends back to the MME an optimized TAI list, which contains all the TAs in the Frequent TAI path set. These TAs are, with high probability, the TAs that the UE will traverse next. When the UE performs a new TAU, the MME sends the new TAI list to the UE. If the UE moves into one of the foreseen TA, it does not need to send a TAU Request, as showed in A.11.

Interaction between MME and PM continues in the same way.

Example

Figure A.12 shows an example of usage of MPO. The UE usually stay for long time at home and at work. At the beginning, the user attaches in the home in a TA with TAI = 1. After the first standard Periodic TAU (15 min., in the example), the new 'long' periodic TAU Timer is configured (e.g. 1 hour) and a . As the UE does not move out of his home, 4 PTAUs are avoided and the Paging area is limited to only one tracking area (TAI = 1). When the UE moves the PTAU timer and the TAI List are configured again as the standard ones. When the UE moves again, the MPO algorithm is able to predict the mobility path of the user: the PTAU is configured as the long one and therefore the following PTAUs due to mobility are avoided because the TAI List is configured with the right path of the user.

Conclusions

As mentioned above, MPO allows for a more granular control of mobility management for users with predictable, stable mobility behaviour, for example commuters that move only from home to work, 4G USB keys or M2M-type devices, which don't change their point of attachment to the network. The main envisaged benefit is a lower signalling overhead, brought by:

- less Periodic TAU when the UE is stationary in one of his 'low mobility' zone
- a smaller Paging area in which to page the user, optimally only one (the 'home' zone)
- less TAU when the user is moving in a frequently used 'path' of TAs

[4] a set of TAIs, adjacent to the 'home' TAI can be assigned to the user. This is useful when the low mobility location of the mobile device is on the border between more TAIs.

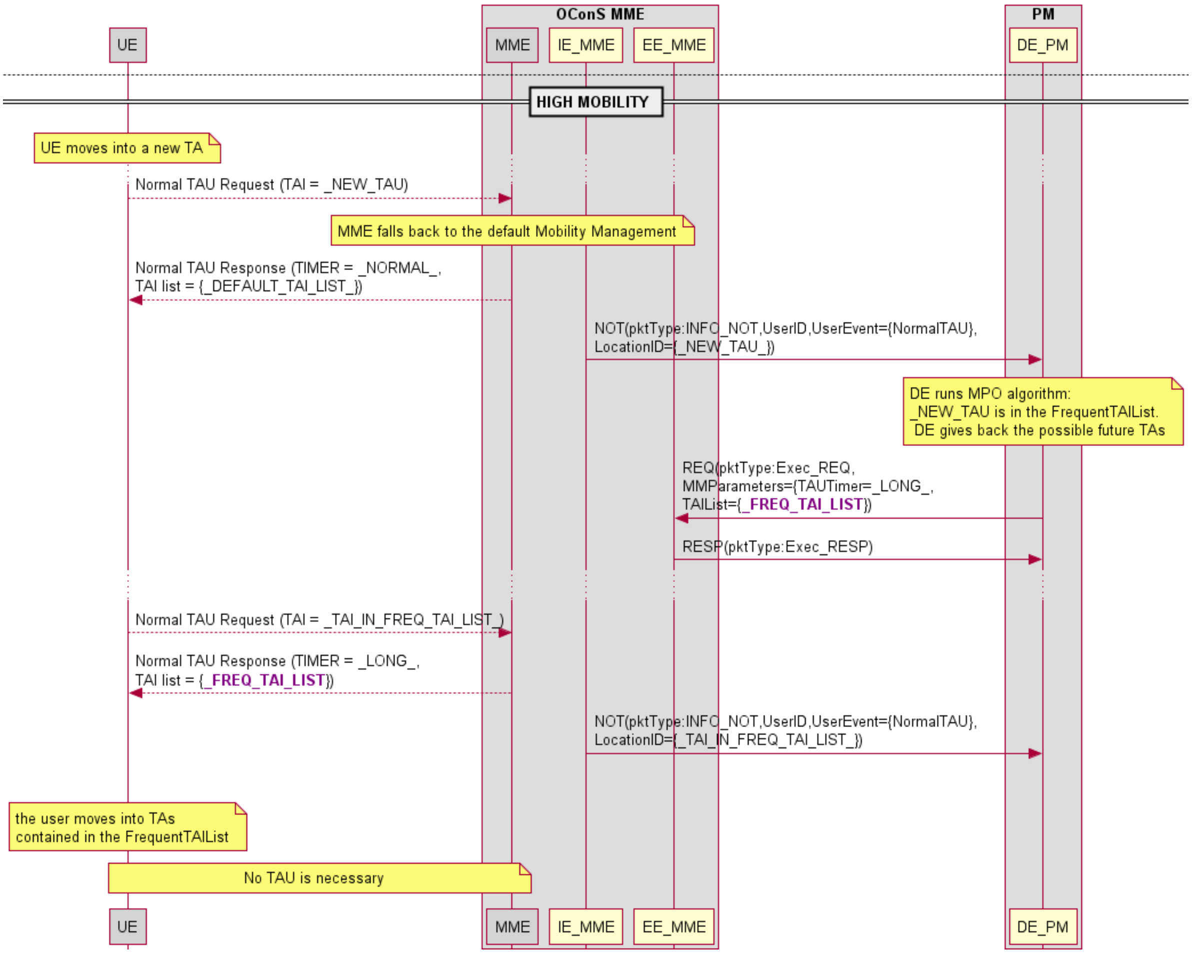

OConS MME
PM
UE
MME
IE_MME
EE_MME
DE_PM
HIGH MOBILITY
UE moves into a new TA
Normal TAU Request (TAI = _NEW_TAU)
MME falls back to the default Mobility Management
Normal TAU Response (TIMER = _NORMAL_,
TAI list = {_DEFAULT_TAI_LIST_})
NOT(pktType:INFO_NOT,UserID,UserEvent={NormalTAU},
LocationID={_NEW_TAU_})
DE runs MPO algorithm:
_NEW_TAU is in the FrequentTAIList.
DE gives back the possible future TAs
REQ(pktType:Exec_REQ,
MMParameters={TAUTimer=_LONG_,
TAIList={_FREQ_TAI_LIST}})
RESP(pktType:Exec_RESP)
Normal TAU Request (TAI = _TAI_IN_FREQ_TAI_LIST_)
Normal TAU Response (TIMER = _LONG_,
TAI list = {_FREQ_TAI_LIST})
NOT(pktType:INFO_NOT,UserID,UserEvent={NormalTAU},
LocationID={_TAI_IN_FREQ_TAI_LIST_})
the user moves into TAs
contained in the FrequentTAIList
No TAU is necessary
UE
MME
IE_MME
EE_MME
DE_PM

Figure A.11: MPO optimization: the UE is assigned new parameters

A.13 Routing and Forwarding Strategies in DTNs

In this section we briefly introduce the principles of the HURRy protocol, and then present some illustrative flow charts that specify how the functional blocks of the OConS entities have been implemented. The whole process of the HURRy mechanism in a DTN node is represented in Figure A.13. DTN nodes have individual IE, DE and EE elements, so that the routing decision can be performed and enforced locally in a hop-by-hop basis, as occurs in opportunistic topologies.

Figure A.13 shows the flow chart of the whole mechanism from a node's perspective, node A, when it detects a new physical connection to node B. P_(A,B) is the direct probability of node A contacting its neighbour node B, and calculated using the inter-contact time since their last encounter (new T_{inter}). After that, node A would update the rest of its own probabilities, P_(A,k), through the transitivity values learnt from B (node B informs about its probability of reaching the rest of nodes, P_(B,k)). If node A detects any other simultaneous connection (other direct neighbours), it will exchange its own stored probabilities with them. If physical connection with node B is lost due to disconnection, the value of P_(A,B) is calculated again with the last contact duration (new T_{intra}). From this outline, we can already notice a couple of modifications to PRoPHET, where there is no check for updates in transitivity values while connected to node B, and there is no need for updating P_(A,B) at disconnection, since PRoPHET does not consider contact duration times. The specific components to calculate direct and transitivity probabilities are further described later in Figures A.14 and A.15.

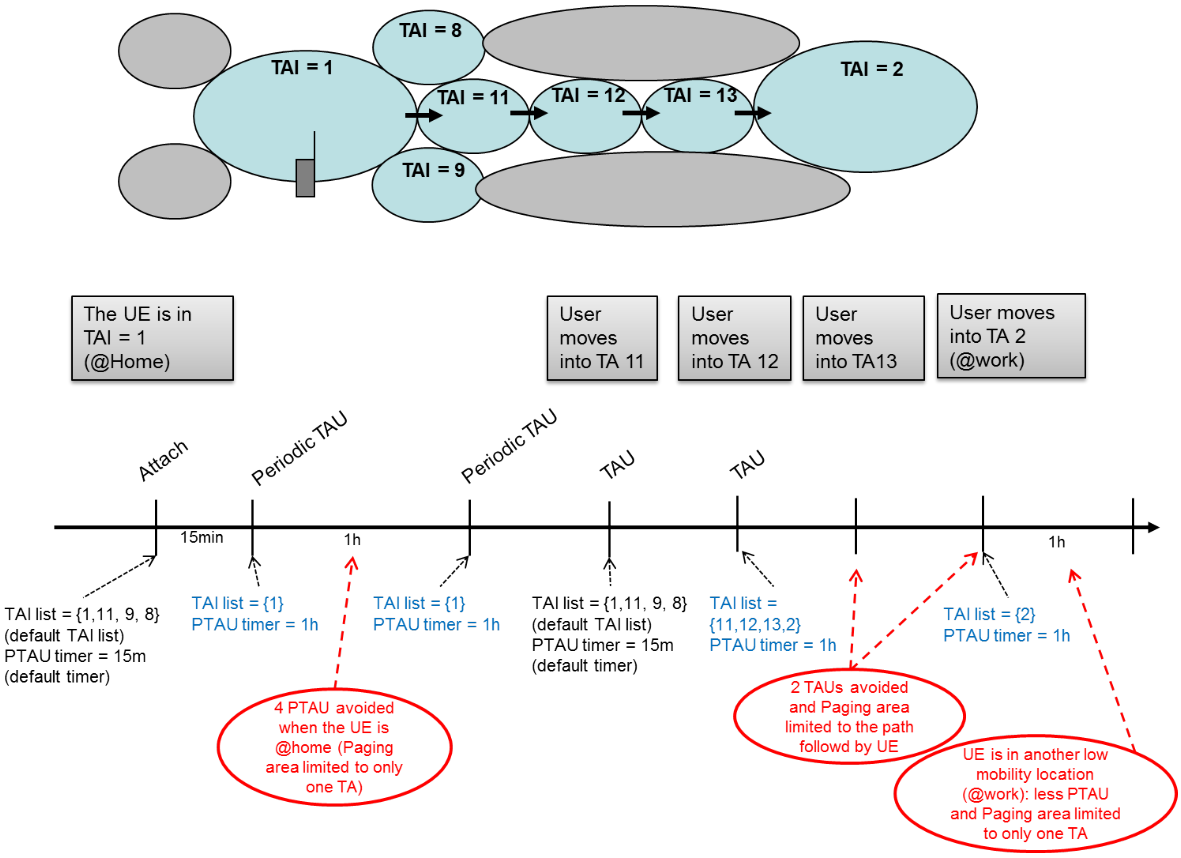

TAI = 8
TAI = 1
TAI = 11
TAI = 12
TAI = 13
TAI = 2
TAI = 9
The UE is in TAI = 1 (@Home)
User moves into TA 11
User moves into TA 12
User moves into TA13
User moves into TA 2 (@work)
Attach
Periodic TAU
Periodic TAU
TAU
TAU
15min
1h
1h
TAI list = {1,11, 9, 8} (default TAI list) PTAU timer = 15m (default timer)
TAI list = {1} PTAU timer = 1h
TAI list = {1} PTAU timer = 1h
TAI list = {1,11, 9, 8} (default TAI list) PTAU timer = 15m (default timer)
TAI list = {11,12,13,2} PTAU timer = 1h
TAI list = {2} PTAU timer = 1h
4 PTAU avoided when the UE is @home (Paging area limited to only one TA)
2 TAUs avoided and Paging area limited to the path followd by UE
UE is in another low mobility location (@work): less PTAU and Paging area limited to only one TA

Figure A.12: Example of optimization with MPO

Nodes in a challenged network can easily register the inter-contact (T_{inter}) and contact-duration (T_{intra}) time values of their historical contacts with others. But the process of estimating a representative average value, considering the history of values registered, might not be so immediate. HURRy bases this estimation on the statistical features that characterise both mathematical distributions. Assuming these distributions are highly dependant on several factors, such as the minimum time slot detected, or the aggregation of values into certain time intervals, it seems that a good approximation can be achieved deriving a histogram for each magnitude. A node implementing HURRy will have predefined time intervals, both for inter-contact times and for contact durations, which will register an incremental number of repetitions according to the history of encounters. The size of these configurable intervals does not need to follow a linear basis, so we can define smaller interval sizes for the lower range and larger sizes for the higher range of the scale considered. Equation A.6 represents the formula applied by a node to derive a representative mean value of T_{inter} or T_{intra}: $\bar{T_{\mathcal{I}}}$, where $_{\mathcal{I}}$ stands either for *inter* or *intra*.

$$\bar{T_{\mathcal{I}}} = \sum_{n=0}^{n_{curr}} \sum_{i=0}^{V} \frac{v_i e_n^i}{E_n} \alpha_n \tag{A.6}$$

n represents the sequence of discrete time, and n_{curr} stands for the current time, so $\bar{T_{\mathcal{I}}}$ is calculated at a certain instant, using the history of values registered. Take V as the maximum interval defined for each magnitude and v_i as the individual values of all those intervals. The number of occurrences per interval is denoted by e_n^i, whereas the total number of occurrences is the number of all encounters registered up to the current time instant, E_n. α_n in Equation A.6 is a factor that awards the three most recent occurrences of v_i in the summation. The values registered in

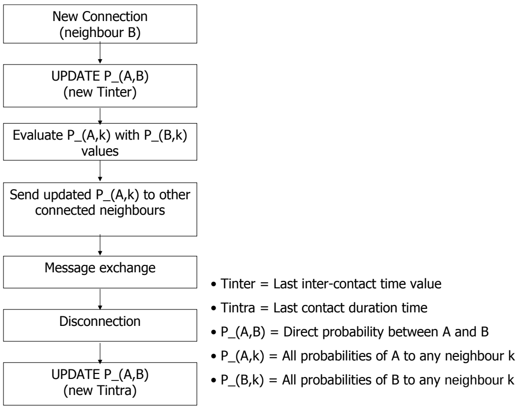

New Connection (neighbour B)
UPDATE P_(A,B) (new Tinter)
Evaluate P_(A,k) with P_(B,k) values
Send updated P_(A,k) to other connected neighbours
Message exchange
Disconnection
UPDATE P_(A,B) (new Tintra)
• Tinter = Last inter-contact time value
• Tintra = Last contact duration time
• P_(A,B) = Direct probability between A and B
• P_(A,k) = All probabilities of A to any neighbour k
• P_(B,k) = All probabilities of B to any neighbour k

Figure A.13: Sequence of components implemented in a DTN node

most recent encounters are prioritized in the same proportion as older encounters are penalized. In the case that only three (or less) encounters have occurred, α_n does not modify the average value calculated (i.e. $\alpha_n = 1$).

Each of the HURRy components is implemented by a specific algorithm. Figure A.14 shows the detail of the component that estimates a direct probability P_(A,B).

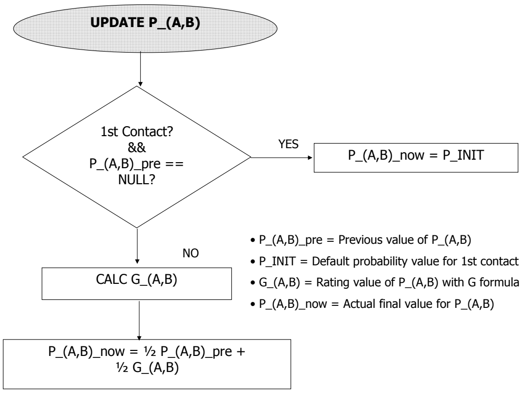

UPDATE P_(A,B)
1st Contact? && P_(A,B)_pre == NULL?
YES
P_(A,B)_now = P_INIT
NO
CALC G_(A,B)
P_(A,B)_now = ½ P_(A,B)_pre + ½ G_(A,B)
• P_(A,B)_pre = Previous value of P_(A,B)
• P_INIT = Default probability value for 1st contact
• G_(A,B) = Rating value of P_(A,B) with G formula
• P_(A,B)_now = Actual final value for P_(A,B)

Figure A.14: Detail of the estimation of direct probabilities

In Figure A.14 the functional block *CALC G_(A,B)* estimates the goodness (G) of a contact. If node A has its first contact with node B, their direct probability is initialized with a default value P_INIT. Otherwise, this component is in charge of deriving a neighbour's quality by using the G formula:

$$G = \frac{F(T)^{1-\gamma}}{(1-FT)^{\gamma}}, \gamma \epsilon [0,1] \quad \text{(A.7)}$$

Assuming both parameters are normalized to the same period in Equation A.7, F denotes the inverse value of $\bar{T}_{inter}$ and T stands for $\bar{T}_{intra}$. The goodness G of a neighbour is proportional to the frequency of contacts occurred (inversely proportional to the inter-contact time), and to the mean contact duration of past encounters. HURRy introduces a tuning factor γ in order to allow the user or application service to balance the priority among both parameters. It is easy to verify that when $\gamma = 1$ the frequency of contacts is being prioritized, whereas if γ takes values near 0 the goodness is prioritizing the contact duration. This will also influence the transitivity formula described by Equation A.8 below. The last block in Figure A.14 smooths the evolutionary slope of accumulated mean values of the probability under calculation.

Figure A.15 shows the detail of the component that updates the values of transitivity probabilities in node A. *P_(A,k)* represents the transitivity probabilities stored by node A to reach any of its historical neighbours in the DTN (denoted by k).

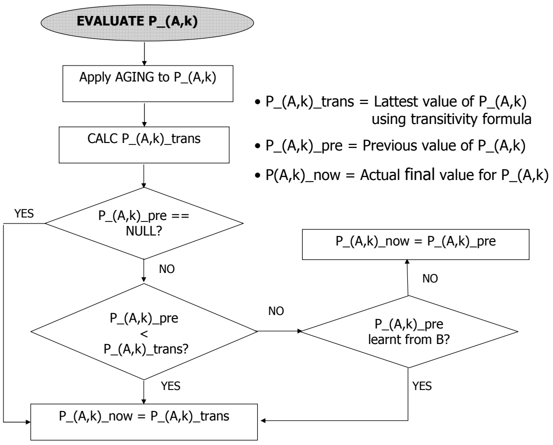

EVALUATE P_(A,k)
Apply AGING to P_(A,k)
CALC P_(A,k)_trans
• P_(A,k)_trans = Lattest value of P_(A,k) using transitivity formula
• P_(A,k)_pre = Previous value of P_(A,k)
• P(A,k)_now = Actual final value for P_(A,k)
YES
P_(A,k)_pre == NULL?
NO
P_(A,k)_now = P_(A,k)_pre
NO
P_(A,k)_pre < P_(A,k)_trans?
NO
P_(A,k)_pre learnt from B?
YES
YES
P_(A,k)_now = P_(A,k)_trans

Figure A.15: Detail of the estimation of transitivity probabilities

Unlike previous approaches, the aging process distinguishes if a third neighbour k is either (i) currently connected or (ii) not. If (i), and because HURRy considers the contact duration, the value of $P\_(A,k)$ will be incremented since last update; if (ii) the $P\_(A,k)$ value will be decremented since last update. This way, the aging may result in a positive factor if node A has been permanently connected to node k since last calculation of $P\_(A,k)$. Furthermore, in our proposal node A updates its $P\_(A,k)$ values of other currently connected neighbours before sending that information to node B. This enhancement results in a smarter management of the information exchanged within each encounter among nodes in the vicinity. It helps reducing the transitory events of intermittent connections: for instance if a third node is not simultaneously detected by two previously present neighbours due to unstable links, the first node detecting a third entity would immediately inform its connected neighbour through transitivity (e.g. Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) did not exchange new neighbours detected during a previously established connection at once). Equation A.8 represents the transitivity formula applied in the module named *CALC P_(A,k)_trans*:

$$\left(\frac{1}{P\_(A,k)}\right)^{\frac{1}{\gamma}} = \left(\frac{1}{P\_(A,B)}\right)^{\frac{1}{\gamma}} + \left(\frac{1}{P\_(B,k)}\right)^{\frac{1}{\gamma}} \tag{A.8}$$

If we only considered contact durations (i.e. $\gamma \simeq 0$), transitivity would come from the minimum value of the comparison between $P\_(A,B)$ and $P\_(B,k)$. If we only considered frequency of contacts (i.e. $\gamma = 1$), transitivity would be given by the inverse combination of both probabilities. Since we introduced γ as a tuning factor, it also influences the combination law for transitivity, where Equation A.8 provides a good intermediate law for the combination function.

Detailed evaluation results for this mechanism will be reported in [3].

A.14 Network Coding for M-to-N Routing in DTNs

In this research line efforts have been conducted towards analysing the impact of different social connectivity metrics on the performance of Epidemic Routing (ER) and Random Linear Network Coding (RLNC) when applied to broadcast and multicast communications in challenged network environments. This scope goes in line with the envisaged flash crowd scenario where information corresponding to some spontaneous phenomena (e.g. produced locally but people in the crowd) is to be disseminated over a set of ad-hoc established, yet historically correlated, wireless connections. A simulation-based study has been performed on recorded connectivity traces during a practical experiment deployed in an indoor environment, aimed at identifying those metrics that, when used as a probabilistic parameter driving the nodes' forwarding procedure, optimise the performance (i.e. delivery rate and time) of ER and RLNC. Given the sharply changing dynamics of the scenario at hand, the values of the analysed metrics have been sampled over time frames of duration α (in seconds), which in turn balances the trade-off between the tracking of the metric dynamics and the complex manageability derived from this computation. These metrics are processed by the corresponding DEs, which might follow different topologies depending on the specific scenario. For instance, if we consider the EwLC, we assume opportunistic connectivity and real-time processing of the RLNC, so it would be desirable that information about the contact history gathered by each node's IE could have been monitored and processed off-line. In that way, the metrics to be presented in this sections would be directly used during operation time by the DE in order to decide wether an incoming packet should be network encoded or not. Each node's DE would then enforce the final forwarding decision (in conjunction with the routing DE) throughout the associated EE. Thinking about a completely different scenario, if this analysis is performed over a well known and (probably) periodically established set of wireless connections (instead of spontaneous), all the metric processing could be performed in a centralised DE that would gather information from all the IEs. We present here some preliminary results from the offline analysis of contact traces from a real experiment. In what follows, T will denote the number of seconds for every daily experiment, whereas $t \in \{1, \ldots, \lfloor T/\alpha \rfloor\}$ will define the time frame index within the day under consideration.

The research specially focus on the following indicators:

1. $\varepsilon(t,i)$: Total number of edges in the connectivity graph representing the network during time frame t and day i, i.e. total number of established connections within the time frame.

2. $\delta(t,i)$: Link density of the graph, defined as $2 \cdot \varepsilon(t,i)/[N(N-1)]$, where N denotes the number of nodes in the scenario.

3. $\Phi(t,i)$: Average degree of a node in a graph, defined as $2 \cdot \varepsilon(t,i)/N$ (every edge is counted twice).

4. $\mu(t,i,n)$: Closeness centrality of node n at time t and day i, given by

$$\mu(t,i,n) \triangleq \left(\sum_{\forall m \neq n} \text{Dijkstra}\left(m, n, \mathbf{A}(t,i)\right) \right)^{-1}, \tag{A.9}$$

where $\mathbf{A}(t,i)$ denotes the adjacency matrix at time t and day i, and $\text{Dijkstra}\left(m, n, \mathbf{A}(t,i)\right)$ returns the length of the shortest (i.e. minimum hop) path between nodes m and n given the adjacency matrix $\mathbf{A}(t,i)$. In words: the higher the closeness score of a given node is, the lower the sum of distances to other nodes will be and intuitively, the more "central" the node at hand will be.

5. $\rho(t,i,n)$: Betweenness centrality of node n at time t and day i, given by

$$\rho(t,i,n) \triangleq \sum_{\forall m,p \neq n} \frac{|\mathrm{Dijkstra}\,(m,p,n,\mathbf{A}(t,i))\,|}{|\mathrm{Dijkstra}\,(m,p,\mathbf{A}(t,i))\,|} \tag{A.10}$$

   where $|\cdot|$ denotes set cardinality, and Dijkstra $(m,p,n,\mathbf{A}(t,i))$ extends its previous definition to return the set of shortest paths between nodes m and p going through node n. Accordingly, the newly defined Dijkstra $(m,p,\mathbf{A}(t,i))$ denotes the set of all shortest paths between nodes m and p, i.e.

$$\mathrm{Dijkstra}\,(m,p,n,\mathbf{A}(t,i)) \subseteq \mathrm{Dijkstra}\,(m,p,\mathbf{A}(t,i))\,.$$

   This function essentially quantifies the relative participation (importance) of node n in the shortest paths between every pair of other nodes in the network.

6. $\zeta(t,i,n)$: Eigenvector centrality, which measures the influence of a node in a network by assigning relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. The eigenvector centrality of a node n is proportional to the sum of the eigenvector centrality of its neighbors, i.e.

$$\zeta(t,i,n) \triangleq \frac{\sum\limits_{m\in\mathcal{N}(n)} \zeta(t,i,m)}{\lambda}, \tag{A.11}$$

   where $\mathcal{N}(n)$ denotes the set of neighbors of node n. We can rewrite the above expression in a compact matrix form as $\mathbf{A}(t,i)\mathbf{e} = \lambda\mathbf{e}$, where $\mathbf{e}$ represents the vector of nodes' centrality scores. λ is usually set to the maximum eigenvalue of $\mathbf{A}(t,i)$.

Besides, the following network-wide metric has been investigated, as it may shed light on the overall dynamics of the experiment.

7. Distance distribution $\mathcal{D}(t,i,x)$: It is given by the number of pairs of nodes at a given distance x (in hops), divided by $N(N-1)$ (i.e. the total number of pairs). Conceived as a probability distribution, this indicator can be regarded as the distribution of inter-node distances (hops) in the network. In mathematical notation:

$$\mathcal{D}(t,i,x) \triangleq \frac{\sum_{\forall m,n} \mathbb{I}\left[|\mathrm{Dijkstra}\,(m,n,\mathbf{A}(t,i))\,| = x\right]}{N(N-1)}, \tag{A.12}$$

   where $\mathbb{I}[\cdot]$ denotes an indicator function taking value 1 if the argument is true and 0 otherwise.

Detailed evaluation results for this mechanism will be reported in [3].

A.15 Network Coding and Transport (TCP) over wireless

It has been theoretically shown that NC techniques allow increasing throughput over WMNs. However, the combination of NC and TCP does not return the expected gains. First, the destination node shall receive packets by various wireless links (legacy and coded segments) and, if they are prone to errors, this might compensate the benefits of the coding process, since TCP flows will be exposed to a higher loss rate, compared to the legacy store and-forward scheme, resulting in a reduction of the TCP congestion window and overall performance. On the other hand, there is a potencial increase of packet loss synchronization betweenTCP flows. The loss of a single coded packet within a WMN scenario will be similar to *several flows experiencing simultaneous packet* *drops*. This results in a much prominent loss synchronization in coded WMNs. We have designed the NC mechanism so as to be integrated into the OConS framework.

- What it does: jointly combining flows to as obtain both performance improvements as well as more reliability.
- OConS service level: this implies flows, since various TCP flows need to be jointly coded; we can also say that the packet level is also affected, since the coding process is done at a segment level.
- Mechanism category: routing and coding/encoding.
- What it guarantees: enhance performance and improve reliability.
- Mechanism constraints: none, although this has been conceived to be applied in WMN at the access part.
- What can be configured: coding nodes, coding parameters (buffer size, timeout to keep unprocessed packets) and decoding nodes (decoding buffer size).
- Needed IEs: with information about available native packets, network topology, flow configuration or transmission error rate.
- Needed EEs: corresponding coding and decoding EEs.
- Needed Runtime Resources: some overhead with coding, although it might not be too relevant. Buffering for the packets to be encoded/decoded, also, buffering with the list of the overheard packets identifiers.
- Which other mechanism(s) it complements/works with: multipath.

The mechanism (integrated with the OConS) work as follows: when the request of connectivity is received, via the OSAP, the SOP handles and instantiates the appropriated mechanism, if the use of NC is suitable, the corresponding DE will be in charge of choose the set of packets to be coded together for maximizing the possibilities to have a correctly decoding at the destination node. For achieving this goal the appropriate IEs will offer different information, as which packets have been already received at the destination, so as to select those with which the original packet can be coded; also, the decoding node has to keep track of a certain number of overheard packets that could be used for future decoding operations. In this way, the NC can take advantage of the functionalities of the IEs in the OConS framework. Once the DE has taken the decision it will send the coding and decoding request to the EEs within those nodes involved in the communication.

For this line of investigation some initial results can be found in [56].

A.16 OConS Supported Dynamic Radio Resource Allocation for Virtual Connectivity

The virtualization of the wireless access as integral part of virtual networks is a challenging problem, since in wireless networks the changes in capacity/availability of radio resources, due to the inherently limited capacity, may affect the achievement of VNet contracted requirements. A VNet Radio Resource Allocation (VRRA) mechanism, called DynamicVRRA, is proposed to address the provision of requested capacity (data rate) for virtual connectivity over wireless heterogeneous networks, maintaining the isolation among the virtual networks. VRRA allocates radio resources adaptively and cooperatively, from different Radio Access Technologies, to the virtual resources in order to achieve the VNet requirements. The mechanism was presented and modelled according to OConS framework in Deliverable D-4.1 [1] (Section 7.1.3). In brief, after an OConS user connectivity request is received, via the Orchestration Access Point, the Service Orchestration Process handles and instantiates or (re)configures the VRRA mechanism for the new connectivity requirements, e.g., QoS type for the virtual resource, capacity or delay. VRRA is implemented on a DE in a cluster manager, which is responsible to manage a given set of BSs, and one DE per BS for local resource management. The decision for initial allocation of radio resources, based on a pre-configured strategy, is sent from the cluster manager DE to the several EEs (link schedulers) in the BSs. In each BS, the DEs use the key performance indicators (KPIs), e.g., wireless rates and utilization, in the IEs for detection of under utilization and/or lack of capacity situations. To support the decision for reallocation of radio resources in co-located BSs, the calculation of the BS cost is performed based on collected KPIs. A short description of the mechanism capabilities is provided next:

- What it does: Management of virtual access over wireless heterogeneous networks.
- OConS service level: link.
- Mechanism category: Dynamic radio resources allocation for virtual resources.
- What it guarantees: Provision of data rate contracted, isolation among virtual resources, efficient use of resources.
- Mechanism constraints: Only used if virtual access resources are instantiated over wireless heterogeneous networks.
- What can be configured: Requirements for virtual resources (requested QoS parameters, QoS type), strategies used to select access nodes, set of access nodes in the cluster.
- Needed IEs: Wireless rates,
- Needed EEs: Link Scheduler.

The OnDemand VRRA (VNet Radio Resource Allocation), as the previous Dynamic VRRA algorithm is developed to perform the mapping between virtual and physical links, adapting on demand the allocation of radio resources to the wireless networks conditions and Virtual Base Station (VBS) usage. The main difference with the algorithm presented before is that the Radio Resources Units (RUs) are not pre-allocated to the virtual resources but they are being allocated as they are requested by end-users. The OnDemand VRRA functionalities are distributed between the virtual resource allocation and the Radio Resource Management (RRM). Since one is dealing with heterogeneous networks, it will be implemented at the cooperative RRM level, managing all the heterogeneous wireless networks in the area, and at the RRM level, being locally implemented at the BSs. At the cooperative RRM level, it manages the aggregated capacity provided to the virtual resource, by sharing the set of available radio resources, from all RATs. At the RRM level, it maps the requested capacity to a particular Radio Access Technology (RAT) onto RUs allocated to end-users, and applies data rate reduction strategies.

Virtual networks are created with a certain level of guarantees for their requirements, according to the contract being established. This is also applied to the VBSs composing the VNet in the several geographical locations. Two different possibilities are considered within this evaluation: Guaranteed (GRT), and Best-Effort (BE). The former ensures that the requested constraints will not be violated at any time, under normal network operating conditions, while the latter provides a best-effort service, i.e., no guarantees are given when data will be delivered. OnDemand VRRA is responsible for dynamically (re)allocating RUs to reflect the network operation condition, satisfying the VNet minimum capacity. This is supported by a VNet priority scheme and a data rate reduction strategy, besides the access selection mechanism. Concerning the access selection, end-users are connected to the different virtual resources according to the requested service and their contract with the operator(s). The physical connection is established over one of the existing RATs in the coverage area, according to a list of preferences related to the requested service, the available capacity, and the strategy defined for resource evaluation. This strategy, e.g., minimum load, minimum cost, and/or minimum energy state, is based on a cost function derived from [57], where several Key Performance Indicators are weighted. The VNet priority scheme runs in a Cluster Manager (CM), allowing the differentiation in handling end-users according to the type of VNet and the VBS serving data rate. VNets are initialised to be handled with priority, all the BSs in the cluster being informed of this, to activate the data rate reduction process. When the minimum contracted data rate is reached, the priority to be given to end-users who wish to connect to this

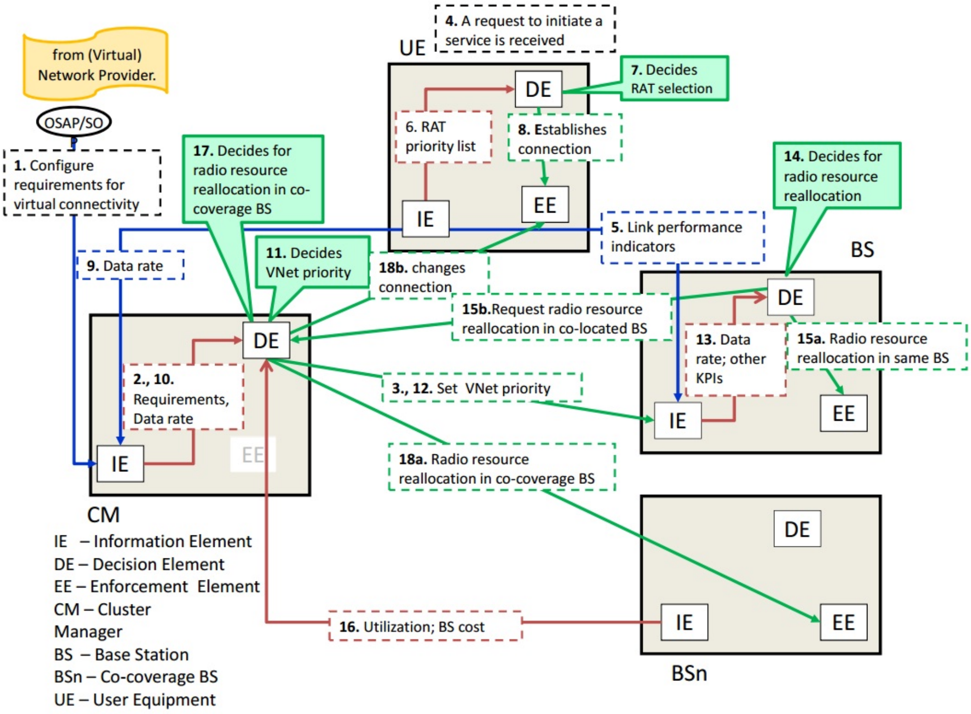

4. A request to initiate a service is received
UE
from (Virtual) Network Provider.
DE
7. Decides RAT selection
6. RAT priority list
8. Establishes connection
OSAP/SO
1. Configure requirements for virtual connectivity
17. Decides for radio resource reallocation in co-coverage BS
IE
EE
14. Decides for radio resource reallocation
5. Link performance indicators
BS
9. Data rate
11. Decides VNet priority
18b. changes connection
DE
15b.Request radio resource reallocation in co-located BS
DE
2., 10. Requirements, Data rate
13. Data rate; other KPIs
15a. Radio resource reallocation in same BS
3., 12. Set VNet priority
IE
EE
IE
EE
18a. Radio resource reallocation in co-coverage BS
CM
DE
IE – Information Element
DE – Decision Element
EE – Enforcement Element
CM – Cluster Manager
BS – Base Station
BSn – Co-coverage BS
UE – User Equipment
16. Utilization; BS cost
IE
EE
BSn

Figure A.16: OConS model of the On-demand Virtual Radio Resource Allocation mechanism.

virtual resource is deactivated. This priority scheme based on the VBS serving data rate, allows one to implement a data rate reduction strategy whenever the GRT VNets have priority, preventing starvation on BE VNets if the contracted data rate in GRT VNets is reached. The data rate reduction strategy is as follows. Whenever the VNet priority is activated for a GRT VNet, and the EUser tries to connect to a BS in which there are not enough RUs to assign to the end-user, BE end-users connected to the BS are reduced according to:

1. the QoS priority class of the performed service [58], end-users performing services with lower priority being the first to be reduced;
2. their Signal-to-Interference-plus-Noise-Ratio (SINR), end-users with lower SINR being reduced first to allow optimising radio resource utilization;
3. if still there are not enough RUs to reach the requested data rate, the CM is requested to do the evaluation of co-located BSs, in order to select the one with enough RUs available and the minimum cost for handover of end-users.

OnDemand VRRA was modelled according to the OConS architecture, to take advantage of its flexible approach, e.g., concerning the activation and configuration during network operation. A Decision Element (DE) has been identified in the Cluster Manager (CM) that is responsible to manage a given set of BSs, and local resource management is performed by other DEs per BS. The former is responsible to apply the priority scheme described above, and to reallocate RUs in co-located BSs for vertical handovers; the latter, based on the VNet priority scheme, implements the data rate reduction strategy. An additional DE is taken at the end-user Equipment to deal with the access selection mechanism; although it can be external to the OnDemand VRRA algorithm, it has been also considered within this work. Figure A.16 illustrates the mechanism mapping, the numbers in the boxes being a possible sequence of steps produced.

A.17 Radio Resource Management Mechanism for Multi-Radio Wireless Mesh Networks

An OConS mechanism for radio resource management of multi-radio WMNs is proposed. Different aspects of this mechanism are described in [59], [60], [61], [62] and [63]. It is supported by a radio agnostic abstraction-layer is proposed, between the Network and Data-Link layers, enabling to control and operate multiple radios on a multi-radio MAP. The proposed mechanism is implemented in this abstraction layer, supported by the functional OConS architecture as follows:

- The IE does the monitoring and sharing of radio resources of the MAP and its neighbours, essential information for the distributed mechanism.
- The DE consists of the mechanism itself, which periodically optimises the transmission power level, bit rate and channel based on the information collected by the IE.
- The resulting decisions are then enforced in the EE, the operational part of the MAP.

This OConS mechanism has been presented and described in detail in Deliverable DC1 [1] as well as in [60]. A short description of the mechanism capabilities is provided next:

- What it does: Management of WMN connectivity.
- OConS mechanism level: link.
- Mechanism category: Optimisation of radio interfaces' operating bit rate, transmitted power level and channel.
- What it guarantees: Max-min fair throughput for all flows.
- Mechanism constraints: Only used in multi-radio WMNs.
- What can be configured: Number of available channels, available bit rates and associated minimum SINR, number of available power levels.
- Needed IEs: Channel utilisation by neighbouring nodes, load, distance to the gateway, number of flows crossing the node
- Needed EEs: radio interfaces' operating bit rate, transmitted power level and operating channel
- Which other mechanism(s) it complements/works with: It is a link layer mechanism for multi-radio WMN. It can work with any mechanism.

A.18 Cognitive Radio Systems through Spectrum Sensing Techniques

Spectrum sensing and decision (i.e. to capture spectral measurements over a certain bandwidth, based on which a decision on the spectrum occupancy is taken under a certain hypothesis rule) are essential tasks of any cognitive radio system. As such, many different schemes have been explored so far in order to perform this task in the most accurate and reliable way, which can be a priori classified as:

- conventional non-collaborative spectrum sensing, where a decision is taken at every node of the network in isolation;
- collaborative spectrum sensing, where the spectral measurements registered by different nodes are combined - either in a centralized or distributed fashion - so as to produce a decision with higher reliability than the case where the decision is taken based on a single measurement.

OConS analysis opted for the latter approach: after nodes monitor and estimate the signal power level, the DE of each node compares this estimated level with a certain pre-established threshold to decide if a certain channel is occupied or not. This strategy is based on a pure hard decision of each network node. Nevertheless, and due to the inherent unreliability of the estimation problem with low SNR, it is more accurate to associate this estimation with a metric representing the probability of channel occupancy. Using this "soft" decision, each network node relaxes the responsibility of the final decision, relying on a higher level DE to combine a set of soft-decisions. Therefore, it is this second level DE which elaborates a final (hard) decision based on the occupancy probabilities received by every other node.

A number of collaborative spectrum sensing techniques pre-process the spectral measurements of every compounding sensing node of the network so as to produce a binary (hard) local decision on their occupancy, followed by a hard-decision fusion approach (e.g. OR, AND) that generates the final spectral occupancy metric. Technical advantages of these hard-decision based techniques are found on their simplicity for their implementation in conventional digital hardware. However, as stated in the theory of evidence developed by Arthur P. Dempster and Glenn Shafer [64], binary local decisions can be easily outperformed, in terms of credibility and reliability when applied to outcomes of a same event, by soft fusion techniques where the unprocessed outcomes of the said event are input to a unique soft test. This interesting result motivates the upsurge of soft-decision based fusion techniques applied to spectrum sensing for cognitive radios. Among the broad portfolio of such soft-decision combining approaches, in OConS we have concentrated on the so-called Linear statistics Combination (LC) ([65]). LC hinges on linearly combining the unprocessed spectrum energy measurements captured by cognitive radio nodes by means of a set of configurable coefficients, based on whose result a decision of "occupy" or "free" is taken. When the LC is applied at each band of a broad bandwidth, the resulting scheme is rather known as multiband LC. The values of such coefficients are usually set equal to each other in the conventional implementation of the LC approach. Taking into account this state of art, the main objective of the work has been focused on assessing the benefits of optimized linear collaborative multiband spectrum sensing in cognitive radio networks with respect to its non-optimized counterpart. Specifically, we have concentrated on linearly combining the spectrum information registered by each compounding sensing node of a cognitive radio network based on a set of heuristically-optimized coefficients. Such an optimization hinges on maximizing the aggregate throughput while keeping the interference at each subband below a certain threshold.

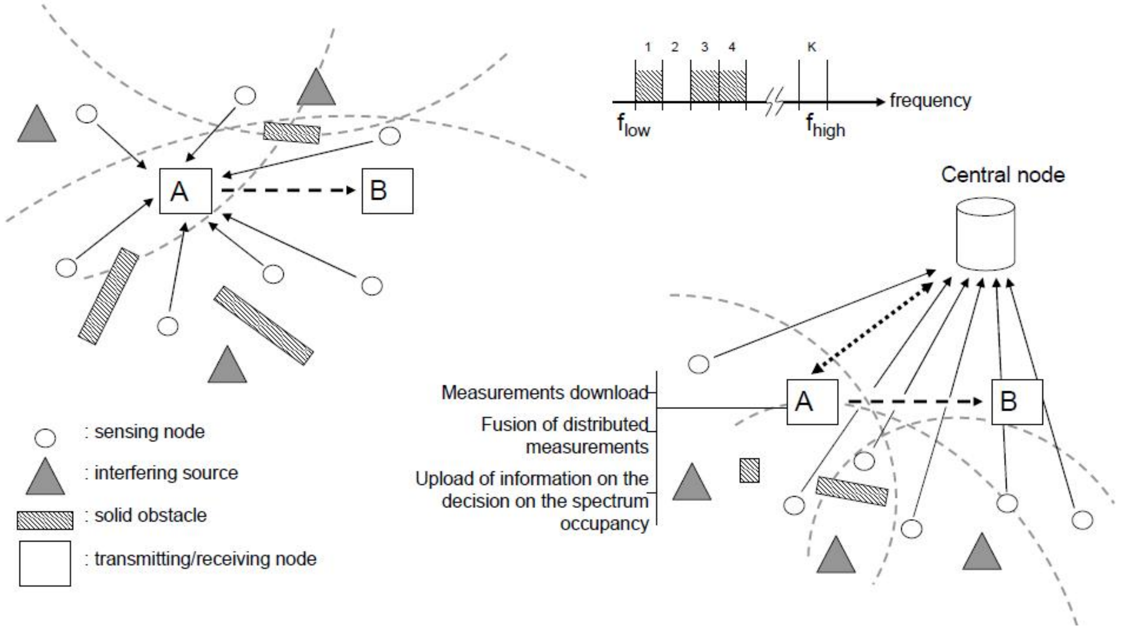

A
B
: sensing node
: interfering source
: solid obstacle
: transmitting/receiving node
1
2
3
4
K
frequency
f_{low}
f_{high}
Central node
Measurements download
Fusion of distributed measurements
Upload of information on the decision on the spectrum occupancy
A
B

Figure A.17: (Left) Example of the considered cognitive radio network scenario addressed in our proposal; (Right) Mapping from the general setup depicted on the left to the technical architecture envisaged in OConS

In essence, we deal with the cognitive radio scenario depicted in Figure A.17 (left) where a given pair of nodes (labeled with ”A” and ”B” in the figure) aims at communicating with each other over a K-subband spectrum support in presence of multiple interferers, which are denoted with triangles in the plot. Such interferers may represent primary cognitive transmissions held on the same spectrum band, nodes broadcasting data on the same given band, or any communication link of higher priority. On this purpose, transmitter ”A” must adapt its transmission parameters as a function of the spectral information captured by a set of N sensing nodes (marked with circles in the figure), which extracts a K-length normally-distributed spectrum level vector by means of an energy detector implemented at each subband. It should be noted that this spectral information registered by any given sensing node is subject to multi-path and shadowing due to solid obstacles existing in the network, and/or distance-dependent path losses, which the reliability of the spectrum measurements.

This sensed information is forwarded to a fusion central node, which may be located at transmitter ”A” itself (as assumed in the depicted network), at a node specifically deployed to this end, or at any sensing node spread over the network. The information is forwarded from the sensing nodes to the fusion centre via dedicated control channels, or by storing such spectrum measurements in a common database, which can be accessed by all nodes in the network. The aforementioned fusion centre combines the received distributed spectrum measurements so as to exploit their spatial diversity and enhance the reliability of the sensing process. In this context, as aforementioned in the previous section several combining strategies (e.g. OR, AND) operate on hard-decided distributed spectrum measurements, where the term ”hard” denotes that a decision on the absence (”0”) or presence (”1”) of signal in a certain spectrum subband is taken on every received measurement prior to its combination with the other hard-decided measurements. Notwithstanding the ease of implementation of these hard-decision fusion techniques, soft-decision approaches are preferred by virtue of their enhanced flexibility and their generally higher reliability. Having said this, we recall that a major focus will be placed on linear statistics combination (LC): if we arrange the level sensed by the n-th sensing node in the k-th subband as $Yk(n)$ (with n=1N and k=1K), LC performs the following test:

$$Wk^TYk = \sum_{n=1}^{N} Wk(n)Yk(n) \tag{A.13}$$

where $Yk = [Yk(1)Yk(2)Yk(N)]$, T is the vector of measured energy levels, Wk is the vector of weights and Tk is the decision threshold for the k-th subband. When $Wk > Tk \Rightarrow$ it means the k subband is occupied; on the contrary, if $Wk \leq Tk \Rightarrow$ it means that k subband is free.

The proposed scheme has been mapped to a simulation set up as depicted in Figure A.17 (right), first of all there is a network of sensing nodes which scan the different bands of the spectrum and also send all their measurements to a server containing a data base. Besides, there are some nodes which act as secondary/opportunistic users of the spectrum-holes. One of these nodes will act as the coordinator assigning free spectrum bands to the other ones, this coordinator node will connect to the measurements data base and will integrate the different measurements deciding which frequency bands are free and storing its decisions in the data base again.

Detailed evaluation results for this mechanism will be reported in [3].

A.19 CQI channels in OFDMA networks

Ensuring that the QoS requirements of each application are met under varying channel conditions, OFDMA networks adjust the MCS for every frame to the wireless channel condition of the intended receiver. When the channel condition is good, a more efficient MCS can be used. However, when the channel condition deteriorates, a more robust and less efficient MCS is appropriate.

To help the Base Station (BS) determine the appropriate MCS, every Mobile Station (MS) measures and sends CQI to the BS. The BS allocates a CQI channel for every active MS. The CQI bandwidth is a scarce resource, whose allocation must be adjusted to the actual needs of the MSs. However, allocations and de-allocations of CQI channels require expensive signalling messages between the BS and each of the MSs, and therefore should be minimised. The goal is to improve efficient allocation and bandwidth utilisation of the CQI channel, for each active MS in an OFDMA network.

The proposed mechanism was described and mapped into the elements of the OConS architecture in Deliverable D-4.1 [1] (Section 7.1.2). Our work addresses the allocation of periodic CQI feedbacks by the BS. We define an allocation framework, in which collisions and fragmentation are not allowed. Every active MS is entitled to its minimum demand CQI channel before any other allocation, and we rely on a function that quantifies the profit of the system from any allocation.

The details of our approach are published in [18]. In summary, we defined a power-of-2 CQI channel allocation, which is meant to prevent collisions between two different CQI channels (i.e. contain the same slot): rather than using a CQI slot in each frame, our scheme uses only the slots in a power-of-2 frames. A power-of-2 allocation is performed over a complete binary tree, referred to as a CQI allocation tree, while the bandwidth of each CQI super-channel is maintained (the CQI super-channel is a fixed time slot in every frame which is used for CQI reports). The allocated nodes are then assigned with the fraction of the super-channel bandwidth that is assigned to the corresponding CQI channel. Different bandwidth requirements can be assigned to different MSs by means of different tree levels. Further, our scheme does not allow CQI channel fragmentation, namely, when an MS is allocated 2 different tree nodes, thereby avoiding a non-optimal allocation.

We then address the following 3 problems with specific algorithms that optimise the allocation process:

- How to allocate channels to the MS when the tree (super-channel) is empty
- How to reallocate the bandwidth of a released channel to some unsatisfied MSs
- How to change the bandwidth of a CQI channel due to changes in the profit values of an MS. Such changes are likely to be consequence of new mobility patterns. The algorithm minimises the amount of signalling messages required for the bandwidth re-assignment.

As any other OConS mechanisms, our proposed mechanism is orchestrated by the orchestration functionality. During the bootstrapping phase, it is discovered, registered in the OR, and launched (if configured to be launched automatically). At runtime, the orchestration function utilizes the OSAP interface for communicating with the user (request, status). Our optimised CQI allocation mechanism operates at L1-L2 protocol layers of any OFDMA-based wireless networks. As such, it might be transparently included as part of the two OConS for CloNe and OConS for NetInf use cases. Its execution has to be coordinated by the SOP orchestration module, considering the availability of the following other OConS mechanisms:

- Dynamic Radio Resource Allocation for Virtual Connectivity, A.16. Our proposed mechanism operates at the physical level, while the A.16 mechanism operates at the virtual level
- Radio Resource Management for Wireless Mesh Networks, A.17 mechanism is handling multiple wireless, while our proposed mechanism optimises one interface. The activity of those two mechanisms should be coordinated.

Otherwise, the CQI channel mechanism does not collide or substitute any other OConS mechanism.

A.20 Multi-Path Benchmarking: the Trade-off Between Control Plane Load and Data Plane Efficiency

Software Defined Networking SDN, is a new emerging network architecture built on the following main concepts [66]:

- Separation between data and control planes
- Flow-based datapath, where flows (not packets) are the fundamental unit of control
- A control protocol, such as OpenFlow [67], between a logically centralized controller and the switch flow tables for programming and controlling the datapath
- A means to provide network virtualization by slicing the network and isolating the slices

Using these concepts, SDN allows service providers and enterprises to create infrastructure-as-a-service models by enabling on-demand procurement, provisioning and configuration of these services. New instances of network services can be established very quickly, and capacity can be scaled up or down in a real time.

One of the main promises of SDN is that it allows an efficient convergence between circuit switching and packet switching [68]. This is because the network operator can easily establish, modify and take down circuits, based on the requirements of the packet switching layer. However, this convergence imposes a trade-off between the control plane load and the data plane efficiency. Suppose that there is a demand for a certain bandwidth between two nodes in order to deliver a traffic aggregate ("traffic flow"). Thus, the SDN controller needs to establish one or more paths (circuits) that can deliver the required bandwidth with minimum data plane cost. However, this incurs a load imposed on the control plane by path setup (e.g. message exchange between OpenFlow controller [67] and every node along the path), and Operations And Maintenance (OAM) demands (for fast detection of data plane failures on the path, which translates into OAM protocols continuously executed along each path).

We study this tradeoff in two different optimization problems, both aiming at minimizing the control load:

- The controller is given a flow that satisfies the bandwidth demand between the source and destination nodes. This case is appropriate for operators that are primarily concerned with data plane efficiency
- Only bandwidth demand is given, and a set of paths must be found by the controller. This case is more appropriate for operators that are more concerned with control plane load

In both cases, we use the number of paths or the number of nodes traversed by the path as the primary factors for the control load.

It is important to note that the results of this research are applicable not only to SDNs with a centralized controller, but also to more traditional virtual circuit technologies, such as MPLS and Generalized Multi-Protocol Label Switching (GMPLS), in which paths are set up in a distributed manner. Using routing protocols such as Open Shortest Path First - Traffic Engineering (OSPF-TE) and Multi-Protocol Label Switching - Traffic Engineering (MPLS-TE), each router constructs a map of the network topology and the bandwidth available on each link. Then, each ingress router finds a route with sufficient bandwidth to an egress router for each traffic aggregate. The ingress router then establishes an MPLS Label Switched Path (LSP) over this route using a signaling protocol, such as Resource Reservation Protocol - Traffic Engineering (RSVP-TE). Each router along the LSP must keep a state of this LSP. This state must be periodically refreshed using the signaling protocol and maintained using the OAM protocols. Therefore, it is desirable to minimize the number of LSPs or the number of routers crossed by these LSPs in order to reduce the control load.

The mechanism studied here is not meant to be executed over the real-time OConS network infrastructure. Our mechanism is an optimisation study that is executed over a simulated or experimental network, in order to gain better understanding regarding the cost and benefits of multi-path operation. The results of our study can be used as benchmarking that guides network operators with regards to the extent at which multi-path is beneficial. As such, the proposed mechanism does not need to be orchestrated, and is not directly part of the proposed use cases (although, indirectly, its guidance enable more efficient configuration, with higher throughput and congestion-avoidance).

The objectives and scope of this research were initially reported in Deliverable D-4.1 [1] (Section 5.2.5). Since then, we widen the scope of the problem. The details of this study are available in [69]. The reminder of this section provides a summary and applicable conclusions. Our work further illustrates and formally defines both optimization problems, discuss its computation complexities and present various approximation algorithms with different approximation ratios and worst-case performance guarantees. For both problems we present two variants, one which minimizes the number of paths, and the other that minimizes the number of nodes traversed. The actual performance of these algorithms is examined through simulations.

The data plane efficiency problem with minimum number of paths is classified as NP-hard in a strong sense, thus a pseudo polynomial algorithm that finds the optimal solution is unlikely to exist. We showed that a simple greedy decomposition algorithm have an approximation ratio that is independent of the size of the network.

When looking at same problem while minimizing the number of nodes, the problem is shown to be NP-complete. We identified an approximation algorithm for it, reaching the same approximation ratio.

The control plane load minimization with minimum number of paths problem complexity is NP-complete. We designed two approximation algorithms and identified their computational complexity and their worst-case performance guarantees. Via simulation, we found that the actual average performance of those algorithms is not good. We therefore designed a third algorithm that does not have a worst-case performance guarantee, but its actual performance is shown to be very good. This algorithm separates the path flow finding procedure from the flow decomposition stage (the latter breaks the found path into minimum number of paths).

The node minimization flavor of the same problem is also shown as NP-complete. We showed that the approximation ratio for this problem is arbitrarily smaller than the approximation ratio for the path minimization flavor. We then designed an approximation algorithm and identified its value and cost. Similar to the path minimization problem, we found via simulation that the actual average performance is not good, and thus devised a new algorithm with no worst-case performance guarantee, but with improved average performance.

With Simulation, we aimed at comparing the various algorithms, using topologies generated with BRITE [70], as well as real ISP topologies from the RocketFuel project [71]. We identified the algorithm that minimizes the number of paths decomposed for any given bandwidth demand, which also means that the network flow is constructed faster and with fewer iterations. The number of decomposed paths increases linearly with the bandwidth demand, and also increases when the distance between the source and the destination grows.

We identified the algorithm that minimizes the number of nodes traversed by the decomposed path. The algorithm that decomposes the flow into the smallest number of paths traverses about 20 percent more nodes than the one that minimizes the number of nodes. When the number of nodes is of primary concern, is better to choose more paths that are as short as possible, rather than fewer paths, where each of which is of higher capacity.

We then examined the tradeoffs between the bandwidth cost of the network flow and the load imposed on the control plane for setting up and maintaining the paths satisfying the needed flow. SoTA algorithms [72] that demonstrates minimal bandwidth cost, produces a large number of paths. Our algorithm that minimizes the number of paths demonstrates 50 percent more bandwidth cost. However, one of our algorithms demonstrates a preferred tradeoff, with only 10 percent additional bandwidth cost (as compared with the SoTA algorithm), and only 5 percent more paths (as compared with our best minimal path algorithm). One of our algorithms that minimizes the number of nodes, also shows bandwidth cost that is almost as small as that of the SoTA [72].

The results of this research can be used as guidance and benchmarking for operators that deploy multi-path procedures, by looking not only on bandwidth cost, but also minimizing the control plane load. Operators can select the algorithms that optimize their objectives: minimum control load or most efficient data plane. Our work provides a benchmark for the optimal number of paths for any given bandwidth demand or for the distance between source and destination. For minimal number of nodes, operators are suggested to select a larger number of shortest possible paths, rather than fewer paths of higher capacity.

A.21 Resource Management within Heterogeneous Access Networks Benchmarking analysis

In this annex we describe two benchmarking studies which were made with the main goal of finding optimum resource management strategies to be used by network operators (i.e. base stations and access points) within wireless heterogeneous access environments. Both of them are based on game theory techniques. The first one focuses on resource allocation strategies while the second one studies price management policies. Both of them are fundamental research studies and one of the learnt lessons is the high complexity of the corresponding models. The reader might refer to [73, 74] for a more thorough description of these two studies. The ultimate goal was to establish the strategy which shall be adopted by the access elements so as to obtain the highest benefit.

We consider an area with N access elements, characterized by their coverage and capacity. We use a generic and discrete load (and capacity) unit, no matter it refers to time slots (TDMA systems), codes (CDMA systems) or sub-carriers (OFDMA systems). Two different deployment (random and deterministic) strategies were used, while end-users were randomly deployed within the area. The positions of both the access elements and the users lead to the establishment of a set of m areas, which are characterized by the overlapping of the access elements' coverage (provided that there is at least one end-user within them).

In the first one [73], we assumed that they were able to allocate their resources amongst the areas in which they have coverage, and the objective was to maximize the number of connected users. A user who was able to connect to more than one access element decides the one to which he will try to connect, either randomly or using other decision parameters (price and link quality). Based on the access selection strategy, we can establish the expected benefit for the various strategies which can be used by the access elements, so as to pose the corresponding game. We compare the optimum strategy (Nash Equilibrium Point (NEP) of the corresponding problem) with a naive one, in which access elements do not consider any particular resource management strategy and they just handle connectivity requests until their capacity is filled.

The second one [74] focuses on price management strategies. In this sense, the access elements do not allocate their resources between the areas over which they have coverage, but they fix their price, selected between a discrete number of choices. Besides, users always try to connect to the cheapest alternative. The procedure is quite alike, although in this case obtaining the expected benefits is more complicated. The *naive* strategy assumes that all access elements fix the same price.

As mentioned earlier, this analysis qualifies as a benchmarking study, which might be carried out in order to assess which might be the optimum performance which might be obtained by some of the mechanisms which might be implemented within the OConS framework. In particular it focuses on mechanisms which might be at either the network or the link level, used by the networks so as to improve their performance. In this sense, the results of this study can yield the performance of such optimum strategies. In this sense, the main idea behind these analysis is to identify the best performance which might be expected from such mechanisms and the strategy which might be used so as to foster it. The corresponding DEs would use these results so as to tune their corresponding algorithms, provided that the required IEs are available.

A.22 Price based load balancing for wireless access networks

By exploiting the *Enhanced Access Selection Mechanism* (see Annex A.3), the main goal was to assess the feasibility of a distributed load-balancing scheme based on pricing incentives from the base stations. The main idea behind this strategy is that the operators are able to encourage or deter the users to connect to an specific base station, according to its current load, by means of the offered price. We assumed that users are not subscribed to any operator but they are able to select the best (in terms of price) access without taking into account the operator that owns the access.

With the above in mind, the study comprises two different aspects. On one hand, the strategy followed by the base stations consists in a piecewise decreasing function which adapts the offered price to the currently carried load, so encouraging or deterring the users to perform a connection accordingly to low or high network load (this could be seen as an OConS mechanism, in which the DE is in charge of establishing the fee, based on the inputs provided by the relevant IEs (in this case, the load of the corresponding base stations). On the other hand, we analyzed different selection strategies (exploiting the capabilities of the *Enhanced Access Selection Mechanism*. In particular, three parameters have been considered: the first one prioritizes the cheaper access alternatives; the second parameter gives a higher score to lower loaded base stations; finally, the third one fosters the base station currently being used, as a means to integrate the cost of change, thus reducing the number of handovers. In order to sort the available access alternatives, a linear combination of these parameters was used.

As can be seen from the previous discussion, both the enhanced access selection mechanism and the pricing policy implemented by the base stations exploit the possibilities of the OConS. The former takes the price offered by the base stations (corresponding IE) as another parameter within the selection process, while for the second one, the base stations need to collect information about their current load.

The SOP within this OConS domain would orchestrate the two mechanisms which, albeit rather orthogonal, shall be used simultaneously; the base stations would not achieve any load balancing by tweaking the offered price if the end-users are not really considering the price of the connection as a parameter within the access selection process.

B A consolidated view for the OConS Information Model

This section presents the overall OConS Information Model, providing the suitable level of abstraction for the concept, such as naming, types, main attributes, but not fully specifying all the envisaged details. The model is split into the figures Figure B.2 and Figure B.1. That Information Model described in UML is complemented with further details and attributes on page 114.

The listing presents attributes and potential values for a selection of OConS concepts:

Domain
- id/naming
- participant nodes (see OConS.Node for details)

OConS Node (both network-side/terminals, both nominal and currently available and used)
- id/naming
- Service Orchestration Process (SOP) attributes (see OConS.SOP for details)
- link interface related attributes (see OConS.Interface.Resources for details)
- network related attributes (see OConS.Network.Resources for details)
- flow related attributes (see OConS.Flow.Resources for details)
- caching resources (hdd, flash)
- processing resources (cpu, ram)
- other resources (power/battery, screen, contact/neighbour history, certificates/keys for SA, etc.)

SOP
- id/naming (e.g., if we need to register them in a DNS-like system)
- mechanisms it orchestrates (see OConS.Mechanisms.Capabilities for details)
- services it has generated for each applicative flow (e.g., it can be a set of mechanisms ids/names)

OConS Registry
- entities, mechanisms, and services it contains in each of those

OConS Entity
- id/naming and type (IE,DE, or EE)
- capabilities (IE what is measured, EE what is actuated, DE what it decides)

OConS Mechanism (see also the Mechanism Manifest from Table 3.1)
- their ids/category
- the needed IE/DE/EE entities
- what it guarantees under which constraints

OConS Service
- link, network and flow services, see next

Link Connectivity Services (i.e., Layers 1/2)
- id/naming
- type (umts, lte, wifi, bluetooth, etc.)
- status (available, not available, connected/not-connected and with what access/operator)
- typical/max/guaranteed data-rate, loss, delay, jitter, radio-related, etc.
- mobility events history
- mechanisms available and running (see OConS.Mechanisms)
- available bandwidth
- supported protocols

Network Connectivity Services (i.e., Layers 3/4)
- id/naming
- type (IPv4, IPv6, MPLS, OF, UDP/TCP/SCTP/DTN)
- mechanisms available and running (see OConS.Mechanisms)

Flow Connectivity Service
- id/naming for that flow (including, e.g., the application id and the user id)
- requested/current/maximum QoS
- mechanisms available and running for that flow (see OConS.Mechanisms for Layers 1/2, Layers 3/4, and upper)
- lifetime of the flow

OConS Operator
- orchestration rules (e.g., the mechanism(s) it complements or works with, it substitutes, or collide with)
- maximum load for a node/link
- preferred access for an application/flow
- preferred Data-Centre/CDN for an application/flow
- maximum QoS for a user/application/flow (usually the maximum bit-rate)

OConS Client (i.e., Application or User requests through the OSAP, see Table 3.2)
- requested QoS / QoE (also per-flow requirements, if that is the case for a given application)
- preferred interface for an application/flow (e.g., wildcards to set it for all applications)
- preferred operator (it can be per-application, and so on)
- maximum price for an access (it can be per-application, and so on)

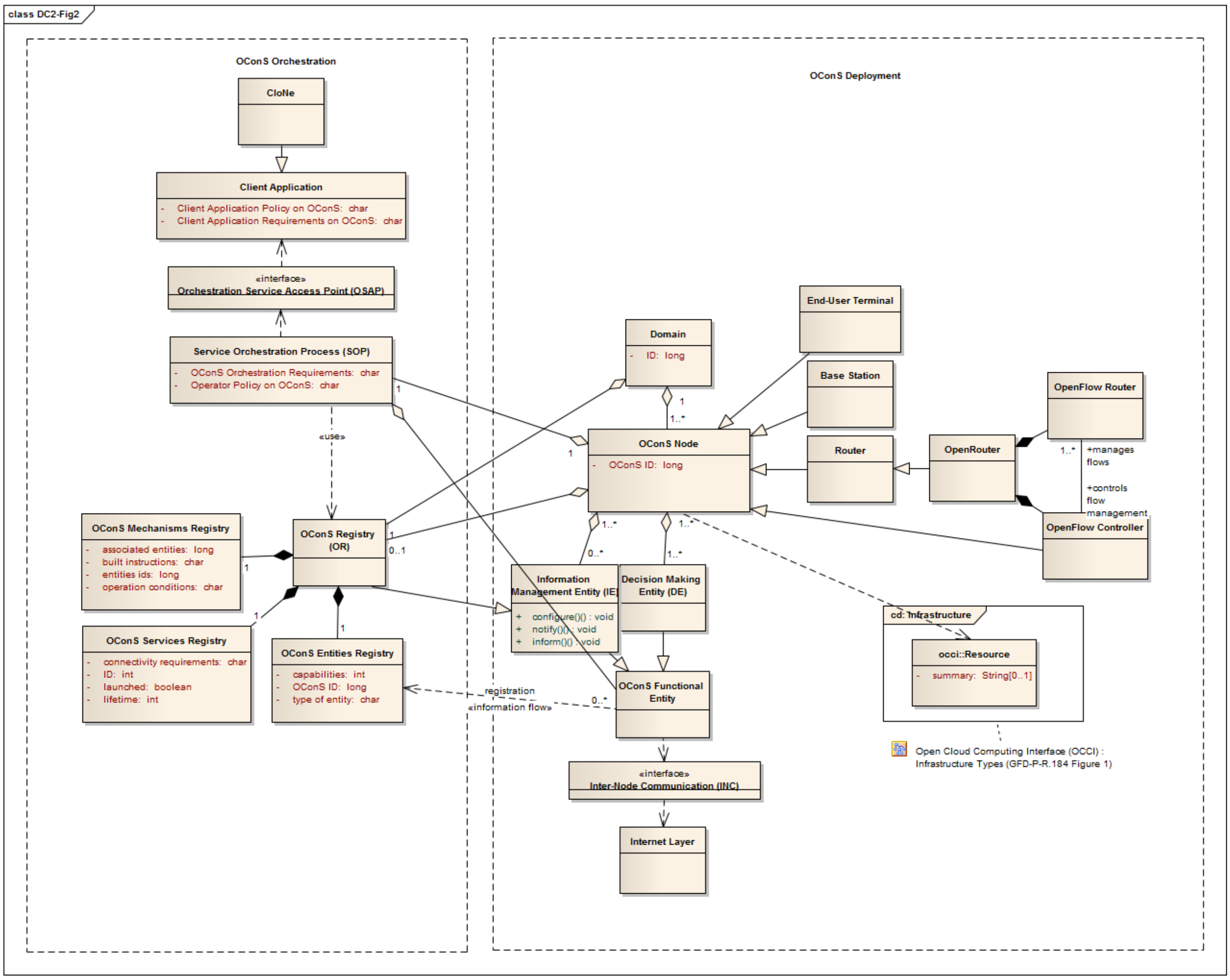


Figure B.1: OConS Information Model: Orchestration and Deployment

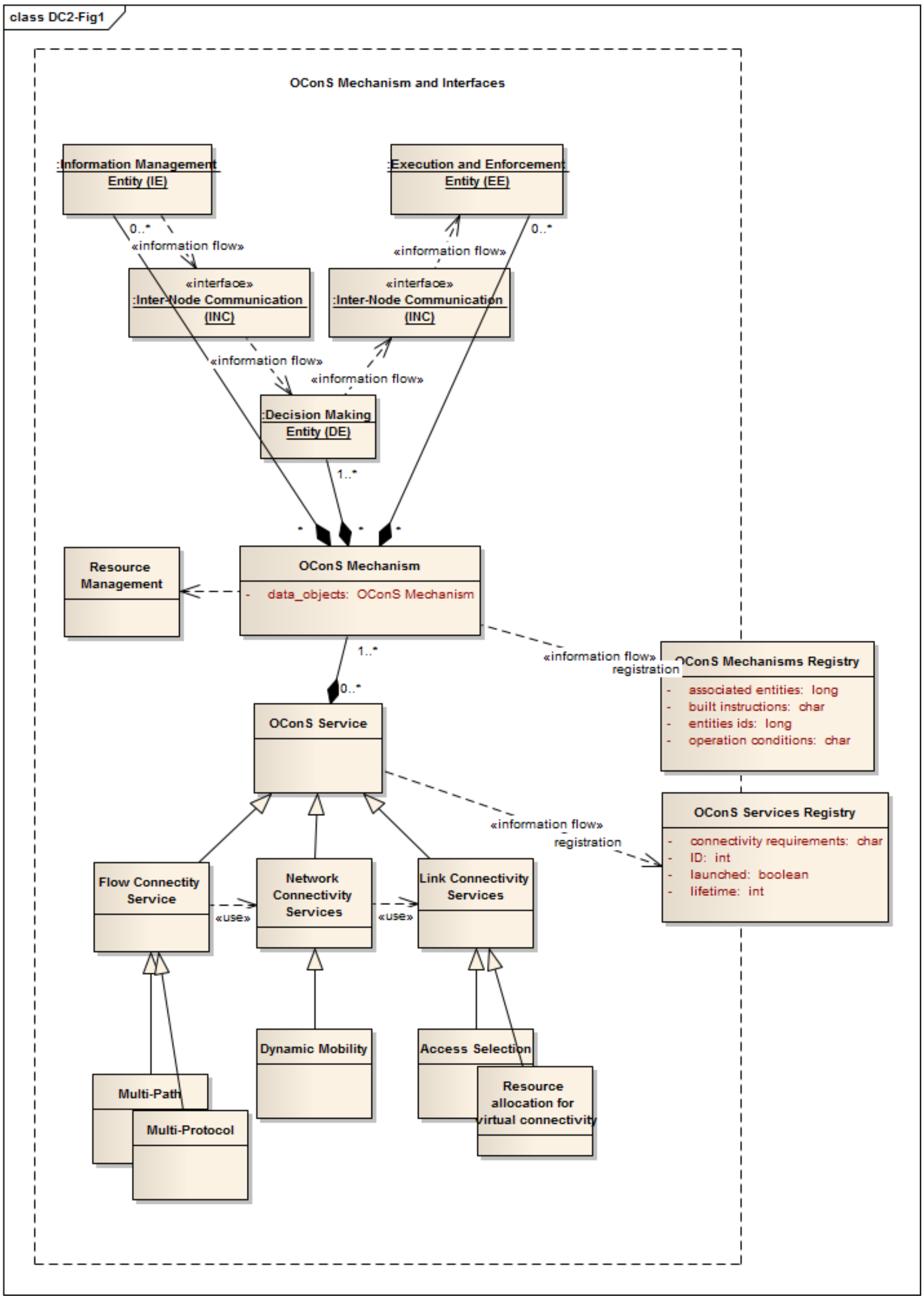

class DC2-Fig1
OConS Mechanism and Interfaces
:Information Management Entity (IE)
:Execution and Enforcement Entity (EE)
0..*
«information flow»
0..*
«information flow»
«interface» :Inter-Node Communication (INC)
«interface» :Inter-Node Communication (INC)
«information flow»
«information flow»
:Decision Making Entity (DE)
1..*
*
*
*
Resource Management
OConS Mechanism
- data_objects: OConS Mechanism
1..*
0..*
«information flow»
registration
OConS Mechanisms Registry
- associated entities: long
- built instructions: char
- entities ids: long
- operation conditions: char
OConS Service
«information flow»
registration
OConS Services Registry
- connectivity requirements: char
- ID: int
- launched: boolean
- lifetime: int
Flow Connectity Service
«use»
Network Connectivity Services
«use»
Link Connectivity Services
Dynamic Mobility
Access Selection
Resource allocation for virtual connectivity
Multi-Path
Multi-Protocol

Figure B.2: OConS Information Model: Mechanisms and Interfaces

B.1 OConS relation to CloNe

For the data-centre use case, OConS uses type definitions that are also employed by CloNe. This approach has been chosen to guarantee a better support of the on-demand management and control of the resources by the OConS connectivity service. CloNe uses the Open Cloud Computing Interface (OCCI) Core [75] and Infrastructure Model [76] and additionally introduced an extension of OCCI, the Open Cloud Network Interface (OCNI) Model. Details for the use in CloNe are described in D.D.2 [20].

The OCCI has been specified to provide a high-level definition of the OCCI RESTful Protocol and an API for all kind of management tasks with a strong focus on interoperability and extensibility.

One main element of the OCCI Core Model is the Resource type. It has been defined to represent real world resources. Beyond this the OCCI Core Model defines Mixins for extensions. They only apply at instance level and allow to dynamically mix-in new resource attributes at creation and/or run-time. The OConS Node class uses OCCI Core types Resource and Mixins in the context of data-centre interconnect. While the Resource type is included in the OConS information model, the Mixin type is not shown in the diagram.

The OCCI Infrastructure Model adds amongst others the children of the Resource type Network, Compute and Storage.

The OConS Information Model depicts this situation by linking to the OCCI Resource object on the right hand side of the diagram of Figure B.1.

The Network type represents L2 networking resources with attributes, defining VLAN identifier, tags and state information.

The Compute type represents a generic information processing resource and specifies attributes describing the CPU architecture, the number of CPU cores, the hostname, CPU speed in gigahertz, maximum Random Access Memory (RAM) in gigabytes and the current state of the instance.

The Storage type represents resources that record information to a data storage device. The class comprises the storage size in gigabytes and the current state of the instance.

The OCNI [77] extends the OCCI specification by including a cloud networking centric extension and a number of specialized network Mixins, e.g. OpenFlow.

Especially the OpenFlow Mixins play an important role in the data-centre interconnect use case as it includes parameters that can be used to describe the characteristics of the DCU (i.e., domain centralized SOP) and the interworking between the OConS Nodes.

The OConS Information Model defines the OpenRouter class in order to include an extensible open source routing platform, which allows to dynamically introducing new protocols and functionalities. The OpenRouter combines the OpenFlow Controller for routing decision making and the OpenFlow Router for carrying out the decisions of the controller.

The newly defined OCNI Model describes the FlashNetworkSlice, representing the resource providing a network service and two further resources CloNeNode and CloNeLink. CloNeNode specifies a networking resource of the FlashNetworkSlice and CloNeLink a network link of the FlashNetworkSlice.

The CloNeNode and the CloNeLink type comprise valuable attributes that can also be utilized from OConS. CloNeNode defines the attributes availability of the entity, its location and the current state of the instance. The CloNeLink type specifies also availability and state. Beyond this it defines attributes needed for resource management decisions like bandwidth, latency, jitter, loss and the transmission type of the instance.

The following Table B.1 gives a 'pseudo-formal' example of the resource demand profile for an OConS service as requested by CloNe. The actual encoding at the CloNe–OConS (northbound) interface can be realised in JavaScript Object Notation (JSON) or eXtensible Markup Language (XML).

Field/Object	Description
list of *link_descriptions***:**	defines the (link) connectivity of OConS to CloNe attachment points (i.e., PE to CE)
• *link_id*	CloNe id for data link
• *remote_attachment_point_address*	CloNe address (IP address, VLAN id, etc.) for data link connection
• *link_attributes*	QoS requirements such as bandwidth, delay, jitter, etc.; link protocol (mechanism) to be used (IP, VLAN, etc.)
• *remote_controller_address*	IP address, port number for CloNe DCP peer
list of *forwarding_rules***:**	defines CloNe 'single router abstraction' of connecting network
• *source_network_address*	(optional, for source routing) e.g. IP address, network mask , or OpenFlow address-tuple
• *destination_network_address*	e.g. IP address, network mask , or OpenFlow address-tuple
• *next_hop*	*link_id* (as defined in *link_descriptions*) for forwarding
• *network_attributes*	OConS network mechanisms to be activated on per-link basis (e.g. performance monitoring, traffic measurements, address resolution, network address translation etc.), incl. their configuration parameters
list of *flow_descriptions***:**	defines flows between connected CloNe attachment points to be established and maintained by OConS
• *source_attachment_point*	CloNe id (as defined in *link_descriptions*)
• *destination_attachment_point*	CloNe id (as defined in *link_descriptions*)
• *flow_attributes*	OConS flow mechanisms to be activated on per-link basis (e.g. bandwidth / QoS guarantees, performance monitoring, traffic measurements, etc.) incl. their configuration parameters

Table B.1: Example of CloNe Resource Demand Profile

C Details on OConS Interfaces

C.1 Message Structure

OConS messages are essentially TLVs with a common header. This allows for a compact on-wire formulation without loosing much expressive power nor extensibility[1].

C.1.1 Common Packet Header

The OConS message header contains mostly the destination and source node and entity IDs, as well as additional fields to identify and classify the message and support later evolution of the protocol. It is described in Table C.1.

The `Version` field is present to support evolution of the OConS messaging protocol. It must be set to 0x01, and any packet received by an entity with a version different than 1 should be discarded. The `Flags` provide a similar mechanism with a finer granularity. They allow intra-version extensions of the protocol. As no such extension currently exist, this field must be set to 0x00, and ignored by the receiver.

The OConS IDs are split in their two components, and distributed across the header. This follows a design decision driven by the observation that both components are not manipulated at the same time and source/destination couples are more relevant to be grouped together. This scheme thus allows the possibility of bundling messages to/from different entities hosted on the same destination/source nodes to reduce header overhead on the underlying transport. This can be done by splitting the header of Table C.1 after the `SrcNodeID` field (after the 38th byte). The higher part thus becomes an envelope header for the packet, while the lower part is repeated as a header for each bundled message.

The TLV-formatted message then follows. Its generic format is a 1 B `MsgType` allowing to identify the format of the following data, and a 2 B `MsgLength` which gives the size (in bytes) of the remaining message, starting with the `MsgType` field. Regardless of their type, each message contains an `MsgSeq` counter which allows to uniquely identify subsequent messages. The rest of the message, of length `MsgLength`-4 B, then contains one or more TLV elements further specifying the message, and containing relevant information.

C.1.2 TLV Format

An OConS message is composed of one or move TLV-formatted data elements. The format of OConS TLVs is shown in Table C.2. It is composed of the following fields.

- `Type` is a 16-bit field which uniquely identifies the TLV, and
- `Length` is a 16-bit field indicating the number of bytes in the current TLV, including the type and length (*i.e.*, from the MSB of the `Type` to the LSB of the value).

[1] The messages are encoded in such a way that other entities overhearing them can gain up-to-date information without knowing the context of their transmission, in a way inspired from Publish-Subscribe systems, in order to limit on-link signaling overhead.

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">`Version` (1 B)</td><td colspan="16">`PktLength` (2 B)</td><td colspan="8">`Flags` (1 B)</td></tr>
<tr><td colspan="32">...
`DstNodeID` (16 B)
...</td></tr>
<tr><td colspan="32">...
`SrcNodeID` (16 B)
...</td></tr>
<tr><td colspan="16">`DstEntId` (2 B)</td><td colspan="16">`SrcEntId` (2 B)</td></tr>
<tr><td colspan="8">`MsgType` (1 B)</td><td colspan="16">`MsgLength` (2 B)</td><td colspan="8">`MsgSeq` (1 B)</td></tr>
<tr><td colspan="32">...
message (`MsgLength`-4 B)
...</td></tr>
</table>

Table C.1: Common OConS header encapsulating an example TLV message. The packet's total length `PktLength` is `MsgLength`+36 B.

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="16">`Type` (2 B)</td><td colspan="16">`Length` (2 B)</td></tr>
<tr><td colspan="32">...
value (`Length`-4 B)
...</td></tr>
</table>

Table C.2: Format of an OConS TLV.

C.2 Message Taxonomy

As we have already mentioned, there are three main types of OConS messages:

Requests are unsolicited messages, they elicit a response from the receiver;

Responses are sent as direct replies when requests are received;

Notifications are unsolicited response-like messages; they can be sent, *e.g.*, periodically or when a specific event happens.

Additionally, four semantic classes of messages are envisioned, depending on their role within OConS:

Entity-handling messages are used by the orchestration functionalities as to initially register entities and identify the available mechanisms (inspired from the IEEE 802.21 standard messages and operations, see [11]);

Publish/Subscribe messages are exchanged within the OR to handle mechanisms and services (i.e., based on the publish/subscribe approach with its benefits);

Mechanism-handling messages are sent by the SOP to instantiate and enable the required mechanisms (inspired from the CRUD operations used in web services);

Inter-entity messages are exchanged between the entities, usually involved in the same mechanisms.

C.2.1 OConS Messages and their mapping on Logical Interfaces

This section specifies in more details the messages used in OConS.

C.3 Bootstrapping

The bootstrapping phase is the initial one that every entity goes through. It is used to register the local entities to the local SOP, if one exists. Then, it allows to support remote discovery and discovery, either of the needed entities to support a mechanisms or by an orchestration process looking for the last missing components.

During the local registration, each entity uses a well-known or OS-implementation specific local method (*e.g.*, a named Unix socket) to connect to the INC. At this time, the entity receives a fresh entity ID which will identify it during the current session, and is registered to the OR. The local SOP can then query the newly-spawned entity for its capabilities. This is illustrated in Figure C.1).

Once registered, an entity is also available for remote discovery. Entities or SOPs running on remote OConS nodes can discover them by sending discovery messages, as illustrated in Figure C.2.

When inter-node communication is involved, especially in a multi-domain environment, the use of an IP network to address and interconnect the nodes can be envisaged.[2] [3]In this case, *prior to actual OConS's orchestration bootstrapping process (sec. 3.7.3)*, a Domain Name System (DNS))can be used to discover IP addresses of other OConS nodes, starting from a well-known Fully-Qualified

[2] This could be a private IP network, built to enable communications between OConS nodes of specific domains only, or it could simply be the Internet. However, a managed DNS infrastructure is required, to apply this discovery mechanism.

[3] If an IP network is not available, other link layer transport can be envisaged to support this remote discovery, e.g. similar to the Generic Advertisement Service (GAS) from IEEE 802.11u.

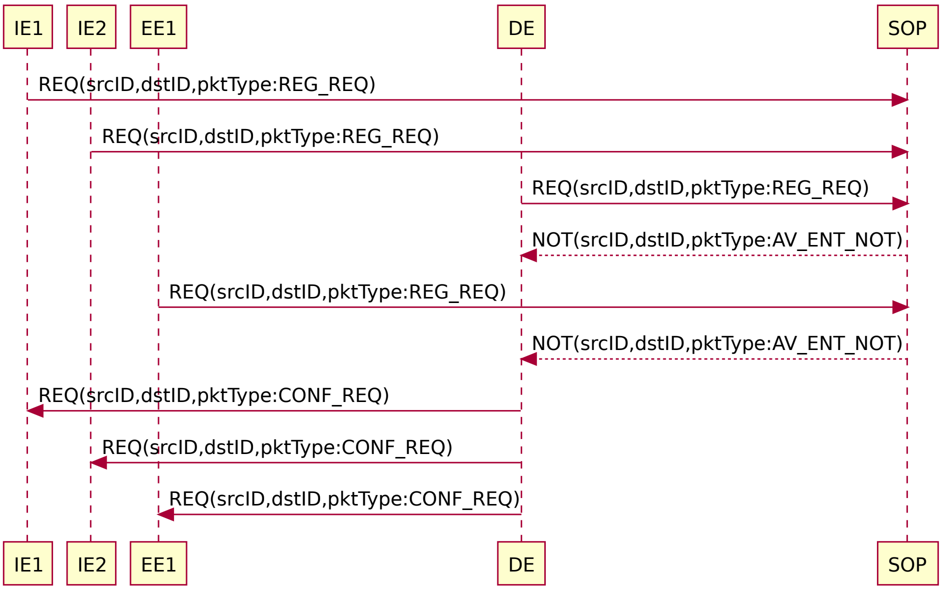

IE1
IE2
EE1
DE
SOP
REQ(srcID,dstID,pktType:REG_REQ)
REQ(srcID,dstID,pktType:REG_REQ)
REQ(srcID,dstID,pktType:REG_REQ)
NOT(srcID,dstID,pktType:AV_ENT_NOT)
REQ(srcID,dstID,pktType:REG_REQ)
NOT(srcID,dstID,pktType:AV_ENT_NOT)
REQ(srcID,dstID,pktType:CONF_REQ)
REQ(srcID,dstID,pktType:CONF_REQ)
REQ(srcID,dstID,pktType:CONF_REQ)
IE1
IE2
EE1
DE
SOP

Figure C.1: Message flow during the local registration phase. Each entity registers to the local OR via the SOP, while the DEs get information about the relevant others and can configure them according to its need. Though all communication goes through the INC it is not semantically relevant, and is not shown here.

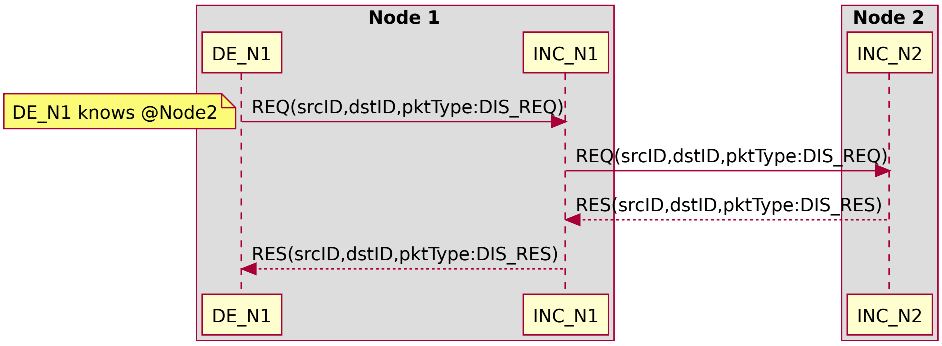

Node 1
Node 2
DE_N1
INC_N1
INC_N2
DE_N1 knows @Node2
REQ(srcID,dstID,pktType:DIS_REQ)
REQ(srcID,dstID,pktType:DIS_REQ)
RES(srcID,dstID,pktType:DIS_RES)
RES(srcID,dstID,pktType:DIS_RES)
DE_N1
INC_N1
INC_N2

Figure C.2: The complexity of remote discovery is abstracted by the INCs.

Domain Name (FQDN)[4]. Moreover, DNS enables the use of Resource Records (Resource Record (RR)) of different type, to discover more than just the IP address, thus facilitating the process of connecting to other OConS entities. For example, it can be used to discover if the remote node provides multiple instances of an OConS entity (for load balancing), or to discover which transport protocol and port (*e.g.*, TCP, UDP) are used to implement each OConS interface (*e.g.*, the DE-EE interface).

An optional usage profile for utilizing DNS to discover OConS entities is described in annex D. It makes use of `NAPTR` (Name Authority Pointer) and, optionally, `SRV` (Service) RRs and can be summarized as follows.

1. Node X wants to contact an INC hosted on a remote node Y, to send it an OConS Discovery Message;

2. Node X makes a query for an `NAPTR` (rather than `A`) RRs to a DNS server, specifying a well-known FQDN for node Y's INC, formed, for example, prepending the keyword "`ocons-inc`" to a domain name(*e.g.*, `ocons-inc.router-1.domain2.com.` or `ocons-inc.eth0.access-point-a1.domain3.com.`); the query can also be used to retrieve information about all INCs in a domain: `ocons-inc.domain1.com.`

3. The DNS server returns an answer that contains either of the following RRs.

   `NAPTR`, defining a list of of FQDNs that points to several IP addresses corresponding to several instance of the requested OConS entity

   `A` or `AAAA`, containing IP addresses of all the instances pointed by the `NAPTR` RRs.

   `SRV` (optionally), specifying protocols and ports used by the requested OConS entity

C.4 Security

The OConS protocol is the basic interface between entities, enabling and supporting all higher level functionalities such as orchestration and service composition. As such, it presents a very interesting target for an attacker. This section considers the attack surface exposed by the OConS protocols (as a subset of the threat model presented in Appendix E), derives which security properties should be provided, and details how they are.

Threat Model

All four aspects listed in Appendix E are relevant to the OConS interfaces. However, only a subset of the attacks listed there are valid. An attacker could obtain some control over the orchestration process or a given mechanism by:

- Impersonating a legitimate OConS node,
- Postponing or replaying OConS messages or,
- Modifying or forging messages.

Privacy is a prime concern, as OConS can be used to exchange potentially sensitive data. It should therefore be ensured that only the intended recipient(s) can access and use the data sent to them, and that any eavesdropper cannot. In addition a Denial of Service (DoS) against the INC would be an tempting attack angle that needs be protected against.

It is also important to consider the underlying protocols used by OConS. Indeed, gaining some control over these could eventually lead an attacker to compromise the OConS processes. These

[4] The INC in each node is responsible for mapping OConS ID to the FQDN names of remote entities. This mapping is implementation dependant and it is not specified in this deliverable

underlying protocols include TCP/IP, Ethernet, or even DNS in the case of the FQDN-based bootstrap. Requirements as to which security features are required for these as OConS building blocks is also addressed below.

Required Security Properties

In light of this threat model, the following security properties must be provided by the OConS interfaces in order to adequately protect this attack surface.

Integrity is a basic property required of any communication system. Though it does not provide actual protection against a motivated attacker, it is a necessary building block on which security mechanisms can be built.

Authenticity allows to mitigate the risks of impersonation and forged messages. Coupled with the integrity property, is also allows to alleviate the risk of modified messages.

Confidentiality is the key element in providing protection from eavesdroppers.

Availability the OConS elements realising the interfaces should remain available under load.

Timeliness of information is also needed to avoid any attack based on message reordering or duplication.

Security Mechanisms in the OConS Protocol

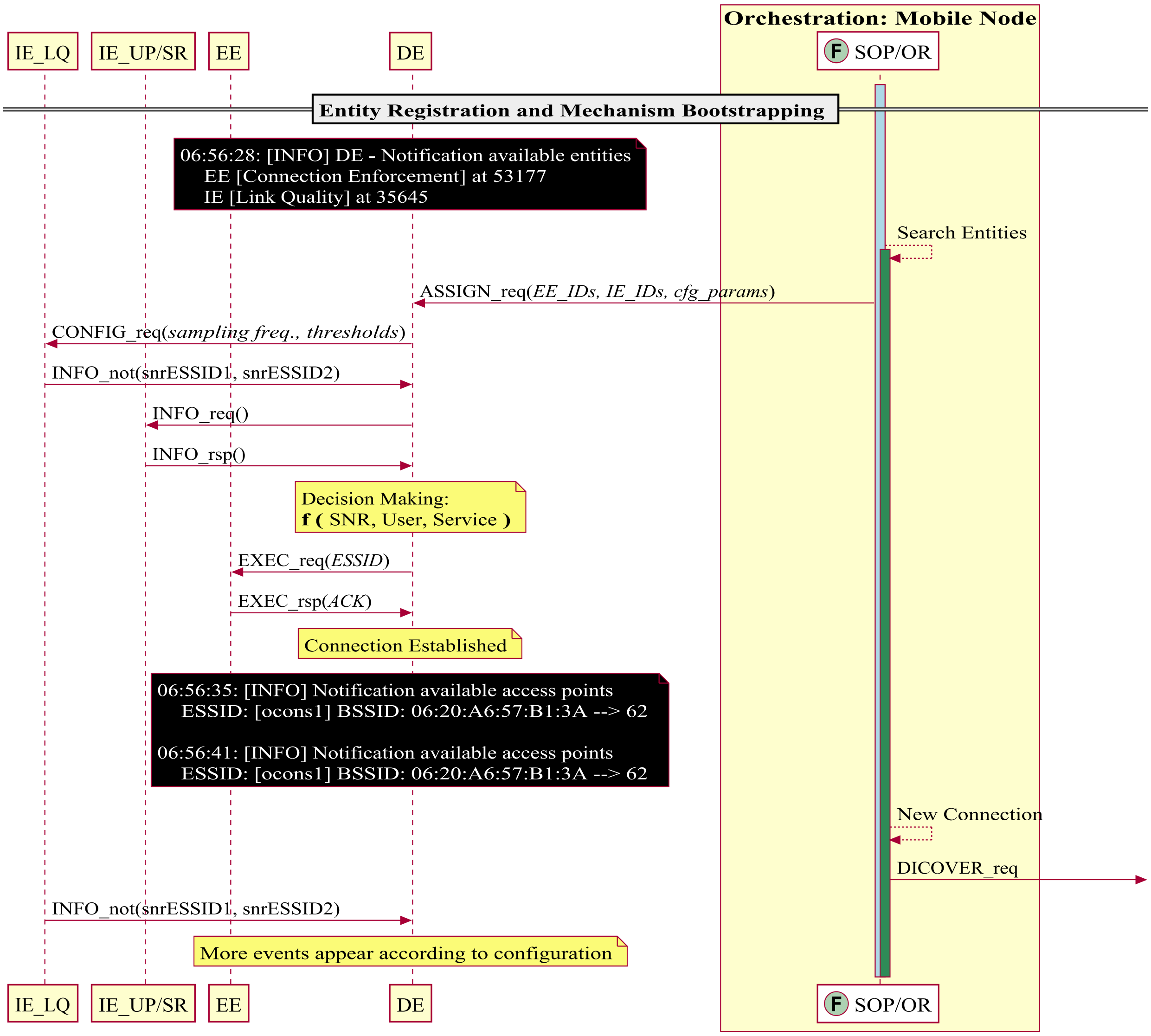
Additional TLVs are introduced here, which encapsulate messages from the entities. The INC is in charge of adding the necessary security encapsulation. OConS entities need not worry about the task. If a message from a remote source is considered correct, it will be passed on to the local destination entity. Otherwise, it will be silently discarded.

Integrity protection is provided in the form of a hash over the entire content of the *message* (which may comprise more than one TLV). To support hash-function agility, the TLV contains a field identifying the hash function in use. Authenticity (and integrity) are provided through the use of a Hash-based Message Authentication Code (HMAC). Similarly to the integrity protection, this HMAC covers the entire message, and supports hash agility in the same way. Confidentiality can be provided using a symmetric cypher over the entire message. As for the previous TLVs, algorithm agility is also supported.

For both the HMAC and encryption TLVs, keys are assumed to be already distributed. Key exchange and public-key cryptography could be supported by additional key exchange messages, by relying on Public Key Infrastructures (PKIs) or leveraging the DNS infrastructure (in a way similar to IPsec [78]).

Timeliness of the information is enforced by a Timestamp TLV attached to any message. To support proper time comparison, OConS nodes should comprise some mechanism to update their clock, such as Network Time Protocol (NTP) [79]. Security considerations for NTP are discussed in [79, sec. 15].

For DNS bootstrapping DNSSEC [80] can provide integrity and authenticity of the records. Its use is recommended within a single operator's infrastructure, and mandatory for inter-operator or customer–operator interaction.


Orchestration: Mobile Node
IE_LQ
IE_UP/SR
EE
DE
F SOP/OR
Entity Registration and Mechanism Bootstrapping
Search Entities
ASSIGN_req(*EE_IDs, IE_IDs, cfg_params*)
CONFIG_req(*sampling freq., thresholds*)
INFO_not(snrESSID1, snrESSID2)
INFO_req()
INFO_rsp()
Decision Making:
f (SNR, User, Service **)**
EXEC_req(*ESSID*)
EXEC_rsp(*ACK*)
Connection Established
New Connection
DICOVER_req
INFO_not(snrESSID1, snrESSID2)
More events appear according to configuration
IE_LQ
IE_UP/SR
EE
DE
F SOP/OR

```
06:56:28: [INFO] DE - Notification available entities
    EE [Connection Enforcement] at 53177
    IE [Link Quality] at 35645
```

```
06:56:35: [INFO] Notification available access points
    ESSID: [ocons1] BSSID: 06:20:A6:57:B1:3A --> 62

06:56:41: [INFO] Notification available access points
    ESSID: [ocons1] BSSID: 06:20:A6:57:B1:3A --> 62
```

Figure C.3: Example DMM message exchange.

C.5 Illustrative Examples

C.5.1 Distributed Mobility Management

The access selection mechanism from Appendix A.3 and Distributed Mobility Management (DMM) from Appendix A.11 have been combined through OConS [22]. The access selection logics ensure, through OConS-based interaction with the infrastructure, that the mobile device is connected to the best access network. Meanwhile, the DMM mechanism takes care of selecting mobility anchors to seamlessly forward flows belonging to sessions established on the previous access networks.

Figure C.3 shows an example message sequence of this integration, while Table C.3 lists some of the involved TLVs.

C.5.2 Optimisation of Mobility Parameters

In sec. A.12, a mechanism for MPO is described. An example of usage of OConS inter-entity message exchange, as defined in sec. C.2.1, is depicted in Figure C.4, which shows an INFORMA-

Name	Code	Type	Description
AccessNetworkID	TBD	`UTF8String`	Network Name (*e.g.*, ESSID)
RemoteInterfaceID	TBD	`NumericString`	Link Address (*e.g.*, BSSID)
SNR/SINR	TBD	`Unsigned8`	Link Quality
ContentType	TBD	`Unsigned16`	Application content type (*e.g.*, voice, video-streaming)
UserPrefNetworks	TBD	SequenceOf `AccessNetworkID`	Preferred user networks
FlowID	TBD	`Unsigned8`	Flow identifier for DMM

Table C.3: Example TLVs for the integration of the access selection/DMM mechanisms through OConS. Data types are defined according to [81].

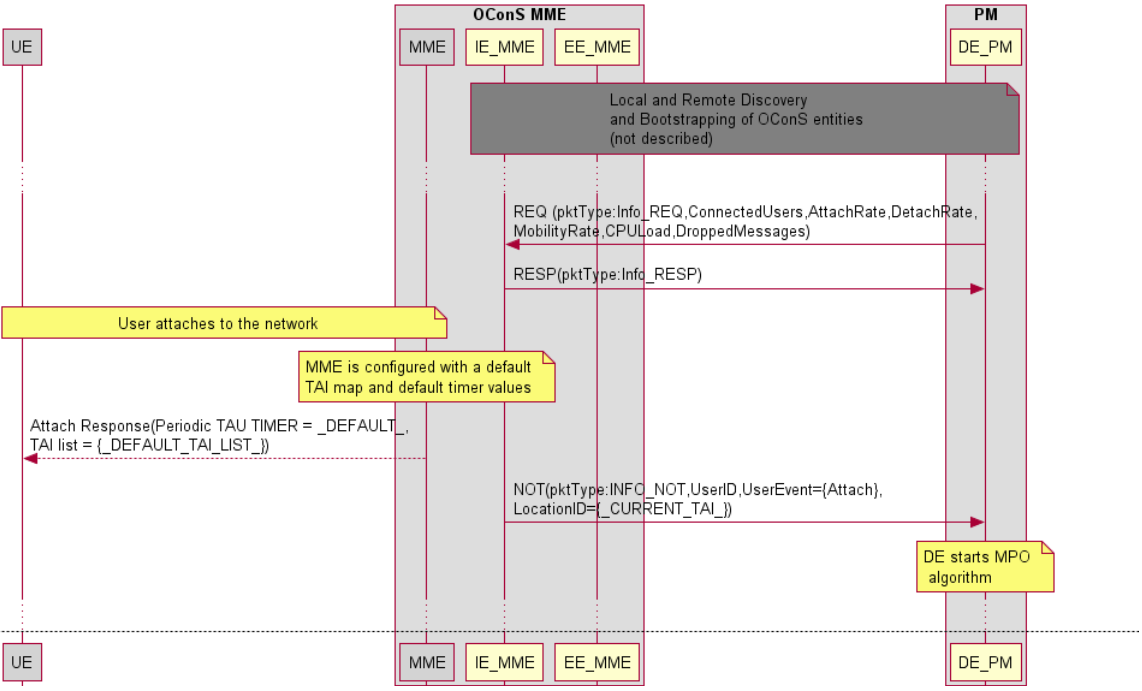

OConS MME
PM
UE
MME
IE_MME
EE_MME
DE_PM
Local and Remote Discovery
and Bootstrapping of OConS entities
(not described)
REQ (pktType:Info_REQ,ConnectedUsers,AttachRate,DetachRate,
MobilityRate,CPULoad,DroppedMessages)
RESP(pktType:Info_RESP)
User attaches to the network
MME is configured with a default
TAI map and default timer values
Attach Response(Periodic TAU TIMER = _DEFAULT_,
TAI list = {_DEFAULT_TAI_LIST_})
NOT(pktType:INFO_NOT,UserID,UserEvent={Attach},
LocationID={_CURRENT_TAI_})
DE starts MPO
algorithm
UE
MME
IE_MME
EE_MME
DE_PM

Figure C.4: Example of INFORMATION REQ/RESP in the MPO mechanism.

TION_Req/_Resp exchange, used to query OConS IEs periodically and an asynchronous INFORMATION_Notification message, used by IE to inform a DE about new information available (in this case, a mobility event).

Table C.4 shows the definition of the TLVs used in the example on the IE–DE OConS interface, according the format defined above.

Name	Code	Type	Description
UserID	TBD	`UTF8String`	User identifier (*e.g.*, IMSI)
ConnectedUsers	TBD	`Unsigned32`	Attached users
AttachRate	TBD	`Unsigned32`	Rate of Attach (msg/s)
DetachRate	TBD	`Unsigned32`	Rate of Detach (msg/s)
MobilityRate	TBD	`Unsigned32`	Rate of MM procedures (msg/s) *e.g.*, TAU, Paging, Handovers
CPULoad	TBD	`Unsigned32`	Percentage of CPU utilization
DroppedMsg	TBD	`Unsigned32`	Aggregate number of dropped msgs
UserEvent	TBD	`Enumerated`	Event that generated the message (*e.g.*, Attach=1, PTAU=2)
LocationID	TBD	`Unsigned32`	ID of the location of the user (*e.g.*, TAI, cellID)

Table C.4: Example TLVs for the IE–DE interaction in the MPO mechanism. Data types are defined according to [81].

D OConS Nodes and Entities Discovery with a DNS-based Mechanism

This section describes an usage profile[1] for Domain Name System, which enable OConS nodes, attached to an IP network to discover their IP addresses and other information relevant for the OConS bootstrapping procedure, as described in C.3.

D.1 Assumptions and limitations

Some working assumptions are necessary, in order to use the DNS to discover OConS entities. The main assumptions are that every OConS entity has an IP address (to perform DNS queries) and that every OConS entity has a valid FQDN, as specified in [82] and [83].

Moreover, a DNS server must be available, it must be configured on requesting OConS entities, it must be reachable and running a correct configuration of all the records, for all the nodes. This means that this mechanism is especially suitable for managed network environments, like an operator's network or a company network.

Therefore, it does not cover ad-hoc networks, which are often unmanaged. It also cannot be applied before having a valid IP address: for example, if a node wants to perform bootstrapping procedure with other link-local entities, before having an IP address assigned, it must do so using link-local lower-layer specific mechanisms, not described here[2].

D.2 Description of NAPTR and SRV usage

A description of the RRs needed is contained in [84], which defines a mechanism called S-NAPTR (Straightforward NAPTR). In short:

NAPTR NAPTR is a very powerful record, that allows to re-write the searched FQDN into one ore more other FQDNs, each specifying a specific 'service' provided by the original FQDN

SRV enables specification of a specific protocol and port for each of the services defined by the NAPTR records

An example, taken from [84] can clarify the use of NAPTR and SRV. The example shows how to discover an EM (Extensible Messaging) service[3].

```
{
Thus, to find the EM services for thinkingcat.example, the NAPTR
   records for thinkingcat.example are retrieved:

thinkingcat.example.
```

[1] With usage profile, we mean an OConS-specific configuration of a DNS server. In this sense, this section does not describe any new mechanism, but a way of using DNS in the context of OConS

[2] A possible way to re-use DNS in this context is using Local DNS, LDNS, but it is out-of-the-scope of the current document

[3] The example follows the syntax of configuration files of BIND, a largely used open-source DNS server

```
;;   order pref flags
IN NAPTR 100  10   "s"   "EM:ProtA"                  (   ; service
                         ""                              ; regexp
                        _ProtA._tcp.thinkingcat.example. ; replacement
                                                     )
IN NAPTR 100  20   "s"   "EM:ProtB"                  (   ; service
                         ""                              ; regexp
                         _ProtB._tcp.example.com.        ; replacement
                                                     )
IN NAPTR 100  30   "s"   "EM:ProtC"                  (   ; service
                         ""                              ; regexp
                         _ProtC._tcp.example.com.        ; replacement
                                                     )

   Then the administrators at example.com can manage the preference
   rankings of the servers they use to support the ProtB service:

   _ProtB._tcp.example.com.
    ;;     Pref Weight Port  Target
   IN SRV 10    0      10001 bigiron.example.com.
   IN SRV 20    0      10001 backup.em.example.com.
   IN SRV 30    0      10001 nuclearfallout.australia-isp.example.com
}
```

In this example, three NAPTR records are returned to the client, in response to a query for the FQDN `thinkingcat.example.`: the first one points to a SRV record (flags = ’s’), defines a service named ’EM:ProtA’, which is hosted by a server named `_ProtA._tcp.thinkingcat.example.`, which indicated that ProtA uses TCP for transport protocol. The second and third are similar to the first: they provide the same service ’EM’, but with different protocols (’ProtB’,’ProtC’) and with different preferences (to enable prioritization of servers). The SRVs records contain other names at which the server called `_ProtB._tcp.example.com.` can be found. It can be seen that there are three SRV RRs, pointing to different servers (all to the same TCP port), with different preference levels.

D.3 Usage profile for OConS

To formalize the DNS usage profile:

- FQDNs are used to define OConS entities. Examples of valid FQDNs are:
  - `ocons_inc.AP-meeting-room.nodes.company.com`
  - `ocons_inc.nodes.epc.mnc001.mcc222.3gppnetwork.org` [4]
  - `ocons_inc.nodes.example.com` [5]
- An OConS entity , like an INC, willing to discover another INC on another node, first issues a DNS query, asking for a NAPTR record of the FQDN of the INC.
- DNS server answers with all the NAPTR, SRV, A RRs defined for the INC

[4] In this example, the standard 3GPP network suffix is used
[5] In this case, all the INCs of a specific domain are returned

A very simple example of DNS configuration (without SRV RRs) can be the following:

```
{
ocons_inc.nodes.example.com
;;  order  pref flags service  regexp  replacement
IN NAPTR 100   10    "a" "INC:ocons"  ""  inc.server1.nodes.example.com
IN NAPTR 100   20    "a" "INC:ocons"  ""  inc.server2.nodes.example.com
IN NAPTR 100   10    "a" "INC:ocons"  ""  inc.ap1.nodes.example.com
IN NAPTR 100   10    "a" "INC:ocons"  ""  inc.ap2.nodes.example.com

;; all A records follow
inc.server1.nodes.example.com IN A 10.10.10.1
inc.server2.nodes.example.com IN A 10.10.10.2
inc.ap1.nodes.example.com IN A 10.0.0.1
inc.ap2.nodes.example.com IN A 10.0.0.2
}
```

D.4 Names definition for Mechanism Manifest

As described in Sec. 3.2.2, the Mechanism Manifest contains the characteristics of each mechanims, including its name and the names of the needed entities (one DE and zero/more IEs and EEs). This section provides guidelines for the FQDN names to be used in the Mechanism Manifest.

Name Type	FQDN	Notes
Mechanism Name	`[name].[level].example.com`	• name = e.g. multipath, access-selection, net-coding etc. • level = flow, network, link
Entity Name	`de_ID.[type].[mechanism_name].example.com`	• type= e.g. terminal, access-point, DCU etc.

Some examples follow. The first two example show how a domain-independent naming scheme could work (similarly to the APN name definition in GRPS, which uses the `.gprs` suffix); the latter three show instead how to identify, in a very flexible and powerful way, a single DE for a specific mechanism within a complex network in a single domain.

- `multipath.ocons.`
- `net_coding.ocons.`
- `de234.terminal.access_selection.madrid_pop.telefonica.es`

- `de123.terminal.access_selection.granada_pop.telefonica.es`
- `de456.access-point.access_selection.granada_pop.telefonica.es`

D.5 Conclusions

The S-NAPTR resolution mechanism provides very high flexibility in the deployment and discovery of OConS nodes. Potentially, every OConS entity, not only the INCs, can be assigned a FQDN. Other examples of advanced usage of S-NAPTR can be:

- Use One/multiple protocols/ports for the same entity
- Use one dedicated physical interface for each OConS entity on a node
- Use one physical interface for all OConS entity on a node
- Load-balancing among multiple OConS entities on a node
- Discover an INC in another domain, if a common well-known syntax is used across domains for FQDNs, like the one that is in place in 3GPP PLMNs.

E Threat Model on OConS

E.1 Introduction

The work on OConS and the design and specification that have been produced and presented in this deliverable (see also [85]) are an challenging attempt to improve service delivery based on innovative concepts for the service control. One aspect to consider in the design and specification of OConS is in fact security, namely questions like: is the proposed solution secure and can the envisaged new solution be developed and operated without additional security risks that may in the end negatively affect its operation in a communication infrastructure? One way for starting to encompass security aspects is an initial threat analysis, which later on in the development will turn into security requirements and eventually design decisions that formulate qualified security properties or functions of the systems under consideration.

The contribution of this annex is a threat analysis of OConS and it is considered to contribute to the development and technical improvements of the OConS concept.

E.2 Threat Modelling

Threat modelling is an appropriate instrument to assess the IT security of a system [86], allowing to discuss the security properties of a systems in relation to e.g. defined misuse or resource depletion cases, and can be used as dedicated instrument within development processes. These processes are typically iterative and often start when the system design is not focusing on security yet. However, threat modeling presents eventually analysis results that contribute to a development processes and helps designers early on to encompass or elaborate the adequate and required security properties in the next cycle of design [87]. The quality of such a threat model is determined by completeness and consistency and it is solely based on expert knowledge.

Threat models may take different forms. An *attacker-centric threat model* assumes the role of an attacker, describes her ambitions, motivations and capabilities and analyses what harm can be done to the analysed system, which is the Orchestration in our case. An attacker-centric threat analysis has been performed in [88], in which the following threat categories have been distinguished:

Orchestration misuse which basically attacks the integrity of the coordination of mechanisms that are subject of the Orchestration such that either the attacker will have an advantage or other users will perceive an unexpected behaviour or disadvantage.

Orchestrated mechanism misuse attacks the subjects of the orchestration for the attackers advantage, for example, to save money or receive a priority compared to other users of the system.

Disturbance of Orchestration is an attack that not necessarily results in an advantage for attackers. This attack is addressing resource depletion mainly.

Privacy violation that puts the data protection of end user or operator at risk.

The following subsections give a list of attacker-centric threats for the OConS orchestration. For a complete description of the identified threats the reader is referred to [88].

E.2.0.1 Orchestration misuse

- by spoofing the id of a legitimate interface via the interface registry
- by sinkholing the registry that maintains interface references
- by MitM attacking the content of signaling messages
- by MitM attacking the order of signaling messages
- by manipulating the cache of a node
- by replacing/manipulating (parts of) the implementation on an OConS node
- by manipulating the client application policy/requirements (somewhere also referred to as 'demand profile')
- by manipulating network state input
- by spoofing the OR
- by tampering with OR notifications/registrations
- by manipulating cascaded orchestration

E.2.0.2 Orchestrated mechanism misuse

- by spoofing the value IEs are supplying to the orchestration
- by tampering with IE values by affecting measured variable
- by manipulating a decision taken by a DE
- by manipulating a enforcement carried out by an EE
- by spoofing the id of a legitimate mechanisms
- by manipulating or replacing the implementation or configuration of a legitimate mechanisms
- by high-jacking OConS service subscription

E.2.0.3 Disturbance of Orchestration

- by knowing how to address an interface, messages can be send such that resources are depleted
- by manipulating mechanism definition
- by preventing OR to update information
- by disturbing resource/mechanism allocation
- by intercepting/retaining/surpressing messages that are necessary for the SOP
- by tampering with time source

E.2.0.4 Privacy violation

- eavesdropping communication
- creating usage profiles

E.3 Protocol-Centric Threat Model

Another approach in modeling theats is the *protocol-centric* threat modelling. In this case protocols from the underlying protocol stack are looked at, investigating to what extend their known vulnerabilities affect the security of the entire systems being analysed.

Part of the specification of OConS is that OConS Functional Entities that are distributed over nodes or reside on the same node make use of INC to interact with each other. Considering this part of the OConS design, a complementing protocol-centric threat analysis seems to be useful, because INC will be an concrete protocol implementation placed on the Internet layer protocols.

E.3.0.5 OConS Naming and addressing

OConS domains, nodes, services, mechanisms, and functional entities use unique IDs to identify. If an attacker can spoof this identity, i.e. if he can make another party believe that he is the rightful owner of this ID authenticity is not guaranteed. This attack can lead to further attacks including illegitimate controlling the orchestration decision e.g. by injecting wrong information/decisions.

Attackers may perform masquerading attacks to feign a false identity, e.g. for the purpose of a Man-in-the-Middle (MitM) attack. Masquerading attacks primarily threaten the authenticity and may later, if successful, induce various kinds of attacks threatening other security goals such as confidentiality, data integrity and reliability.

If the adress space uses a special schema after which it is built up an attacker might use this information to gather knowledge about the topology or the type of devices (e.g. an end-user device) for instance. In 3.5.1 it is described that the OConS ID uniquely identifies the OConS node on which a certain functional entity is deployed. This information gives an attacker the change to identify all functional entities running on a node. He could try to attack the weakest entity in order to maliciously influence the other entities that are running on the same node.

E.3.0.6 INC

Integrity of INC messages are crucial for the correct execution of orchestration processes. Tampering with any kind of INC message can negatively affect these processes resulting e.g. in denial-of-service or the use of hostile mechanisms in orchestration. One special form of packet manipulation is the change of the order of packets. This may cause problems if fragmented data is sent using multiple packets or if decisions are taken on a first-come, first-serve basis.

An attacker might learn the networks topology by eavesdropping broadcasted INC announcements. Depending on the information in the broadcasted messages he could furthermore collect valuable information about possible vulnerabilities.

E.3.0.7 DNS bootstrapping

Several distinct classes of threats to the DNS are described in [89]:

- Packet Interception
- ID Guessing and Query Prediction
- Name Chaining
- Betrayal By Trusted Server
- Denial of Service
- Authenticated Denial of Domain Names
- Wildcards

Most of which are DNS-related instances of more general problems, but a few of which are specific to peculiarities of the DNS protocol. It concludes that the Domain Name System Security Extensions (DNSSEC) extensions do appear to solve a set of problems that do need to be solved, and are worth deploying. The authors of [89] believe that deploying DNSSEC [80, 90, 91] will help to address some, but not all, of the known threats to the DNS.

E.3.0.8 Software Integrity

Software in OConS Node could be altered intentionally or unintentionally. The threat that an attacker exchanges parts of the code with his own implementation can lead to unforeseen behaviour of OConS Orchestration. This behaviour can include information leakage to other malicious nodes, performance drawbacks, loss of control over the orchestration process, or denial-of-service.

E.4 Discussion

The results of the threat analysis presented here and in [88] describe potentially foreseeable attacks on OConS. The threat model addressed here concentrates on the OConS orchestration as detailed in Section 3.7 and it is based on the OConS architecture design that is presented in Section 3.3. Aspects concerning the management of OConS have not been discussed and will require to enlarge the scope of the threat model. This threat model improves the awareness of the designer which aspects of the resulting solution can be misused and how this may happen. As such it prepares the ground for the next cycle of the development process when OConS is brought forward.

In future OConS designs could improve by using e.g. solutions that handle authorized use of OConS Functional Entities in a way that helps to differentiate legitimate from non-legitimate users. It can be assumed that this will prevent most of the misuse of orchestrations and mechanisms. As part of such a secured naming and addressing solution an interaction across different administrative domains should be considered. Such interaction between administrative domains must support authorization of interactions for authenticated users, which might have to register to become authorized. Interaction need to be protected in a way that the integrity of the interaction is ensured and thus misuse by manipulating e.g. messages or message contents is detectable or avoided. Since the interaction between administrative domain has the potential to establish over some period in time some trust between these domains, namely if the experienced behaviour is as specified before, some logging of interaction will probably also be needed. In many comparable scenarios IPsec as proven to be a suitable support for reliable security associations [92]. However, it needs to be validated if this protocol is an efficient and effective solution, which supports the distribution of OConS Functional Entities e.g. also into the end user terminal or into other administrative domains.

List of Abbreviations, Acronyms, and Definitions

3G	Third Generation
ACK	Acknowledgment
AP	Access Point
API	Application Programming Interface
ARP	Address Resolution Protocol
AS	Autonomous Systems
BGP	Border Gateway Protocol
BIP	Binary integer programming
BPQ	Bundle Protocol Query
BS	Base Station
CE	Customer Edge
CL	Convergence Layer
CloNe	Cloud Networking
CN	Core Network
CPU	Central Processing Unit
CQI	Channel Quality Information
DC	Data Centre
DCC	Domain Control Client
DCM	Distributed Cloud Manager
DCP	Distributed Cloud Protocol
DCP-NL	DCP Network Level
DCP-LNP	DCP Link Negotiation Protocol
DCU	Domain Control Unit
DDC-ARM	Distributed Data Center Address Resolution Mechanism
DDC-WIM	Distributed Data Center WAN Interconnectivity Mechanism
DE	Decision Making Entity

DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
DMM	Distributed Mobility Management
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DTN	Delay Tolerant Networking
EE	Execution and Enforcement Entity
eNB	evolved Node B
ER	Epidemic Routing
EPC	Evolved Packet Core
EwLC	Event with Large Crowd
FM	Flow Mechanisms
FNS	Flash Network Slice
FQDN	Fully-Qualified Domain Name
GMPLS	Generalized Multi-Protocol Label Switching
GUI	Graphical User Interface
HA	Home Agent
HMAC	Hash-based Message Authentication Code
HURRy	HUman Routines optimise Routing
ICN	Information Centric Networking
IE	Information Management Entity
IETF	Internet Engineering Task Force
INC	Intra-/Inter- Node Communication
IPC	Inter-Process Communication
ITU	International Telecommunication Union
JSON	JavaScript Object Notation
LAN	Local Area Network
LC	Linear statistics Combination
LM	Link Mechanisms
LMA	Local Mobility Anchor

LSP	Label Switched Path
LTE	Long Term Evolution
MAP	Mesh Access Point
MAG	Mobile Access Gateway
MCS	Modulation and Coding Scheme
MitM	Man-in-the-Middle
MFM	Multihomed Flow Management
MP	Multi-Path
MPO	Mobility Parameters Optimization
MPLS	Multi-Protocol Label Switching
MPLS-TE	Multi-Protocol Label Switching - Traffic Engineering
MPO	Mobility parameters optimization
MS	Mobile Station
MT	Mobile Terminal
NC	Network Coding
NDO	Named Data Object
NEP	Nash Equilibrium Point
NI	NetInf Identifier
NetInf	Network of Information
NM	Network Mechanisms
NNRP	NEC NetInf Router Platform
NRS	Name Resolution Service
NTP	Network Time Protocol
OAM	Operations And Maintenance
OCCI	Open Cloud Computing Interface
OCNI	Open Cloud Network Interface
OConS	Open Connectivity Services
OF	OpenFlow
OFDMA	Orthogonal Frequency-Division Multiple Access
OMPNetInf	OConS Multi-path Network of Information

OR	Orchestration Registry
OSAP	Orchestration Service Access Point
OSI	Open Systems Interconnection
OSPF-TE	Open Shortest Path First - Traffic Engineering
PE	Provider Edge
PGW	Packet Data Network Gateway
PKI	Public Key Infrastructure
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random Access Memory
RAT	Radio Access Technology
RLNC	Random Linear Network Coding
RR	Resource Record
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
RTT	Round-trip time
SAIL	Scalable and Adaptive Internet Solutions
SDN	Software Defined Networking
SGW	Serving Gateway
SNR	Signal-to-Noise Ratio
SOP	Service Orchestration Process
TCP	Transmission Control Protocol
TED	Traffic Engineering Database
TLV	Type-Length-Value
UDP	User Datagram Protocol
UE	User Equipment
VLAN	Virtual Local Area Network
VM	Virtual Machine
VNet	Virtual Network
VPLS	Virtual Private LAN Services

VRRA	VNet Radio Resource Allocation
WAN	Wide Area Network
WMN	Wireless Mesh Network
XML	eXtensible Markup Language

List of Figures

List of Tables

References

[1] SAIL. Architectural Concepts of Connectivity Services. Deliverable FP7-ICT-2009-5-257448-SAIL/D.C.1, SAIL project, July 2011. available online from http://www.sail-project.eu.

[2] SAIL. Architectural Concepts of Connectivity Services - Addendum. Deliverable FP7-ICT-2009-5-257448-SAIL/D.C.1 Addendum, SAIL project, January 2012. Available online from http://www.sail-project.eu.

[3] SAIL. Applications for Connectivity Services and Evaluation. Deliverable FP7-ICT-2009-5-257448-SAIL/D.C.4, SAIL project, February 2013. available online from http://www.sail-project.eu.

[4] SAIL. Demonstrator Specification and Integration Plan. Deliverable FP7-ICT-2009-5-257448-SAIL/D.C.3, SAIL project, May 2012. available online from http://www.sail-project.eu.

[5] SAIL. Demonstrator for Connectivity Services. Deliverable FP7-ICT-2009-5-257448-SAIL/D.C.5, SAIL project, February 2013. available online from http://www.sail-project.eu.

[6] ONF Architecture WG and Design Team. ONF Architecture Document - v0.0.1, December 2012. `http://www.opennetworking.org/`.

[7] Architecture and Framework Design Team. ONF Framework Document - v0.4.2, December 2012. `http://www.opennetworking.org/`.

[8] ISO/IEC/IEEE 42010:2011 Systems and software engineering – Architecture description. JTC 1/SC 7 Standard, Nov 2011.

[9] Ed. C. Perkins, D. Johnson, and J. Arkko. Mobility Support in IPv6. RFC 6275(Standard), 2011.

[10] 3GPP. Policy and charging control architecture. TS 23.203, 3rd Generation Partnership Project (3GPP).

[11] IEEE. IEEE802.21 Standard for Local and Metropolitan Area Networks: Media Independent Handover Services. Nov. 2008.

[12] Tremblay, Benoit. D-2.3 (d.a.3) final harmonised sail architecture. Technical report, FP7-ICT-2009-5-257448-SAIL, 2013.

[13] Pierrick Louin and Philippe Bertin. Network and host based distributed mobility. In *WPMC2011: Wireless Personal Multimedia Communications*, Brest, France, October 2011.

[14] 3GPP. General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access. TS 23.401, 3rd Generation Partnership Project (3GPP).

[15] 3GPP. Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3. TS 24.301, 3rd Generation Partnership Project (3GPP).

[16] David Gómez, Sofiane Hassayoun, Arnaldo Herrero, Ramón Agüero, and David Ros. Impact of network coding on TCP performance in wireless mesh networks. In *23th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE Proceedings*, September 2012.

[17] David Gómez, Sofiane Hassayoun, Arnaldo Herrero, Ramón Agüero, David Ros, and Marta García-Arranz. On the addition of a network coding layer within an open connectivity services framework. In *4th International Conference on Mobile Networks and Management (MONAMI), 2012*, September 2012.

[18] Reuven Cohen and Guy Grebla. Efficient allocation of CQI channels in broadband wireless networks. In *Infocom'2011 (mini-conference), Shanghai, China*, April 2011.

[19] Edwall, Thomas. D-2.9 (D.A.9) Description of Overall Prototyping Use Cases, Scenarios and Integration Points. Technical report, FP7-ICT-2009-5-257448-SAIL, 2012.

[20] SAIL. D-5.3 (D-D.2) Description of the implemented prototype. Technical report, FP7-ICT-2009-5-257448-SAIL, August 2012.

[21] SAIL. NetInf Content Delivery and Operations. Deliverable FP7-ICT-2009-5-257448/D-3.2, SAIL project, May 2012. available online from http://www.sail-project.eu.

[22] Luis Diez, Olivier Mehani, Lucian Suciu, and Ramón Agüero. Design and implementation of the open connectivity services framework. In Lucio S. Ferreira and Lucian Suciu, editors, *MONAMI 2012, 4th International Conference on Mobile Networks and Management, OConS workshop*, Lecture notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Brussels, Belgium, September 2012. TUHH, EAI, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[23] ETSI. Industry Specification Group on Network Functions Virtualisation (NfV). First meeting to be held in January 2013.

[24] Video lan client (vlc), (last accessed August 2012). `http://www.videolan.org`.

[25] D-H Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. In *In Proceedings of the Computer Networks and ISDN Systems*, August 1989.

[26] Alan Ford, Costin Raiciu, Mark Handley, and Olivier Bonaventure. TCP extensions for multipath operation with multiple addresses. Internet-Draft draft-ietf-mptcp-multiaddressed-09.txt, IETF Secretariat, Fremont, CA, USA, May 2012.

[27] Janardhan R. Iyengar, Paul D. Amer, and Randall Stewart. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. *IEEE/ACM Transactions on Networking*, 14(5):951–964, October 2006.

[28] Randall R. Stewart. Stream control transmission protocol. RFC 4960, RFC Editor, Fremont, CA, USA, September 2007.

[29] Hakim Adhari, Thomas Dreibholz, Martin Becke, Erwin P. Rathgeb, and Michael Tüxen. Evaluation of concurrent multipath transfer over dissimilar paths. In *Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, WAINA '11, pages 708–714, Washington, DC, USA, 2011. IEEE Computer Society.

[30] Golam Sarwar, Roksana Boreli, Emmanuel Lochin, and Ahlem Mifdaour. Performance evaluation of multipath transport protocol in asymmetric heterogeneous network environment. In Yingjie J. Guo, Anthony Parker, Sanjay Jha, Minoru Okada, Xiaojing Huang, Yeong M. Jang, Jinhong Yuan, Supavadee Aramvith, Gang-Ding Peng, Honggang Zhang, and Dhammika Jayalath, editors, *ISCIT 2012, International Symposium on Communications and Information Technologies*, October 2012. To appear.

[31] VINT Project. *The ns Manual (formerly ns Notes and Documentation)*, January 2009.

[32] Johnny Choque, Ramón Agüero, and Luis Muñoz. Optimum selection of access networks within heterogeneous wireless environments based on linear programming techniques. *MONET*, 16(4):412–423, 2011.

[33] Carmen López, Ramón Agüero, Johnny Choque, and Luis Muñoz. On the equilibrium of resource allocation for heterogeneous wireless access networks. In *PIMRC*, 2012.

[34] Johnny Choque, Ramón Agüero, and Luis Muñoz. Simulation framework for the evaluation of access selection algorithms over heterogeneous wireless networks. In *MONAMI*, pages 46–60, 2011.

[35] Olivier Mehani, Roksana Boreli, Michael Maher, and Thierry Ernst. User- and application-centric multihomed flow management. In Tom Pfeifer and Anura Jayasumana, editors, *LCN 2011, 36th IEEE Conference on Local Computer Networks*, pages 26–34, Los Alamitos, CA, USA, October 2011. IEEE Computer Society, IEEE Computer Society.

[36] New definitions for inclusion in recommendation P.10/G.100. Recommendation P.10/G.100 Amendment 2, ITU-T SG12, Geneva, Switzerland, July 2008.

[37] The E-model, a computational model for use in transmission planning. Recommendation G.107, ITU-T SG12, Geneva, Switzerland, March 2005.

[38] Estimating end-to-end performance in IP networks for data applications. Recommendation G.1030, ITU-T SG12, Geneva, Switzerland, May 2006.

[39] Opinion model for video-telephony applications. Recommendation G.1070, ITU-T SG12, Geneva, Switzerland, April 2007.

[40] Xi Li, Olivier Mehani, Ramón Agüero, Roksana Boreli, Yasir Zaki, and Umar Toseef. Evaluating user-centric multihomed flow management for mobile devices in simulated heterogeneous networks. In Ramón Agüero and Susana Sargento, editors, *MONAMI 2012, 4th International Conference on Mobile Networks and Management*, Lecture notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Brussels, Belgium, September 2012. TUHH, ICST, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[41] Umar Toseef, Yasir Zaki, Andreas Timm-Giel, and Carmelita Görg. Uplink QoS aware multi-homing in integrated 3GPP and non-3GPP future networks. In *4th International Conference on Mobile Networks and Management Monami 2012*, Hamburg, Germany (Accepted), September 24-26 2012.

[42] Umar Toseef, Yasir Zaki, Liang Zhao, Andreas Timm-Giel, and Carmelita Görg. Qos aware multi-homing in integrated 3gpp and non-3gpp future networks. In *the Seventh International Conference on Systems and Networks Communications, ICSNC 2012*, Lisbon, Portugal (Submitted), November 18-23 2012.

[43] Reuven Cohen and Anna Levin. Handovers with forward admission control for adaptive TCP streaming in LTE-Advanced with small cells. In *IEEE ICCCN 2012, Munich, Germany*, 2012.

[44] Fariborz Derakhshan, Heidrun Grob-Lipski, Horst Roessler, Peter Schefczik, and Michael Soellner. On converged multidomain management of connectivity in heterogeneous networks. *Future Network and Mobile Summit (FuNeMS2012), Berlin.*, July 2012.

[45] D. Klein, R. Pries, M. Scharf, M. Soellner, and M. Menth. Modeling and evaluation of address resolution scalability in vpls. In *Communications (ICC), 2012 IEEE International Conference on*, pages 2741 –2746, june 2012.

[46] L. Dunbar and S. Hares. Scalable Address Resolution for Large Data Center Problem Statements. Internet-Draft draft-draft-dunbar-arp-for-large-dc-problem-statement, Internet Engineering Task Force, July 2010. Work in progress.

[47] Y. Li. Problem statement on address resolution in virtual machine migration. Internet-Draft draft-liyz-armd-vm-migration-ps, Internet Engineering Task Force, October 2010. Work in progress.

[48] IETF Working Group. Address Resolution for Massive numbers of hosts in Data center (Active WG). `http://tools.ietf.org/wg/armd/`, October 2011.

[49] P. Knight and C. Lewis. Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts. *Communications Magazine, IEEE*, 42(6):124 – 131, june 2004.

[50] M. Lasserre and V. Kompella. Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling. Technical Report rfc4762, Internet Engineering Task Force, January 2007.

[51] K. Elmeleegy and A.L. Cox. Etherproxy: Scaling ethernet by suppressing broadcast traffic. In *INFOCOM 2009, IEEE*, pages 1584 –1592, april 2009.

[52] Rami Cohen and Danny Raz. Cost effective resource allocation of overlay routing relay nodes. In *Infocom'2011, Shanghai, China*, April 2011.

[53] Pierrick Seite and Philippe Bertin. Distributed Mobility Anchoring. Internet Draft draft-seite-dmm-dma-05.txt, Work in progress, July 2012.

[54] Ed. S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy Mobile IPv6. RFC 5213(Standard), 2008.

[55] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia. Fast Handovers for Proxy Mobile IPv6. RFC 5949(Standard), 2010.

[56] David Gómez, Sofiane Hassayoun, Arnaldo Herrero, Ramón Agüero, and David Ros. Impact of network coding on tcp performance in wireless mesh networks. In *PIMRC*, 2012.

[57] A. Serrador and L.M. Correia. Policies for a cost function for heterogeneous networks performance evaluation. In *18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007.

[58] IEEE. 802.1Q-2005: Virtual Bridged Local Area Networks, 2005.

[59] L.S. Ferreira and L.M. Correia. Energy-efficient radio resource management in self-organised multi-radio wireless mesh networks. In *IEEE PIMRC 2011: 22nd IEEE Symposium on Personal, Indoor, Mobile and Radio Communications*, Toronto, Canada, September 2011.

[60] R. Aguero, L. Caeiro, L.M. Correia, L.S. Ferreira, M. Garcïïœœa-Arranz, L. Suciu, and A. Timm-Giel. Ocons: Towards open connectivity services in the future internet. In *MONAMI: 3rd International ICST Conference on Mobile Networks and Management*, Aveiro, Portugal, September 2011.

[61] L.S. Ferreira and L.M. Correia. Radio resource management for optimising multi-radio wireless mesh networks deployments. In *WPMC 2011: 14th International Symposium on Wireless Personal and Mobile Communications*, Brest, France, October 2011.

[62] L.S. Ferreira and L.M. Correia. Efficient and fair radio resources allocation for spontaneous multi-radio wireless mesh networks. In *ISSSE 2012: International Symposium on Signals, Systems and Electronics*, October 2012.

[63] L.S. Ferreira and L.M. Correia. Efficient and fair radio resources allocation for spontaneous multi-radio wireless mesh networks. In *ISSSE 2012: International Symposium on Signals, Systems and Electronics*, Potsdam, Germany, October 2012.

[64] G. Shafer. A mathematical theory of evidence, 1976.

[65] Z. Quan, S. Cui, A. Sayed, and H. Poor. Optimal multiband joint detection for spectrum sensing in cognitive radio networks. IEEE Transactions on Signal Processing, 2009.

[66] Software-defined networking, April 2009. In INFOCOM, (keynote talk).

[67] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. In *SIGCOMM Comput. Commun. Rev., 38*, March 2008.

[68] G. Parulkar S. Das, N. McKeown, D. Getachew P. Singh, and L. Ong. Packet and circuit network convergence with openflow. In *In Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference*, 2010.

[69] Reuven Cohen Gabi Nakibly and Liran Katzir. On the trade-off between control plane load and data plane efficiency for mpls multi-path flows. Technical report, Technion, Israel Institue for Technology, April 2012. available online at http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2012/CS/CS-2012-04.pdf.

[70] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In *In Proceedings of MASCOTS*, 2001.

[71] N. Spring R. Mahajan and D. Wetherall. Measuring isp topologies with rocketfuel. In *In Proceedings of the ACM SIGCOMM*, August 2002.

[72] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972.

[73] Carmen López, Ramón Agüero, Johnny Choque, and Luis Muñoz. On the equilibrium of resource allocation for heterogeneous wireless access networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 1049 –1054, sept. 2012.

[74] Carmen López, Johnny Choque, Ramón Agüero, and Luis Muñoz. On the equilibrium of pricing assignment for heterogeneous wireless access networks. In *MONAMI 2012, 4th International Conference on Mobile Networks and Management*, 2012.

[75] Ralf Nyrén, Andy Edmonds, Alexander Papaspyrou, and Thijs Metsch. Open cloud computing interface - core. GFD-P-R. 183, Open Grid Forum, April 2011.

[76] Thijs Metsch and Andy Edmonds. Open cloud computing interface - infrastructure. GFD-P-R. 184, Open Grid Forum, April 2011.

[77] SAIL. D-5.2 (D-D.1) Cloud Network Architecture Description, Rev 2.0. Technical report, FP7-ICT-2009-5-257448-SAIL, January 2012.

[78] Michael C. Richardson. A method for storing IPsec keying material in DNS. RFC 4025, RFC Editor, Fremont, CA, USA, March 2005.

[79] David Mills, Jim Martin, Jack Burbank, and William Kasch. Network time protocol version 4: Protocol and algorithms specification. RFC 5905, RFC Editor, Fremont, CA, USA, June 2010.

[80] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose. DNS security introduction and requirements. RFC 4033, RFC Editor, Fremont, CA, USA, March 2005.

[81] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter base protocol. RFC 3588 (Standard), September 2003.

[82] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Standard), November 1987.

[83] P.V. Mockapetris. Domain names - - implementation and specification. RFC 1035 (Standard), November 1987.

[84] L. Daigle and A. Newton. Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS). RFC 3958, January 2005.

[85] Lucio S. Ferreira, Ramón Agüero, Luisa Caeiro, Avi Miron, Michael Soellner, Peter Schoo, Lucian Suciu, Andreas Timm-Giel, and Asanga Udugama. Open connectivity services for the future internet. In *Proceedings of MON-AMI 2012*, 2012.

[86] Claudia Eckert. *IT-Sicherheit: Konzepte - Verfahren - Protokolle.* Oldenbourg, 6., revised and extended issue. edition, 2009.

[87] Charles B. Haley, Jonathan D. Moffett, Robin Laney, and Bashar Nuseibeh. A Framework for Security Requirements Engineering. In *SESS '06: Proceedings of the 2006 international workshop on Software engineering for secure systems*, pages 35–42, New York, NY, USA, 2006. ACM Press.

[88] P. Schoo and R. Marx. Threat model based security evaluation of open connectivity services. In *MONAMI: 4rd International ICST Conference on Mobile Networks and Management*, Germany, July 2012.

[89] D. Atkins and R. Austein. Threat Analysis of the Domain Name System (DNS). RFC 3833 (Informational), August 2004.

[90] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014.

[91] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014.

[92] 3G Security; Network Domain Security (NDS) – IP Network Layer Security.