Objective FP7-ICT-2009-5-257448/D-4.4

Future Networks

Project 257448

"SAIL – Scalable and Adaptable Internet Solutions"

D-4.4
(D.C.4) Applications for Connectivity Services and Evaluation

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

Revision: **1.0**
Duration: **2013-02-28**

Document Properties

Document Number:	D-4.4
Document Title:	**(D.C.4) Applications for Connectivity Services and Evaluation**
Document Responsible:	Lucian Suciu (FT-Orange)
Document Editor:	Avi Miron (Technion), 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), 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:	Initial Version
Version:	1.0

Production Properties:

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

Document History:

Revision	Date	Issued by	Description
1.0	2013-02-28	Avi Miron, Lucian Suciu	Initial Version

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.4 of the EU-FP7 project SAIL (Scalable and Adaptive Internet Solutions, [1]) presents in detail the results of the research work carried out in the WP-C, it discusses the applications for the OConS (Open Connectivity Services), and it also evaluates and assesses these results.

Keywords:

open connectivity services, prototyping, demonstration, cloud networking, network of information, openflow, dtn, icn, multi-path, distributed connectivity control, distributed mobility management

Contents

1 Introduction

This D.C.4 document is one of the final SAIL Open Connectivity Services (OConS) deliverables coming from WPC, reporting Applications for Connectivity Services and Evaluation. Together with the new D.C.2 (Architecture and Mechanisms for Connectivity Services [2]), and D.C.5 (Demonstrator for Connectivity Services [3]) that are all delivered at the end of the project, these three deliverables conclude the WPC research and development activities.

OConS is positioned at the forefront of the research and development effort, in the area of separation of control and forwarding planes. It goes beyond Path Computation Element (PCE), Software Defined Networking (SDN) and OpenFlow, providing a wider scope. It handles a wide variety of services that goes beyond packet forwarding only, thus has the opportunity to provide holistic solutions to connectivity problems. It can run on any network node, including end user devices. The OConS architectural framework provides abstraction and modularity, and includes a rich set of connectivity mechanisms and means to combine them into powerful connectivity services. It considers user profile and application demands for its operation, aiming at meeting the expected demands. Most importantly, OConS offers an environment where cross-layer functionality is handled in a structured manner. As such, it shows a great potential.

This deliverable is built on top of D.C.2 [2], leveraging the fact that both are finalized together. The OConS architectural framework is fully described in D.C.2, and is only assessed here. In addition, the rich set of OConS mechanisms being presented in D.C.2, we discuss here the evaluation of their final results and their relationships with OConS. We therefore assume that the reader of this document is familiar with the content of D.C.2. The main focus of this document is the applications of OConS, and the assessment of the results achieved.

Accordingly, D.C.4 is providing a comprehensive assessment of the research results on OConS. Looking from a *bird eye perspective* it includes an evaluation of:

- The architectural framework, encompassing the models that constraint the design of possible OConS architectures;
- The OConS architecture that describes functional system architectures according to the OConS framework, and encompasses orchestrated mechanisms;
- The OConS mechanisms developed that can be grouped by the OConS orchestration to form an OConS service;
- The applications of OConS to specific application domains (scenarios); we have focused on application domains from the SAIL project, providing enhanced connectivity services for Cloud Networking (CloNe), Network of Information (NetInf) and the flash crowd scenario.

Where applicable these results are backed up with prototyping activities.

Taking *a bottom up approach*, we have developed and evaluated advanced connectivity solutions on different levels: link layer, network layer and flow layer. These are presented as OConS mechanisms (Chapter 2). Despite their great variety and scope, all the OConS mechanisms proposed enhance the connectivity, each for a specific network technology, while involving one or more layers. Thanks to the orchestration, the mechanisms can be combined into custom OConS services in a systematic manner, harnessing the full connectivity potential. Such combined mechanisms are demonstrating significant improvements for a few application areas (Chapter 3), notably for CloNe and NetInf. Structured combination of various mechanisms was not possible prior to OConS. The specific combination of mechanisms can be tailored, depending on the context, the network technologies, the network states, the availability of the mechanisms, and user or application demands, thereby, demonstrating that the combination is better than the sum of its elements. In other words, mechanisms benefit from each other and perform better and more efficient than individually.

The ideas we are following depend on our approach how to perceive and structure the control space (or plane). Actually, we have elaborated on the use of an open interface to the forwarding plane (i.e., the enforcement components), and on how to make it most usable via an Orchestration Service Access Point (OSAP) interface and through the orchestration functionality. These are then exemplified with some applications in Chapter 3. Nonetheless, common principles to all the orchestrated mechanisms are part of our OConS approach, aiming at a specific OConS functional architecture. Thus, the OConS functional architecture describes for a given set of mechanisms how to orchestrate and used them. The principles that each architecture is following are the constituents of the OConS architectural framework. The OConS framework and architecture generalizations, i.e., what holds for all OConS architectures, are then evaluated in Chapter 4.

Indeed, we reckon that one of the main contribution of the OConS approach is to apply software engineering practices for the orchestration processes in the networking control space. By doing so, we have validated the open interfaces to the enforcement plane (forwarding, tunnelling, etc.), also verifying that the prospects for solutions concerning the demanding user-side are staying in a north-south SDN-like perspective. Additionally, the orchestration addresses the potential complexity that may exist in the control space, when service control is distributed or made out of combinations of (existing) mechanisms.

All in all, the OConS orchestration is looking into enriching service control that can handle longer term strategies for orchestrated mechanisms, including new mechanisms developed as delta to already existing mechanisms. This is an approach that facilitates flexibility in a structured way, addressing the complexity introduced by the separation of control and forwarding plane.

The document is organised as follow: Section 2 presents the results and evaluation of each OConS mechanism, and its relationships to OConS. We then move to section 3, which presents the application of OConS for CloNe, NetInf and the flash crowd scenario. Section 4 assesses OConS results, looking at the architectural framework, the mechanisms and the prototyping activities, while trying to take an objective and honest point of view. Finally, Section 5 concludes this deliverable.

2 OConS Mechanisms Results and Evaluation

2.1 Overview

This section provides final results and evaluation of the OConS mechanisms. The OConS mechanisms are discussed in D.C.2 deliverable [2], presenting for each of them, the problem addressed, the approach, and in some cases, initial results and their application to OConS. This section provides for every mechanism a summary of the most significant results, and relate them to OConS.

The structure of this section resembles that of D.C.2 [2], primarily for ease of reading. The mechanisms are grouped into three levels, flow, network and link levels, according to the layer that is affected by them. We also include final results and evaluation for the benchmarking algorithms, which are not part of the OConS real-time execution, but are providing meta-data for combining and configuring the appropriate OConS mechanisms in the most efficient way, while addressing the user and the application's needs.

This section might appear as an aggregation of unrelated algorithms. However, each mechanism is enhancing connectivity for a specific network domain, at one or more layers. The power of those mechanisms should be evaluated only in the context of the applications for OConS, discussed in section 3, when mechanisms are orchestrated to form a holistic OConS service to connectivity problems. This section is presented first, in order to convey the basic knowledge of the mechanisms that are used in the application section.

2.2 OConS Flow Level Mechanisms

2.2.1 Access Selection and decision algorithms

The Access Selection algorithm successfully integrated different parameters within the decision process, namely link quality (as perceived by the end-user device) and load (provided by the access elements). The availability of this information is dynamically delivered to the corresponding Decision Making Entity (DE) during the bootstrapping phase of the relevant entities. It is worth highlighting that this mechanism fits quite well with the orchestration paradigm which the OConS framework fosters. In this sense, the aforementioned prototype successfully integrated it with a *Dynamic Mobility Management* mechanism (see Section 2.3.4). The extensibility of this mechanism facilitates its integration with multi-path mechanisms (more than one access might be selected) or with technology specific mechanisms (for instance the one reported in 2.4.1).

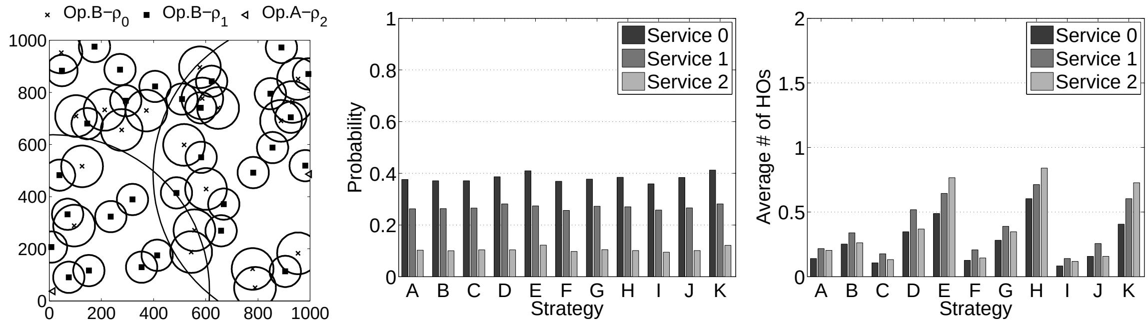
This mechanism has been evaluated using two complementary approaches: first, it has been integrated within a prototype (see [3] and [4]); and the second one corresponds with simulation approach so as to assess the mechanism within more complex scenarios, extending the number of networks technologies and users simulated. In order to perform a more thorough study, we designed and implemented the multi-Constraint Access Selection in heterogeneous Environments (mCASE) simulator. It is an event-driven simulator which allows the fast creation of different network scenarios as well as enabling the use of different types of services. Its main advantage is that it is able to enforce many independent runs of large networks in a rather short execution time. The reader might refer to [5] for a more extensive discussion of the simulator design and implementation.

As will be seen later, the mCASE provides the required flexibility so as to conduct different studies. In order to assess its proper behavior, we carried out a number of illustrative analysis, in which we changed the access selection strategy by linearly combining different parameters of merit (preferred operator, handovers, link quality and load). By changing the weights given to these four parameters we can see the effect of using different selection strategies. In this sense, it mimics an OConS mechanism which uses several Information Management Entities (IEs), as long as they become available and establish the goodness of the available access alternatives based on a combination of such metrics.

In [5] we used a highly heterogeneous network scenario, with several access elements (as can be seen on Figure 2.1a; we studied 11 different selection strategies: the first one (**A**) provided the same weight to the four parameters, a second group (**B-E**) just favored one of the parameters, while the rest of them (**F-K**) combined groups of two of them (more detailed description of the strategies can be found in [5]). In addition, as another proof of the flexibility of the developed framework, we also included three different types of services. Both services 0 and 1 required the same capacity, but 1 calls were three times longer. Service 2 required more capacity.

We can see (Figure 2.1b) that the different strategies do not have a great influence on the probability that a service successfully finishes, but it is clear that services with fewer requirements (in terms of duration/capacity) show a greater probability of being successful (service 0)[1].

On the other hand, Figure 2.1c yields a great influence of the strategies over the number of handovers. In this case, **C**, **F** and **I** show a lower average number of handovers per service, since they prioritize their minimization in the corresponding utility function. On the contrary, for **J**, the impact of the load balancing weight causes a slight increase on the number of handovers.



(a) Network Scenario (b) Successful services (c) Handovers

Figure 2.1: Network deployment used during the analysis and performance of different access selection strategies

2.2.2 Efficient Handover for Adaptive TCP Streaming in LTE-Advanced with Small Cells

This OConS mechanism provides improved and seamless mobility, ensuring effective handover procedures and maintaining transport-layer sessions during the movement of the mobile user from one evolved Node B (eNB) to another, in Long Term Evolution (LTE) -advanced networks. The eNB1 decides whether to forward or to silently drop the packets received for the mobile user after the handover decision has been made, depending on the type of the traffic, the availability of resources and the state of the Transmission Control Protocol (TCP) connection, in order to minimize the

[1] These results are meant to assess the validity of the mCASE simulator, and therefore we intentionally increased the traffic load, which lead to low success rates

disruption to the TCP video streaming service. Further information is available in D.C.2 [2], Annex A.6.

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 Orchestration Registry (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. 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 Dynamic Distributed Mobility Management mechanism that is described in section 2.3.4; When operated over an LTE or LTE advanced network, both mechanisms address session continuity, and thus, might need to be coordinated.

2.2.3 Multi-P Decision and Transmission for the Interconnection of 3GPP and non-3GPP Systems

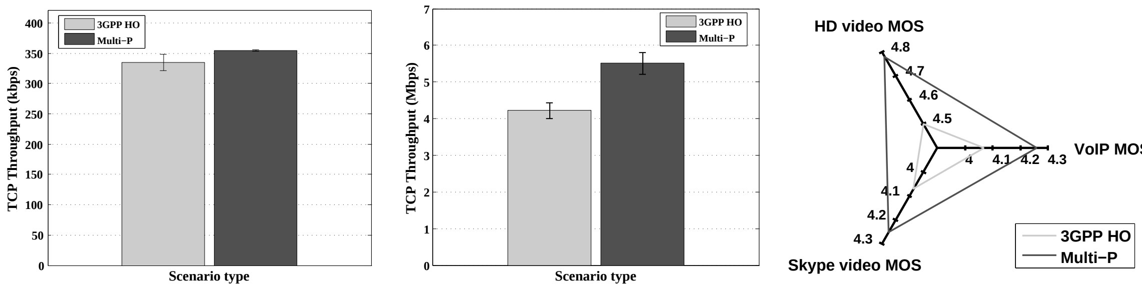
This OConS mechanism deals with the resource allocation problem in integrated 3$^{\text{rd}}$ Generation Partnership Project (3GPP) and non-3GPP networks. The idea behind the integrated scenario is that within today's network, the end users devices are capable of connecting to multiple access networks, like LTE, High-Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX) and Wireless LAN (WLAN). Based on the OConS framework, which collects network and user information for making educated decisions, this mechanism explores how the bandwidth resource allocation can be optimized from the operator points of view for multi-homed devices in both downlink and uplink directions. In [6], the Multi-P algorithm was explained in a scenario with 3GPP LTE and WLAN coverage; it was shown that the OConS Multi-P mechanism is flexible, open and can benefit the end users and the mobile operators (i.e., higher data rates, better resource utilization and more reliable connectivity). In [7] the Multi-P algorithm has been extended to support Quality of Service (QoS)-aware multi-homing in the downlink communication; the results showed that network operators can get significant performance boost for their networks when using the Multi-P algorithm. Similarly, in [8] the QoS-aware Multi-P algorithm have been developed for the uplink communication. The mechanism has been depicted in D.C.2 [2], Annex A.5.

In this section, an analytical solution is developed for the optimized resource allocation in integrated heterogeneous networks. The solution provides the theoretical upper bound for the achievable performance gain for Multi-homed users.

In a wireless access network, frequency spectrum and its usage time are the main network resources which are shared by its users. Considering the common practice where the access networks are assigned with a fixed amount of frequency spectrum, each access network has a given number of network resources. For example, an LTE access network may be installed with 5MHz spectrum bandwidth, or an IEEE 802.11a network typically operates with 16.6MHz bandwidth. Having the fixed network resources, a given network capacity and its performance, in turn, depends on the efficient utilization of these spectrum resources. For example, if the users have good channel conditions, they can employ higher modulation schemes to transmit more data bits for a given amount of network resources. In other words, good channel conditions allow achieving higher spectral efficiency.

Owing to the above facts, in multihoming scenarios, a reasonable strategy to increase network capacity is to serve a user over that particular network path which costs less network resources. This strategy is an advanced extension of the MaxT scheduling technique. The original MaxT exploits only the user diversity, while the proposed strategy exploits the diversity of multiple access networks in order to attain high spectral efficiency. This work adapts the term "network path cost" to represent the required network resources to offer a certain data rate. The network path cost is described in different units for different access networks. For example, it is described in the units of [second] in WLAN networks and in term of [PRB] in LTE networks.

The network path cost for a user can be computed using the pertaining channel condition information. Such information is accessible through cross-layer communication from the Media Access Control (MAC) layer of the corresponding access technology. This information along with the other knowledge about the design and operation of the access technology paves the way to an accurate estimation of the network path cost.



(a) Average HTTP user throughput (b) Average FTP user throughput (c) Average MOS value

Figure 2.2: TCP download throughput experienced by FTP and HTTP users

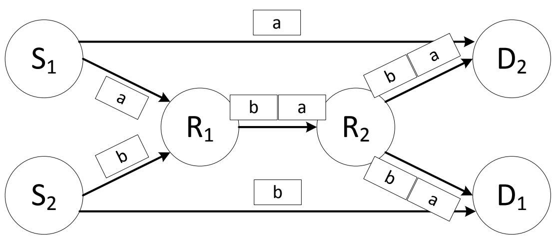
The development of analytical relations to compute the network path cost for LTE and WLAN paves the way for a problem formulation of optimized resource allocation using linear programming. In [9], the access network specific parameters will be incorporated to generate a full-fledged model applicable to the heterogeneous network of LTE and WLAN. Figure 2.2 shows that the proposed mechanism enhances the user Quality of Experience (QoE) for all real-time and non-real time application types compared to the 3GPP-HO approach, which always selects WLANs when they are available and only falls back to LTE networks otherwise. The detailed description of the linear programming model as well as its performance evaluation can be found in above mentioned referenced publication.

2.2.4 Network Coding and transport (TCP) over wireless

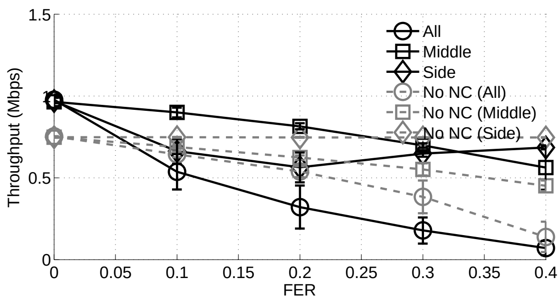
This mechanism brings about a tailored Network Coding (NC) solution which fits within the scope of the OConS architecture. Due to the ever-increasing evolution of communications, the classical network infrastructure paradigm becomes obsolete, thus showing the need of the migration to more open and flexible systems. This mechanism takes advantage of the openness OConS offers regarding the incorporation of new protocols and functionalities. In particular, it studies, by means of simulations, the impact that NC might have over TCP performance when applied over wireless mesh topologies, thus offering both a proof of concept of the integration of novel functionalities under the umbrella of OConS and an off-line simulation analysis which can be useful during the OConS orchestration configuration. On the other hand, the NC mechanism could benefit from its combination with, at least, routing processes (see Section 2.3.5) or multi-path (see Section 3.2.1).

The cornerstone of a NC scheme focuses on its ability to find the best coding/decodig equilibrium, this means that the coding nodes shall always choose the best packet combination in order to be successfully decoded at the destinations. This premise implies a tradeoff between encode as many packets as possible and decode as soon as possible as well (more details about the coding configuration are described in [10]).

This work evaluates the effect of the configuration of some Network Coding buffering parameters over the Butterfly topology, which is one of the most widespread ones that can be found in the literature, see Figure 2.3a. In this reference topology the receiving nodes (namely D_1 and D_2) overhear the transmissions from one of the sources (S_1 and S_2) so that are able to recover all the original information from the coded packets.



(a) Butterfly topology



(b) Simulation results over an error-prone channel (Buffer size = 10 packets)

Figure 2.3: Topology and results.

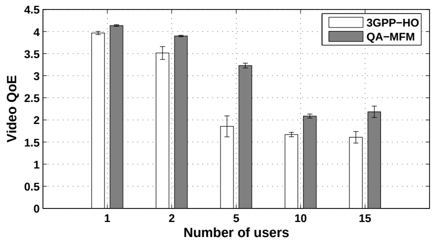
In order to evaluate the performance of the described procedure, the aforementioned scenario has been modeled with a network simulation tool. For this evaluation, the Figure 2.3b shows the data throughput through a traditional store-and-forward scheme, in which two different configurations have been defined: *Normal Links* refers to the most realistic scenario, in which every link is prone to bring about transmission losses; in *Side Links* configuration the packet losses links are only between $S_1 - D_1$ and $S_2 - D_2$, so only it has influence on decoding issues; finally, in the *Middle Links* configuration aims as showcasing the effect of the synchronization between two TCP flows, so only the link $R_1 - R_2$ is affected by errors.

By this way, Figure 2.3b shows that in the *Normal* case, when an error can arise everywhere, the NC yields a better performance until reaching Frame Error Rate (FER) ≈ 0.08. On the other hand, when errors are only prone in the *Middle* link, the NC scheme outperform the legacy one for the whole range of values. Finally, an unexpected behavior is observed in the *Side* configuration, since NC presents a decreasing trend until $FER = 0.02$ and starts growing from this value. This effect is explained by the fact that the receptor cannot decode encoded packets because of the lack on overhearing packets. More detailed results can be found in [10].

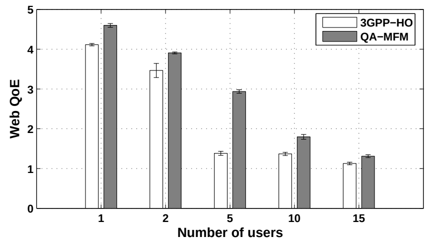
2.2.5 Mobile-driven QoE-aware Multihomed Flow Management

The goal of the Quality-Aware Multihomed Flow Management (QA-MFM) mechanism is to solve Multihomed Flow Management (MFM) problem [11] for end user devices. In order to provide on-line decisions (*i.e.*, solving the problem in real-time) and further evaluate the mechanism in simulation environments, the problem was formulated in [12] as a binary integer optimisation problem, as summarised in D.C.2 [2, Annex A4], and implemented within the OPNET network simulator,[2] interfaced with the CPLEX linear solver,[3] together with the LTE and OConS models from [13, 14].

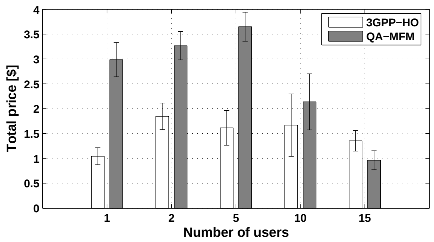
Simulations were conducted with one or more users running the decision algorithm in parallel, and both high-level user metrics as well as operational metrics were observed to evaluate this mechanism.[4] The results compare the QA-MFM proposal to the standard 3GPP-HO, which unconditionally uses WLANs when they are available and only falls back to cellular networks otherwise, considering multi-user scenarios of different number of users. All users in this scenario have a similar traffic model, composed of 3 web sessions, and 2 video flows.



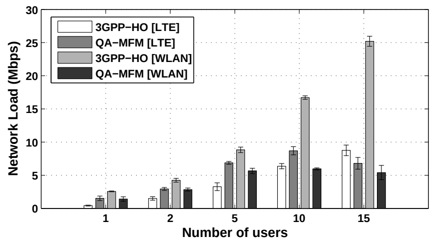
(a) Quality of Experience for video flows.



(b) Quality of Experience for web sessions.



(c) Average price incurred at the end of a run.



(d) Loads generated in LTE and WLAN.

Figure 2.4: Comparing QA-MFM with 3GPP-HO approach. Error bars show an estimate of the 95% CI for the mean.

Figures 2.4a–2.4c show that the QA-MFM manages to achieve a better application quality for both types of flows. However, in these simulations, it was not efficient with respect to costs. This can be explained by selected simulation scenario that, where the QA-MFM technique takes advantage of the opportunity of using more than one link. However, it can be seen from Figure 2.4d that the proposed mechanism allow for flow to be distributed in a more balanced way over the available networks.

[2] `http://www.opnet.com`

[3] `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`

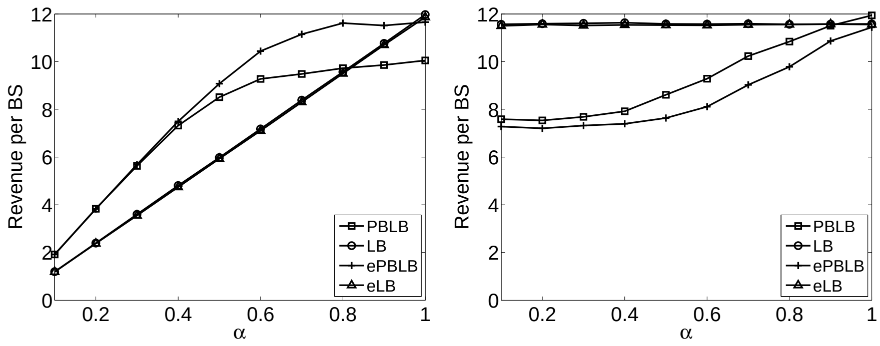
[4] LTE network: single eNodeB with a 5 MHz spectrum (25 PRBs), single cell with a 350 m radius; Wi-Fi network: 802.11g MAC with a 100 m radius; random direction user mobility model with 3 km/h movement.

Based on the OConS framework, the QA-MFM approach relies on communication between the decision system and elements of the surrounding networks (*e.g.*, through access points or eNBs). In the OPNET simulation model, this was done through an OConS layer, which exposes the relevant information from distributed elements (*e.g.*, network load) to the scheduling system. Providing terminals with such information from distributed sources facilitates better decision making, which is nowadays made with much more limited information and is, as the 3GPP-HO result show, less efficient in delivering a good QoE to the user and distributing the network load appropriately across various access networks.

2.2.6 Price based load balancing for wireless access networks

The goal of this mechanism, presented in [15], is 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 a specific Base Station (BS), 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 (for that they can use the *Enhanced Access Selection* OConS mechanism described in Section 2.2.1. We could think of the BSs running an OConS mechanism to establish the offered price, using the load they are currently carrying, which might be provided by the corresponding IEs.

With the above in mind, the study comprises two different aspects. On one hand, the strategy followed by the BSs consists in a piecewise decreasing function which adapts the offered price to the currently carried load, thereby encouraging or deterring the users to perform a connection, according to network load levels. On the other hand, we analyzed different selection strategies. In particular, three parameters have been considered: the first one prioritizes the cheaper access alternatives; the second parameter gives a higher score to low loaded BSs; finally, the third one fosters the BS currently being used, as a means to consider 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 already presented in Section 2.2.1). In this case, according to the different combinations of parameters, we can distinguish a *PBLB* (Price Based Load Balancing) strategy, where the users only take into account the price parameter, while in the *LB* (Load Balancing) strategy the users only take this parameter to carry out the decision making. The two strategies mentioned have been also studied in combination with the handover parameter, leading to *ePBLB* and *eLB* respectively (see [15] for more information concerning the decision configuration). Besides, it is worth mentioning that this study has been carried out by the mCASE simulator, presented in [5].



(a) Novel operator revenue (b) Traditional operator revenue

Figure 2.5: Access selection strategies performance from operator point of view

The most significant results are those which reflect the performance perceived by the operators since this mechanism pursues a load balancing by means of the offered price. Figure 2.5 shows the corresponding revenue, against the ratio (denoted by α) between the price offered by the traditional operator and the maximum one the novel operator can offer. For the traditional operator, the revenue always increases with α. Additionally, Figure 2.5b shows that the novel operator does not perceive significant additional gain when $\alpha > 0.7$; in this case there is a tradeoff between the reduction of the load and the price users are paying. Furthermore, when users try to keep the same BS as much as possible (ePBLB) we can see how the novel operator obtains some additional benefit.

2.3 OConS Network Level Mechanisms

2.3.1 Distributed Datacentre WAN Interconnectivity Mechanism (DDC-WIM)

The Distributed Datacentre WAN Interconnectivity Mechanism (DDC-WIM) has been described in D.C.2 [2], Annex A.7. It is a network level mechanism, which provides managed flows between the OConS edge nodes (Provider Edges (PEs)) for the use of distributed datacentres at the Customer Edge (CE).

By means of a unifying control element in each supported network domain – called Domain Control Unit (DCU) – the control functions are separated from the data forwarding functions by concentrating most of the control plane mechanisms and intelligence in a single entity.

The DCU incorporates PCE-like components for path computation and OpenFlow-like dynamic flow control to enforce the forwarding adjacencies in the switching or routing network. Additionally, the DCU provides intra-domain communication to the packet-forwarding network elements (e.g. routers, switches) in order to collect topological information and to react properly in case of failure. The persistency of the topology and resource information of the network is ensured by a Traffic Engineering Database (TED).

The DCU also supports the inter-domain communication between several domain controllers. As such, the concept of OConS-DCU provides a model also for the cooperation interface between CloNe and OConS. From a high-level point of view, CloNe defines the Flash Network Slice (FNS) concept with a full set of mechanisms to control how compute and storage resources are interconnected using network resources in a virtualised environment. OConS enhances the concept of the FNS by providing the necessary mechanisms to support advanced network features to enhance user QoE. This is further detailed in section 3.1.1.

In [16], the DCU concept and its experimental realisation is discussed for open connectivity services to cope with the complexity of multi-technology networks, especially with regard to control, management and algorithmic flexibility across layer-3 routing, layer-2 switching and in future even optical switching.

The OConS-based concept is discussed in comparison to plain IP, (Generalized)Multi-Protocol Label Switching (MPLS), PCE and OpenFlow.

For experimentation purposes we followed an OpenFlow-like approach to realize a flexible and extensible test installation of the DCU-based networking concept. This enabled us to demonstrate intra- and inter-domain path computation and flow establishment in a mixed environment of co-operating (layer2) switches and (layer3) routers in virtual and physical networks, as it is common in the context of interconnectivity of large distributed datacentres and cloud centers.

2.3.2 Address resolution mechanism for distributed datacentres (DDC-ARM)

More and more services are provided by large data centres 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 datacentre raises to cope with the increasing end-user demand. Current datacentre networks are usually based on Ethernet and mechanisms like load balancing or redundancy between datacentres require a transparent connection of these Ethernet networks over a Wide Area Network (WAN). Due to the large number of hosts, these interconnected datacentre 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 Address Resolution Protocol (ARP).

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

Distributed Datacentre Address Resolution Mechanism (DDC-ARM) has been introduced in D.C.2 [2] Annex A.8 and is based and dependent on the basic mechanism "Distributed Datacentre WAN Interconnectivity Mechanism (DDC-WIM)" as described in D.C.2 [2], Section 5.2 and Annex A.7. During orchestration the mechanism will be provided with the information of available attachment points to adjacent domains (datacentres as PEs), and about the principle connectivity between the edge nodes as a result of DDC-ARM orchestration.

Further on, the mechanism will be realized in OConS edge nodes and consists of:

- IEs that monitor incoming traffic from datacentre CEs and filter the ARP broadcast requests at OConS ingress nodes,
- Cooperating DEs or a DE that know about the connectivity of connected datacentres, and how to forward packets between the involved OConS edge nodes,
- Execution and Enforcement Entities (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 datacentre/ 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 Service Orchestration Process (SOP) for the DDC-ARM.

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 datacentres in terms of involved (private) IP hosts, and number of involved cloud components/sites (degree of distribution).

More details on the proposed mechanism DDC-ARM and of its performance evaluation are given in the Annex A.1 of this document.

An analytical model for the ARP traffic between datacentre locations is introduced that takes into account the number of hosts and connected sites. This model has been used to quantify the ARP traffic for a data centre interconnect solution, e.g. based on Virtual Private LAN Services (VPLS).

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 VPLS switches. The work has been published in [17].

2.3.3 Integration of OConS and CloNe

The selection of use cases around Datacentres by the OConS and CloNe work packages shows how the results complement each other and allow a very powerful integrated solution. The integration between CloNe and OConS allows a CloNe application to request a FNS, providing the high level description of the networking requirements foreseen in the CloNe architecture and delegate the management of the networking infrastructure to OConS.

Section 3.1 gives an in-depth description of the integration of CloNe and OConS. In short, the Datacentre (DC) interconnection use cases foresee the deployment of a Flash Network Slice which describes the resources used in the scenario in terms of compute, storage and communication infrastructure. The later are described with a high level of abstraction in what is known as a *Single Router Abstraction* (SRA). Implementing the Single Router Abstraction (SRA) using OConS allows for a tight control of the infrastructure by deploying different OConS mechanisms controlled by the Service Orchestration Process (SOP). The integration between OConS and CloNe foresees an interaction between the SOP and CloNe's Distributed Cloud (Control) Plane (DCP) to allow the coordination of different networking domains (represented by different SRAs).

The integration of OConS and CloNe enhances the SRA used in CloNe. The communications infrastructure provided to the FNS can be based on and managed by OConS. This implies that the mere best effort interconnection of end-points provided by the SRA can be enhanced and additional requirements like QoS/QoE targets can be imposed.

2.3.4 Dynamic Distributed Mobility Management

The purpose of this mechanism was to design and assess a distributed and a dynamic mobility scheme (see Annex A.11 from [2]); 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. Thus, the 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; tunnelling redirections are maintained on a temporary basis, as long as they are useful for delivering ongoing traffic flows initiated with an "old" IP address;

We have carried out evaluations both in simulations in OPNET (see [18]) and through a laboratory-scale test-bed (reported in [3]), using Linux-based platforms and Wi-Fi as radio accesses. More specifically, we have conducted simulations in OPNET with a dozen of access nodes, and around fifty mobile terminals. The following statistics were collected: the packet loss ratio, the end to end delay measured at the transport level (TCP/UDP), and the handover delay covering both L2 and L3 handovers. The results are reported in full length in [18]. As for the laboratory-scale test-bed, the handover delay and the packet loss ratio were measured and assessed; around 150 handover events were observed and the handover delay results were in line with those obtained from simulations.

Likewise, when mapping this mechanism onto the OConS architectural framework, the focus was put on the Execution and Enforcement Entity (EE), more specifically using a dynamic tunnelling approach to handle the mobility execution. We have also integrated this dynamic and distributed mobility execution mechanism with the enhanced access selection (i.e., mobility decision) mechanism from 2.2.1, using OConS messages and the OConS orchestration functionality, and we have demoed the prototype at several public events.

2.3.5 Network Coding for M-to-N Routing in DTNs

Efforts have been conducted within OConS towards analysing the impact of different social connectivity metrics on the performance of NC techniques, such as Epidemic Routing (ER) and Random Linear Network Coding (RLNC), when applied to broadcast and multicast communications in challenged network environments. This OConS mechanism extends the scope of previous ER and RLNC approaches in two directions: it introduces an adaptive algorithm based on inferred social metrics; and it relates the encoding decision not only with mathematical gain, but also with the history of human routines inferred. The information regarding human based connectivity patterns is retrieved via the OConS Delay Tolerant Networking (DTN) routing mechanism, the HUman Routines optimise Routing (HURRy) protocol (see section 3.2.2).

This mechanism combining connectivity patterns with NC techniques applied over challenged topologies has been analysed through simulation work. After social interactions have been parameterized, a set of Monte Carlo experiments was performed with the aim of estimating the delivery rate and the absolute number of transmitted packets by the proposed adaptive RLNC and ER in a one-to-all broadcast fashion. To this end, we randomly generate 100 original packets at time $t_s \in \{1, \ldots, \lfloor T/\alpha \rfloor\}$ and day i, which are then sent from source nodes (drawn uniformly at random) to the rest of the compounding nodes of the network by resorting to several different mechanisms:

A. Naive ER: conceived in its simplest form (i.e. flooding), each node continuously replicates and transmits its stored messages to newly discovered contacts. For the sake of fairness, no feedback of the already possessed messages from the newly contacted nodes is assumed, thus every node retransmits every incoming packet to its neighboring nodes, and stores in its buffer all such packets that are indeed innovative for itself. The number of transmitted packets at every node depends thus on the number of neighbors at the time tick at hand.

B. Naive RLNC: for every outgoing link, each node encodes the packets in its buffer by using a different linear combination in GF(8) after having solved – by Gaussian elimination – and stored all innovative incoming packets. As before, the number of transmitted packets at every node is kept fixed to the number of neighboring nodes at the time tick at hand.

C. Adaptive ER: this approach is similar to scheme A, but in this case only one copy of every incoming packet is sent to a neighboring node probabilistically based on its relative score with respect to other neighbors (i.e. in a similar fashion to the roulette-wheel parent selection mechanism in genetic algorithms).

D. Adaptive RLNC: analogously to approach B, in this scheme RLNC is used for processing the innovative packets at every node. However, only one encoded packet is sent to a single neighboring node based, again, on its score when compared to other neighbors.

Simulations conducted with this OConS mechanism have resulted in the verification that probabilistic prediction of the encoding decision outperforms broadcast and multicast transmissions in a DTN network. As results detailed in Annex A.2 show, the information related to the history of social interactions among nodes benefits the performance of content propagation in a hop-by-hop basis. In Wireless Mesh Network (WMN) or DTN topologies, where a local SOP can instantiate and combine several available OConS mechanisms, the combination of OConS-DTN routing with NC based on social routines would result in an enhanced performance by either increasing the packet delivery ratio, or reducing the one-to-all delivery time.

2.4 OConS Link level mechanisms

2.4.1 Innovative CQI Channel allocation Schemes for OFDMA Networks

This OConS mechanism proposes an innovative scheme for allocating Channel Quality Information (CQI) channels in Orthogonal Frequency-Division Multiple Access (OFDMA) network, ensuring most appropriate Modulation and Coding Scheme (MCS) setting. The MCS is adjusted for every transmitted frame according to the wireless channel condition of the intended receiver. 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, while allocations and de-allocations of CQI channels require expensive signalling messages, and therefore should be minimised.

Our mechanism optimizes this problem, as reported in D.C.2 [2] Annex A.19. 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. 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 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 complete allocation process, and showed with simulation its superiority:

- 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, discovered, registered, and launched during the bootstrapping phase. 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 network. 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:

- Virtual radio resource allocation, described in section 2.4.2. Our proposed mechanism operates at the physical level, while this mechanism operates at the virtual level
- Radio Resource Management of multi-radio WMNs connectivity, described in section 2.4.4. This mechanism is handling multiple wireless, while our proposed mechanism optimises one interface. The activity of those two mechanisms should be coordinated.

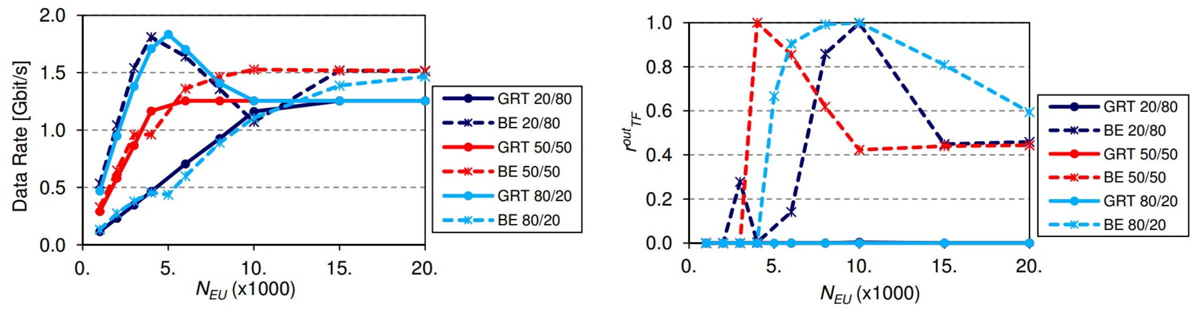
Otherwise, the CQI channel allocation mechanism does not collide or substitute any other OConS mechanism.

2.4.2 Dynamic Radio Resource Allocation for Virtual Connectivity

Two algorithms for VNet Radio Resource Allocation (VRRA) have been presented in [2]: Dynamic VRRA and OnDemand VRRA. Though in a slightly different approach, both intends to manage radio resource allocation to different Virtual Base Stations (VBSs), in order to ensure the amount of capacity negotiated at the Virtual Network (VNet) establishment, optimizing radio resource utilization. The former was evaluated in [19], for two VNets requesting guaranteed virtual connectivity (one providing low data rate services, and another high data rate services) and one providing best effort services. The VNets are instantiated in a cluster of heterogeneous wireless networks composed by 1 TDMA, 7 CDMA, 10 OFDMA and 20 OFDM BSs. Results from simulations show that the introduction of VRRA allows supporting the minimum bandwidth requirement in the cluster. For the VNet providing high data rates and guaranteed services, the Virtual Network Operator Satisfaction Level can be improved by 40% with the introduction of VRRA mechanism. VRRA takes advantage of the set of shared resources available in the entire common cluster, optimising their utilisation in order to maintain the contracted capacity. The number of VBSs composing the VNet has also been analysed in this work, as well as the strategy used to allocate radio resources in the physical infrastructure. Concerning these issues, results demonstrate that VNet performance depends not only on the number of VBSs, but also on the strategies used to instantiate the VBSs in the physical infrastructure.

The OnDemand VRRA mechanism was evaluated through simulation in [20] considering a reference scenario of one cluster composed by 2 TDMA, 1 CDMA, 4 OFDMA and 8 OFDM BSs. Three use cases were considered for the scenario with two VBSs deployed in the cluster. For each, the distribution of end-users in VBSs changes. By increasing the total number of end-users in the cluster, the satisfaction of data rate contracted for Guaranteed (GRT) VBSs and the isolation from the traffic in Best Effort (BE) VBSs was investigated. The simulation results for the use cases considered shows that even when the number of end-users in the network higher, meaning that the requesting data rate is much greater than the VBSs contracted data rate, the GRT VBS serving data rate can achieve the contracted data rate, Figure 2.6a. Although this is independent of the end-users distribution per VBS, the BE VBS serving data rate is dependent on the operation of GRT VBS. The GRT VBS is maintained within contract at the expenses of the BE VBS, which may operate out of the contract whenever the physical network conditions require so. In fact, the out of contract value, for the three use cases analysed, are always zero for the GRT VBS and increase for the BE VBS, depending on the mixing of traffic in both VBSs, Figure 2.6b. Results show that the OnDemand VRRA allows achieving isolation among the virtual resources, since the requested data rate of one VBS does not prevent the other to achieve the contracted data rate, if they are guaranteed. Independently of the requested data rate on each VBS, OnDemand VRRA allows to achieve the contracted data rate of GRT VBS, as soon as the amount of traffic demanded by its end-users require so. By another hand, if the traffic requested in GRT VBS is low the mechanism permit that end-users in BE VBS use the available capacity, optimising physical resource utilization.

The implementation of VRRA within the OConS architecture potentially brings the set of benefits as follows. Enabling mechanisms reconfiguration allows to streamline the process of adapting at runtime the changes of requested capacity for the virtual resources. The communication capabilities among functional entities inherent to OConS nodes, permit to set triggers in the IEs of the group of nodes in the cluster, actuate over the several schedulers, and also inform the cluster manager automatically about changes occurring within the cluster. The OConS capability to launch a service composed by several mechanisms allows to instantiate, e.g., VRRA with the mechanism in 2.2.6



(a) VBS average serving data rate for different number of end-users.

(b) VBS out of contract for different number of end-users.

Figure 2.6: Evaluation resuls of the VRRA mechanism.

or the one in 2.2.3 to optimise the resource utilisation according to end-users policies maintaining the capacity requested for the virtual resources.

2.4.3 Cognitive Radio Systems through Spectrum Sensing Techniques

This mechanism has the objective of using spectrum sensing techniques in order to derive useful information regarding wireless channel occupation. The intention is to detect and identify wireless channel activity and provide this kind of information to any interested decision mechanism. In order to improve the efficiency of wireless networks, the scientific community has made an effort to develop and exploit the opportunistic access to the physical environment through Cognitive Radio technologies. The Network Nodes have the functionality of Spectrum Monitoring or Spectrum Sensing, which corresponds to the IE as defined in the OConS architecture. According to the capacity of each node, the sensing process can be applied to a single channel K, or to a wideband signal composed of several radio channels (Wideband Spectrum Sensing). By applying signal power estimation techniques such as energy detection or waveform-based power estimation, the signal power level S is obtained.

The mechanism described has been mapped to a simulation set up as depicted in Figure 2.7 (right), where 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 manager 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.

A Simulation testbed of 1 to 8 sensor nodes was developed considering the three following propagation channels (see detailed results in Annex A.3):

- Additive White Gaussian Noise (AWGN): The simplest one, in which only additive white Gaussian noise is considered.
- Rayleigh Flat Fading Channel: a single tap channel following a multipath propagation features.
- Shadowing: The logarithmic attenuation of propagated signal follows a Normal distribution. Empirical measurements demonstrates the suitability of this distribution for real static sensing environments.

After OConS 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 occu-

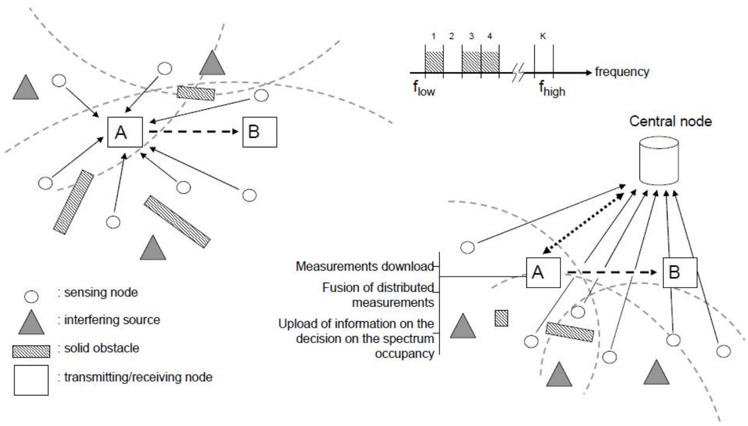


Figure 2.7: (Left) Example of the considered cognitive radio network scenario addressed; (Right) Mapping from the general setup depicted on the left to the technical architecture envisaged in OConS

pied 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 Signal-to-noise ratio (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(coordinator node) 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. In order to improve the accuracy on the hard decision, a cooperative sensing testbed has been developed and evaluated through simulation testbed. Detailed results of this evaluation are reported in Annex A.3. The mechanism described is specially tailored to be combined with OConS enhanced access selection (see 2.2.1, which may result in an enhanced and very specific channel allocation and sensing information based selection criteria.

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

WMNs are an efficient and low-cost solution for providing last-mile broadband Internet access in areas without fixed infrastructure. A novel OConS supported mechanism for radio resource management of multi-radio WMNs connectivity is proposed. It does 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, channel assignment and flow control mechanisms to efficiently guarantee max-min fair capacity to every node. The mechanism was described according to the mechanism-level architecture in deliverable DC1 [21] as well as in [22]. Its main features were described in deliverable DC2[2].

The mechanism was evaluated in [23]. It exploits 100% of the network capacity, achieving the maximum throughput, minimising spectrum and power usage. For a network of 13 dual-radio mesh nodes based on IEEE 802.11a, by using OPNET Modeler, one concludes that 4 channels and a total transmitted power of 34.5 dBm are sufficient to achieve an aggregated throughput of 4.8 Mbps per mesh point. It outperforms several well known strategies for allocation of resources.

An analysis of various hexagonal deployments was also performed in [24]. It shows that this mechanism exploits 100% of the system capacity, guarantees a maximum fair throughput per node, and minimises spectrum and power usage. Using 802.11a for mesh and 802.11b for access, for a 100 m radius scenario a WMN throughput of 57.7 Mbps is achieved with an hexagonal deployment of 19 MAPs, 4 channels for mesh and 3 for access, and transmitted power levels of 33 dBm for the GW and 16 dBm for the remaining MAPs. For a density of 10 000 users/km2, where 5% are active, this corresponds to an offered capacity of 3 Mbps per user.

In [25], its performance is evaluated by considering IEEE 802.11a in a challenging spontaneous neighbourhood community scenario of 20 randomly deployed mesh nodes with 3 gateways to the Internet. Mesh nodes achieve 100% of their allocated capacity, with throughputs ranging between 5.7 and 11.3 Mbps (depending on the specific propagation conditions), no packet loss, and end-to-end delays below 20 ms. It is shown that, thanks to the proposed strategy, max-min fairness is achieved by every mesh node in the share of capacity, there is no interference between nodes, and resources such as power and spectrum are efficiently used. It is concluded that the performance of WMNs is strongly limited by the system's characteristics and not by the WMN intrinsic characteristics of the multi-hop environment and flow of traffic, which can be overcome by the proposed Radio Resource Management (RRM) strategy. Several system improvements (MIMO, higher modulations and receiver sensitivities) result in higher system rates and ranges, enabling higher theoretical WMN capacity levels that can be reached using the proposed RRM mechanism.

OConS services can be built combining this mechanism and access selection noes such as the Access Selection and decision mechanism, Section 2.2.1, or the Optimized Resource Allocation mechanism for Integrated 3GPP and non-3GPP Networks, Section 2.2.3. The combination of this mechanism with other OConS mechanisms to provide an Opportunistic WMN Resources Management OConS service (OWROS) is presented in [26], of special interest for flash crowd scenarios, as described in detail in Section 3.3.3, where is shown how this service increases overall connectivity, coverage and capacity.

2.5 Benchmarking Algorithms

The benchmarking algorithms are optimisation or experimental studies that are executed over a simulated or experimental networks, in order to gain better understanding, guidelines, and benchmarks for the specific problems researched. These algorithms are not part of the real-time OConS logic. Rather, the results of the benchmarking algorithms are used as meta data by the OConS orchestration functionality, to better select and configure the OConS mechanisms for optimized performance, and thus better fulfil the needs of the applications/users. In some other cases, the benchmarking algorithms provide guidance regarding network infrastructure upgrades or configurations which enhance connectivity. The following subsections presents the results of the benchmarking algorithms studied, and elaborate on the implications of those results to connectivity services and to OConS

2.5.1 Resource Management within Heterogeneous Access Networks Benchmarking analysis

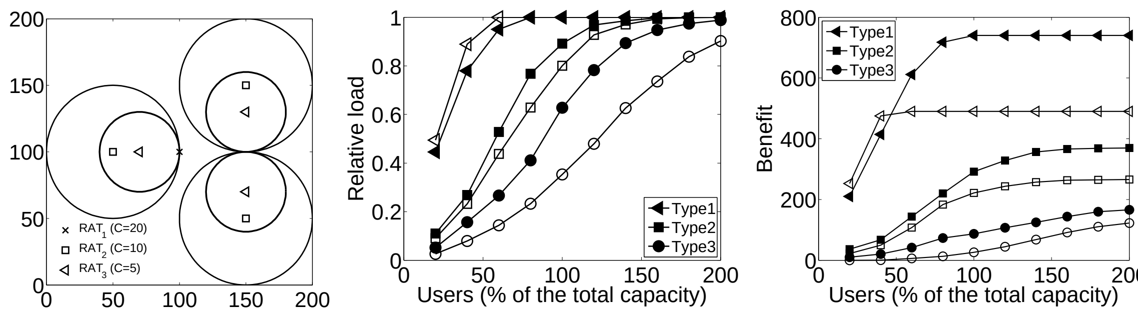
As mentioned in [2], the goal of these two benchmarking studies, which used Game Theory, was establishing the optimum resource management strategies to be used by base stations and access points within wireless heterogeneous access environments. The obtained results establish the performance bounds that the real-time OConS mechanisms implemented by the corresponding elements might expect.

We consider an area with N access elements, characterized by their coverage and capacity. We use a generic and discrete load (and capacity) unit. Two different deployment (random and deterministic) strategies were used (see Figure 2.8a), 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' coverages (provided that there is at least one end-user within them).

In the first one [27], which focused on resource allocation, 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. 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 [28] 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.

First, Figure 2.8b depicts the load for the three Radio Access Technology (RAT) types which are being used (first analysis - allocation of resources). First of all, it is worth highlighting that the highest capable RAT (RAT_1) does not perceive any benefit after applying the NEP strategy, and it can even slightly reduce its carried load. Nonetheless, its capacity is completely filled for a relatively low offered load (end-users around 60% of the overall network capacity). On the other hand, for both RAT_2 and RAT_3, the NEP strategy brings about a remarkable gain; for RAT_2 it allows reaching a full load earlier than the naive one; besides, in the optimum strategy, RAT_3 resources get fully occupied, which is not the case for the naive strategy. The results also yield that the highest *relative* gain (after applying the NEP strategy) is perceived by RAT_3 (less coverage and fewer resources). We also analyzed [27] the impact on adding other parameters (link quality and price) within the decision process, and the obtained results yielded that the use of the optimum strategy can lead to better performance in both cases.

On the other hand, for the second analysis (pricing policies), Figure 2.8c depicts the benefit for the three RAT types which are being used. First of all, it is worth highlighting that the highest capable RAT (RAT_1) gets the greater gain after applying the NEP strategy, but for situations in which users are fewer than the 40% of the network capacity, where the benefit is very similar (or even slightly worse).



(a) Network Scenario (b) Network load per RAT type (c) Benefit obtained per RAT type

Figure 2.8: Network deployment used during the analysis and performance comparison between NEP and legacy strategies

2.5.2 Multi-path Benchmarking: the Trade-off Between Control-Plane Load and Data-Plane Efficiency

One of the main promises of SDN is that it allows an efficient convergence between circuit switching and packet switching [29]. 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; A traffic aggregate requirement is satisfied with one or more paths (circuits), which necessitate path setup and maintenance. 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.

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 (as reported in citeDC2, Annex A.6); it is desirable to minimize the number of Link State Packets (LSPs) or the number of routers crossed by these LSPs, in order to reduce the control load.

Our research defined a formal model to the problems, and proposed a few algorithms:

- 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.
- the same problem while minimizing the number of nodes is shown to be NP-complete. We identified an approximation algorithm for it, reaching the same approximation ratio
- The complexity of the control plane load minimization with minimum number of paths problem is NP-complete. We designed two approximation algorithms and identified their computational complexity and their worst-case performance guarantees. Via simulation, we bad average performance of those algorithms is not good, and therefore designed a third algorithm that does not have a worst-case performance guarantee, but its actual performance is shown to be good
- 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 [30], as well as real ISP topologies from the RocketFuel project [31]. 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.

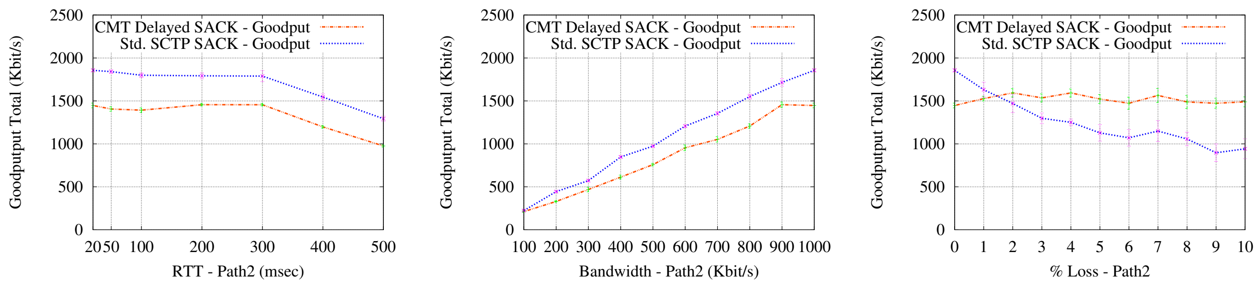
We identified the algorithm that minimizes the number of nodes traversed by the decomposed path. The algorithm that decomposes the flow into the smallest number of paths traverses about 20 percent more nodes than the one that minimizes the number of nodes. When the number of nodes is of primary concern, 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 [32] 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 [32].

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.

2.5.3 Multi-Path and Multi-Protocol Transport for Enhanced Congestion Control

We present here results from investigation simulations described in D.C.2 [2, app. A2]. This study aims at identifying cases where multipath mechanisms of SCTP can provide an improvement over single-path use, and investigates the particular impact of delays, capacity and reliability assymetry along the paths. It showed that overall throughput and goodput performance of multipath transmission decreases as the asymmetry between the paths increases (Figure 2.9). This study also highlighted that performance of asymmetric multipath transmission can be notably improved by increasing the selective acknowledgements (SACK) frequency. Indeed unordered SACK packets that trigger spurious retransmission of data packets and incorrect round-trip time (RTT) estimation can worsen the overall performance.



(a) RTT Asymmetry. As the RTT of Path 2 becomes more different than that of Path 1 (20 ms), the goodput of the whole transfer reduces.

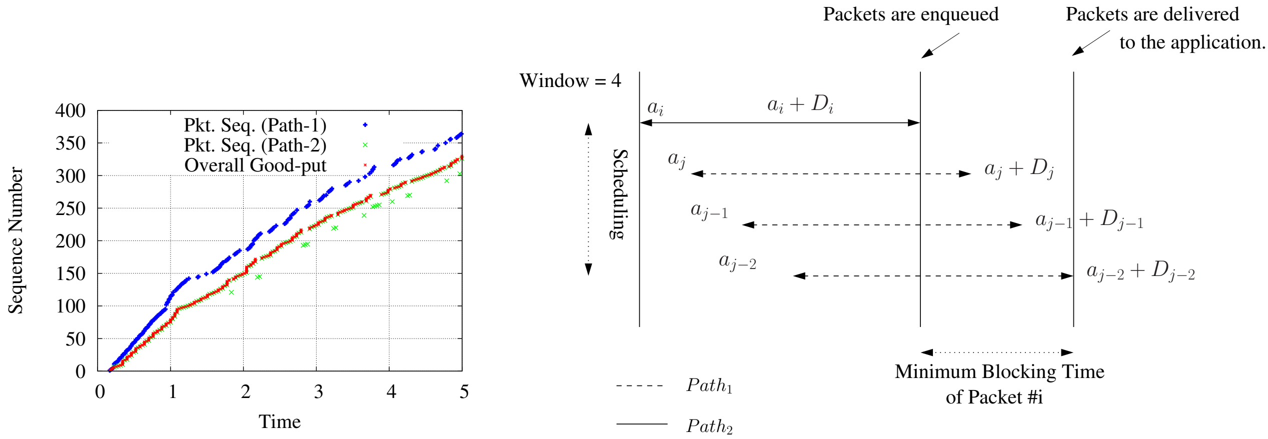
(b) Bandwidth Asymmetry. Despite providing a larger capacity on Path 2, the goodput of CMT-SCTP on it is upper bounded by that of Path 1 (900 kbps)

(c) Loss Asymmetry. CMT offers a better loss resilience despite an increasing unrealiability of Path 4.

Figure 2.9: Performance Evaluation of RTT, Bandwidth and Loss Asymmetry. Overall, delayed acknowledgement are detrimental to the goodput of CMT-SCTP.

Additionally, these experiments demonstrated how receiver's buffer blocking together with unordered packet queuing at the receiver's buffer can become a performance bottleneck for the application goodput which defines the ultimate quality of experience for the mobile device users as can be seen in Figure 2.10a. We have derived a simple model to compute the receiver's buffer blocking time, a major cause of performance degradation, when no packets can be transmitted by the data sender as receiver's buffer is fully occupied (as shown in Figure 2.10b),

$$T_{block} = \max_i \{0, \max_{j<i}(a_j + D_j) - (a_i + D_i)\}.$$



(a) The overall goodput is controlled by the worst path.

(b) Blocking model illustration. Until all packets have been received and ordered by the transport, they are not acknowledged nor delivered to the application.

Figure 2.10: Symptom and analysis of the multipath blocking problem. As the receiver expects packets retarded on one slow path, both congestion windows are blocked , and the sender cannot leverage the available capacity on the other path to increase its goodput.

In allowing a better information about the available paths, the OConS framework allows to only start a multipath transmission when it is worthwile, and avoid the cases illustrated here where the goodput would not be improved by such a mechanism.

2.5.4 Policy-based Routing Enhancements

One of the main reasons that Border Gateway Protocol (BGP-4) 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, while a direct path from A to B may be legal and a direct path from B to C may be legal, 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 new routing middle points or service gateways, forming an overlay routing infrastructure. These in-network devices can be used by the flows to enable shortest path or to validate path concatenation. 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), does not depend on the graph size or the number of pair nodes. We found that in current BGP-4 implementation, 50 percents of paths are inflated by as much as 40 percents (as compared with shortest path possible).

We examined the practical aspects of the scheme by evaluating the gain one can get over the

BGP-4 routing problem, and showed, using up-to-date data reflecting the current BGP-4 routing policy in the Internet, that a relatively small number of overlay servers are sufficient to enable routing over the shortest paths from a single source to all ASs. More specifically, a set of 900 overlay nodes can enable shortest path routing, reducing the average path length from 3.8 to 2.5 hops (40 percent improvement), as shown in Figure 2.11, for 12 randomly-selected source nodes. Interestingly, a set of only 100 overlay nodes is sufficient to enable 80 percent of the total path length reduction, reaching an average 2.8 hops on average.

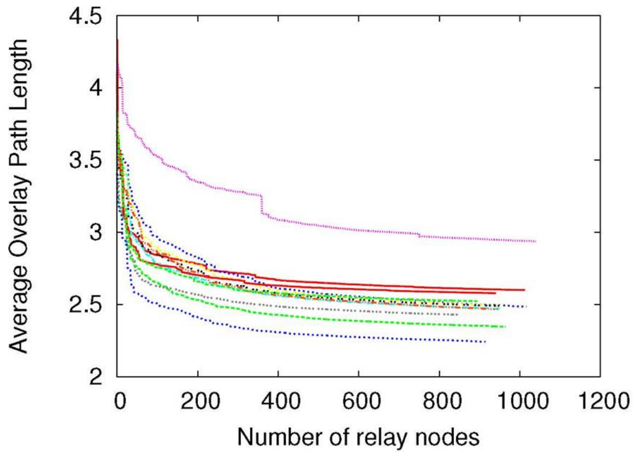


Figure 2.11: Average path length Vs. number of BGP-4 overlay nodes

3 Applications for Open Connectivity Services

This Section presents the application of OConS in the wider scope of the SAIL project. More specifically, we show how the OConS architecture that was presented in [2], and some of the mechanisms whose results were discussed in section 2, bring benefit for three use cases: OConS for CloNe, OConS for NetInf and OConS support to the SAIL project *flash-crowd* scenario. Each of these use cases is presented in a separate subsection, evidencing the applicability of the OConS scheme to real connectivity problems, and how OConS enhances them.

3.1 OConS for CloNe

As an integrated project, SAIL has developed use cases as proof of concept, where components in the different work packages (WPs) are integrated into a single scenario that should show the potential of the SAIL approach. In particular, this Section shows how OConS is integrated with CloNe in order to enhance the DC interconnection scenario and implement mobility support for terminals accessing the interconnected DCs.

We present the implementation of the OConS/CloNe integration proposed in D.C.2 [2]. Further information regarding the architectural aspects of this integration can be found in D.A.3 [33]. We also discuss how OConS can provide some additional advantages when applied to the particular case of the mobile access to CloNe applications.

3.1.1 OConS for CloNe Architecture Consolidation

The harmonised SAIL architecture and the relation between OConS and CloNe are presented in D.A.3 [33]. From a high-level point of view, CloNe defines the FNS concept. The FNS provides a full set of mechanisms to control how computing and storage resources are interconnected using network resources in a virtualised environment. This networking layer is abstracted with the SRA.

OConS enhances the concept of the FNS by providing the necessary mechanisms to support advanced network features to enhance user QoE. As shown in Figure 3.1, the CloNe layer manages the establishment of connectivity between the DCs. The user provides a high-level view of the interconnection scenario and this is split among the different service domains (i.e. the DCs as the endpoints and the interconnecting networks) using the Infrastructure Service Interface (ISI)[1]. The Distributed Cloud Plane - Link Negotiation Protocol (DCP-LNP) would then provide additional information between peering domains in the connection.

OConS provides the mechanisms to monitor the connection, detect breaches in QoE parameters and trigger possible re-configuration actions. Once an OConS domain has been configured, the SAIL architecture allows for OConS or CloNe to propagate changes to all domains involved in the end-to-end connection. The natural implementation will take advantage of the orchestration facilities provided by the OConS architecture.

Figure 3.2 shows how the interconnection between mixed OConS-CloNe domains works. In the case of a pure CloNe domain, inter-domain signalling could be implemented using the DCP-LNP interface. Changes forced by OConS orchestration mechanism that impact the CloNe components can be signalled to pure CloNe domains using the DCP interface.

[1] The practical implementation of the ISI uses the Python OCNI (pyOCNI) library.

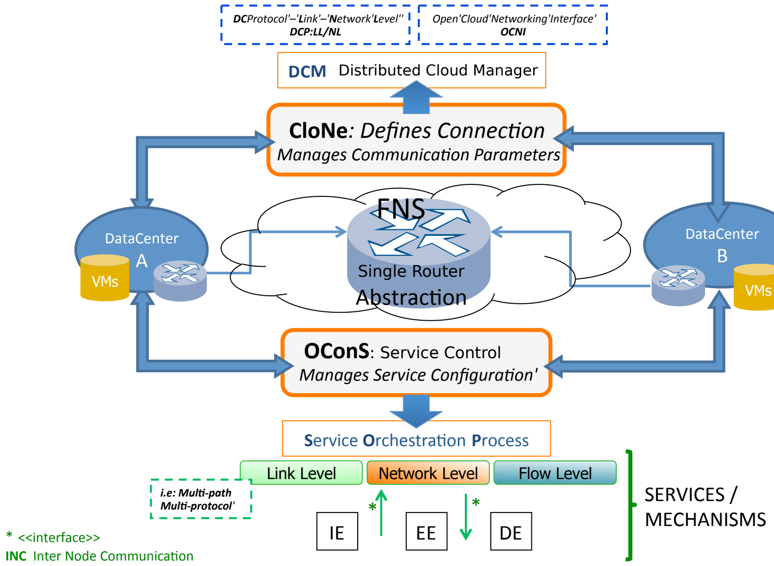


Figure 3.1: The relationship between OConS and CloNe

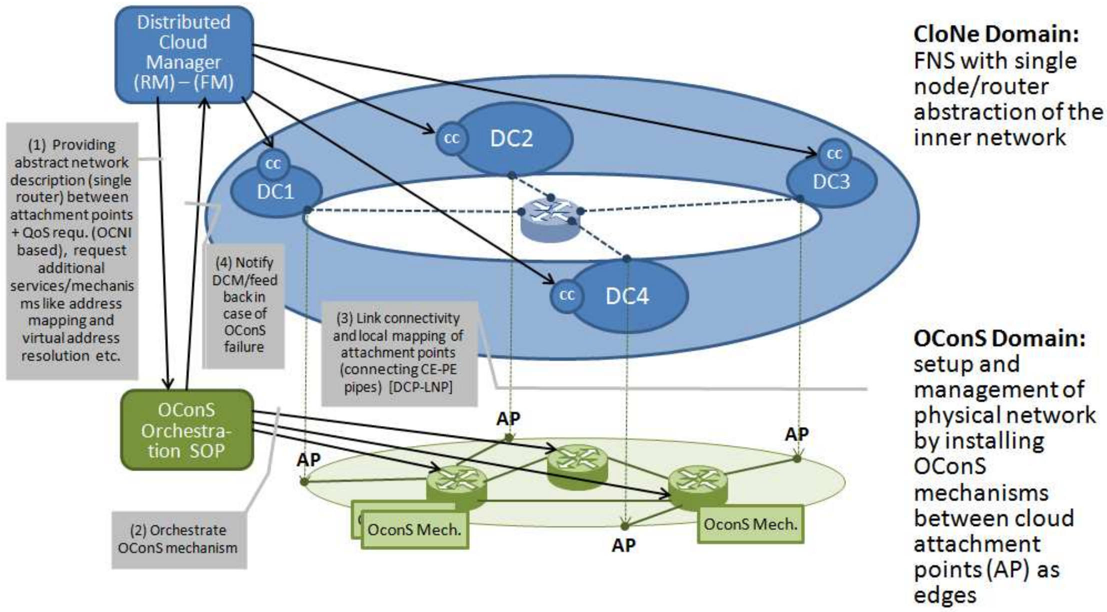


Figure 3.2: Interconnection using OConS and CloNe

The interoperation between CloNe and OConS works in mainly four phases (for the CloNe architecture, refer to D.D.3 [34]):

1. The Distributed Cloud Manager (DCM) as representative of the CloNe DCP will delegate the CloNe resource demand descriptions for storage, processing and datacentre internal connectivity to the datacentre local Cloud Controller (CC). These resources will be provisioned via the Open Cloud Computing Interface (OCCI). At the same time, the DCM will request complementing network resources from its network provider. In the current scenario, this network provider is the OConS domain, represented by the entry point, OSAP, of the corresponding

SOP. The network resource demand description in Open Cloud Network Interface (OCNI) is sent from the DCM to the SOP. The contents of such a demand description covers the involved data links between the PE/OConS and CE/CloNe [the attachment points AP], a possible (optional) SRA representation of the connecting network between the different CloNe attachment points, and optional flow demand descriptions between selected CloNe attachment points.

2. When the OConS-SOP receives the demand request from the DCM-Resource Management (RM) module, it orchestrates and initiates the involved mechanisms in the affected OConS nodes of the own network domain.

   The range of required mechanisms may comprise:

   - at data link level: providing the DCP-LNP engines at the PEs; providing network address translation capabilities; providing address resolution capabilities; providing traffic policing and performance monitoring capabilities.
   - at network level: transforming the 'virtual' SRA description of the connecting network into a realisation by the physical routing or switching network (let it be IP in its simplest case, or MPLS, GMPLS, VPLS, or any other managed technology like OpenFlow/SDN); configuring the involved forwarding nodes with the respective rules.
   - at flow level: computing (and optimizing) the requested path resource allocation (between CloNe attachment points) with given QoS and policy constraints and establishing, monitoring and maintaining them across the OConS nodes.

   An example how such a demand description may look like has been given in [2].

3. Once the initial orchestration of OConS mechanisms and nodes is finished, the OConS mechanisms start operating. An initial action will be the establishment of the data link connectivity between the attachment points, via DCP, to exchange dynamic link configuration parameters. For this, the addresses of the respective CloNe control entities are required.

4. During the operation, OConS will maintain the QoS level requested by CloNe autonomously as long as possible, possibly by rearranging and re-allocating OConS-internal resources (e.g. paths) such that the abstract external description is still satisfied. In case the QoS level cannot be maintained any longer, a feedback message, including the root cause to notify and to trigger CloNe recovery actions, is created and sent to the DCM-Fault Management (FM) for further actions in CloNe.

3.1.2 OConS for CloNe with Focus on Mobility

The OConS access service can be used to support seamless access to CloNe services for mobile users. We consider scenarios where a mobile user requests access to a cloud service in the CloNe datacentres via one or multiple access networks. The mobile user may have multiple interfaces to connect with various wireless access technologies (*e.g.*, Wi-Fi, 3G, LTE or WiMAX). Different access networks are available with overlapping coverage areas resulting in a heterogeneous network environment. Furthermore, the user may request several different Cloud services at the same time (*e.g.*, video, VoIP or file downloads) and each service may have different QoS requirements or expected perceived quality (QoE).

Multiple access selection and flow management mechanisms have been proposed and defined within the OConS framework, whose results are reported in sections 2.2.1, 2.2.3, 2.2.5, and 2.3.4. Each of them addresses different aspects, and results in different decision functions. To orchestrate a CloNe service, any of these mechanisms can be used through OConS to provide enhanced connectivity to mobile users. Moreover, taking into consideration the functions and benefits provided by different mechanisms, multiple mechanisms can as well be combined to provide an overall improved and more flexible service.

One example is that the access selection mechanism (selecting the access networks) and the Multi-P mechanism (distributing the traffic of one application flow among a number of selected access networks) can be combined through the OConS orchestration to further increase the utilization of the available access networks and thereby improve the transmission capacity. In this case, the decision of access selection mechanism (a list of selected networks made by the access selection DE) can be used 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, which could guide the process of mechanism selection and or configuration. If needed, it could also retrieve some information that has been obtained by the access selection mechanism about the current status of the datacentres, or about the end-to-end paths to be used to reach them.

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 datacentre throughout network access changes.

By using different OConS access selection and flow management mechanisms via orchestration, a CloNe user can benefit from OConS facilities.

- 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, utilizing 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, seamless mobility and session continuity to the cloud service.
- By optimizing the network resource usage, the overall network costs can be reduced, which is gain desired by the network operators.

The next section provides some quantitative results of the potential benefits OConS can bring to the CloNe mobile access.

3.1.3 OConS for CloNe - Benefits from the Application's Perspective

As described in the section 3.1.2 above, OConS access selection and flow management mechanisms can be jointly used to enhance the connectivity to CloNe service for mobile users, providing seamless mobility and session continuity to the cloud service.

Section 2.2.5 shows, as a proof of concept, the detailed results of applying OConS Quality-Aware Multihomed Flow Management (QA-MFM) mechanism in scenarios where a mobile user requests seamless access to a service (e.g., a cloud service in the CloNe datacentres) via one or multiple access networks.

Figure 2.4a and Figure 2.4b demonstrate that the QA-MFM mechanism achieves a significantly higher QoE for both types of applications than the 3GPP-HO approach (currently most widely implemented in mobile devices: it has a default preference towards WLAN, and only uses the cellular network if no other connectivity is available).

Furthermore, Figure 3.3 shows that the QA-MFM mechanism achieves a significantly higher QoE by greatly reducing the loss rate for application data. The QA-MFM technique achieves this improvement over 3GPP-HO by adequately reducing the bit-rate of the video flows to what the

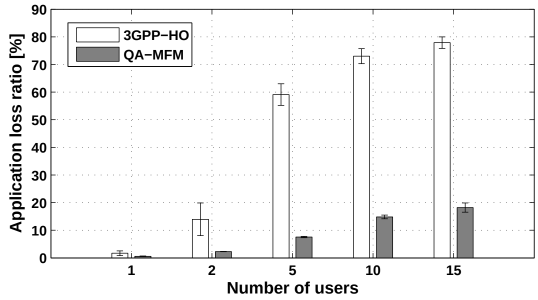


Figure 3.3: Application loss rate for the video flows. With lower coding rates, the QA-MFM approach succeeds in reducing the loss rates on more congested networks.

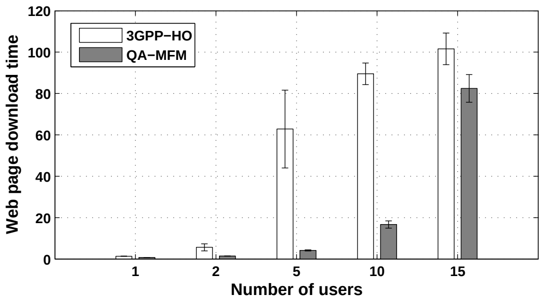


Figure 3.4: Download times for web sessions achieves by both approaches. As the QA-MFM technique can choose to use more than one network at once, it can more effectively distribute the current traffic, and achieve lower transmission times.

network can support given its current load, thereby avoiding additional traffic into the otherwise congested network, which resulted in reduced packet losses. With reduced losses and resultant TCP retransmissions, the web application also experiences a much lower application delays, as seen in Figure 3.4.

Moreover, it can be seen that the QA-MFM mechanism can distribute the application flows in a more balanced way over the available networks and thus can balance the load of each network, as seen in Figure 2.4d (Section 2.2.5).

From these results, it is clear that with the combination of access selection and flow management mechanisms enabled by OConS, the mobile users can achieve better connectivity to the CloNe service. The mobile users enjoy lower delay and losses, seamless mobility and session continuity to the cloud service leading to an improvement of the quality of the service (QoS/QoE).

Finally, the OConS access selection and flow management mechanisms are also able to use the available network resources much more efficiently, incurring lower network costs, while also balancing the load in a fair manner.

3.2 OConS for NetInf

The second use case applies OConS to NetInf. First we depict an integration of a Multi-path OConS mechanism, which is used to enhance the NetInf performance. Then we discuss how a routing procedure that is enriched with social information can be used by OConS to enhance the NetInf connectivity over wireless challenged scenarios, where DTN protocols are used.

3.2.1 OConS for NetInf with Multi-Path

Multi-path connectivity to networks in modern computing devices is becoming the norm in todays computing. These multiple paths can be used in many ways to benefit the users of these devices as well as the different service providers involved. The Information Centric Networking (ICN) architectures that are currently being defined such as NetInf, have considered the use of multiple paths natively. Although these architectures provide the simultaneous use of multiple paths, no formal mechanisms have been defined to utilise them in the best possible manner. The framework and the multi-path content delivery for NetInf mechanism identifies a number of multi-path strategies that utilise the multiple paths to request and receive content with NetInf, in the best possible manner.

NetInf has the built-in capability to perform retrieval of content over multiple attachments or paths. The OConS mechanism together with the components of its framework steers this built-in capability of NetInf in retrieving content. The Decision Elements (DE) of the OConS framework located in NetInf-based devices make decisions that are related to the selection of the best multi-path strategy to use. There are 3 types of strategies:

- Splitting - This strategy relates to distributing the NetInf content request (GET) messages of a single content stream between the multiple paths
- Replication - This strategy replicates the GET messages to the multiple paths
- Distribution - This strategy distributes the GET messages of different content streams to the multiple paths

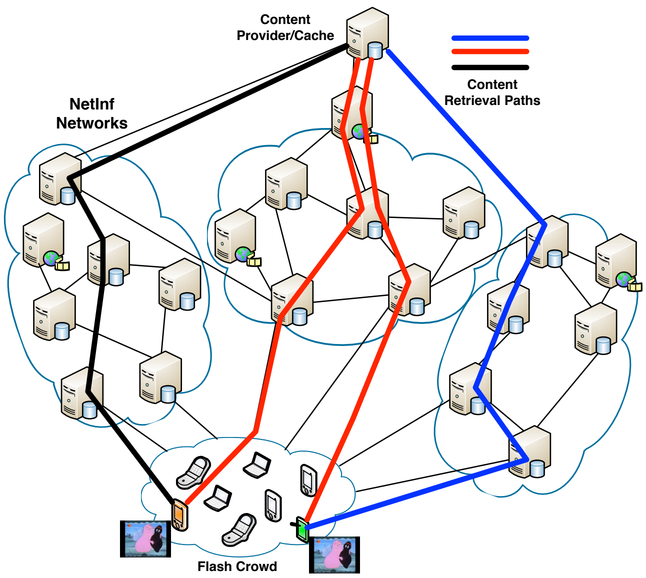


Figure 3.5: Demonstrated OConS Multi-path NetInf Scenario

A proof-of-concept prototype has been developed for this OConS mechanism to demonstrate the feasibility of the identified multi-path mechanisms. There were a number of internal and external demonstrations done with the prototype. The demonstrations were based on the Event with Large Crowd scenario (Figure 3.5) developed in the SAIL project. The scenario consists of a flash crowd that requires downloading content based on spontaneous decisions that are related to the interests of the participants of the flash crowd. The participants use NetInf-enabled computing devices and some of these devices are deployed with this OConS mechanism. These devices are equipped with multi-path support and the operation of the OConS extensions demonstrate the use of multiple paths to deliver the required content.

3.2.2 OConS for NetInf with DTN routing

From an OConS perspective, NetInf performance could be better perceived and supported if extra components are considered to be deployed in the Event with Large Crowd (EwLC) scenario. Specifically, including the HURRy protocol (section 2.3.5) as an available mechanism to be instantiated together with the DTN suite, could bring an ad-hoc optimised packet forwarding among the mobile human carried devices present in the EwLC. Moreover, the implementation of Bundle Protocol Query (BPQ) extension for DTN stack over Android phones provides a seamless integration of NetInf requests over a DTN communication, by using each intermediate device as a local cache for any content exchanged. [35] describes the specification of the BPQ extension block, which allows applications to query the content stored at nodes on the path along which a bundle containing a bundle query extension block is routed. The BPQ extension is intended to allow such queries that can be answered by intermediate nodes, where those nodes do not necessarily have to be addressed by the destination of a corresponding request message. The final purpose of integrating the HURRy protocol with the NetInf implementation of BPQ is to show the cooperation between previous knowledge of social encounters among nodes and the concept of intermediate caching of demanded content.

On one hand, the HURRy routing protocol takes a forwarding decision, making use of a subset of human-related properties that characterise the expected connectivity opportunities to be established among people gathered in the EwLC scenario. On the other hand, OConS support for NetInf comprises the orchestration of this service by selecting and combining both mechanisms (BPQ extension block and HURRy protocol) when available in each single node. See [2], where a representation of how DTN nodes act as NetInf caches and process BPQ requests and responses is depicted. OConS-DTN service has been implemented for a proof-of-concept validation, and the open source project has been published under [36]. The implementation details of this prototype are described in [3], whereas the remainder of this section summarises some evaluation results of HURRy performance.

In order to validate the functionality and evaluate the performance of the OConS support mechanism for DTN routing, the HURRy protocol was implemented in a simulation environment specially designed for opportunistic networks: *The ONE* simulator. The complete environment of *The ONE* is available in [37], as well as some sample scenarios and documentation. Our simulation work consisted in taking the Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) release for *The ONE* simulator as a starting point, and develop the HURRy modification of functional blocks in Java code to be integrated and compiled in the programming environment. Apart from specific functionality tests we executed for the validation of the protocol components, we aimed at the simulation of a proof-of-concept scenario where the enhancements proposed could be validated and compared with the performance of PRoPHET. We selected the scenario represented in Figure 3.6 , where Node A must find a route towards Node C to gain access to the outside world. If Node A requests a specific content (e.g. a video from Youtube) from the Internet, it will need to decide how to reach Node C, that is, decide which intermediate neighbour would most probably reach the destination address faster or in a more reliable way. DTN nodes implementing HURRy are mobile, and they register information about who contacts whom, at what frequency and for how long. The nature of these encounters regarding frequency or duration will vary depending on specific features of the scenario considered. For this set of simulations we considered Bluetooth technology and the four nodes represented in the picture (A, B, C and D), where Node A intends to send information packets to Node C, but there is no permanent path established from A to C. Position (1) in the picture represents the initial situation where Nodes A and C have no neighbours (they are out of range of any other surrounding node), and Nodes B and D are connected through a physical link. When the simulation starts, Node C is continuously moving back and forth from position (1) to (2), so the links established by Node C with A and D are intermittently active. Moreover, the movement of C is quite fast so the contacts between C and D, and C and A, are very short but with a high frequency. On the contrary, Node B has a slower pace: it alternates positions (1) and (2), but once the link with D is broken in (2), Node B establish long connections with A before getting back to (1).

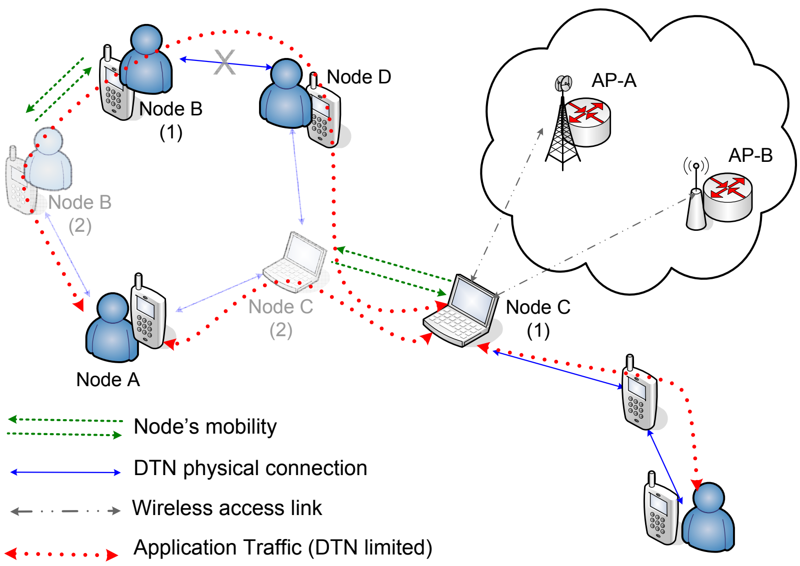


Figure 3.6: Dynamic DTN scenario with diverse wireless connectivity

One of the first results obtained from the comparison between PRoPHET and HURRy in this scenario regards the convergence time needed for all nodes to be aware of the whole topology. If we define the convergence time, *conv_time*, as the time period until all nodes learn about all the rest, and assume generic time units, *t.u*, HURRy outperforms with a gain factor above 2.5:
$conv\_time(PRoPHET) = 746.9t.u$
$conv\_time(HURRy) = 281.8t.u$

This is due to the fact that, for instance with HURRy, node B learns about node C during the first contact between C and D (the link B-D is still active), which does not happen with PRoPHET until the second contact. The same happens for the rest of intermittent connections.

The simulation setup includes the following configuration:

- Node A creates 8 information packets headed to node C, and C creates 3 packets headed to node A (11 packets created)
- The packet size is 15MB
- The transmission rate of Bluetooth links is 250kbps

Table 3.1: Results in terms of # of packets delivered

PRoPHET	**HURRy** $\gamma = 0.05$		**HURRy** $\gamma = 0.5$		**HURRy** $\gamma = 0.95$	
	P.G	H.G	P.G	H.G	P.G	H.G
0	9	9	5	8	3	3

Table 3.2: P_(k,C) values compared by node A

	PRoPHET	**HURRy** $\gamma = 0.05$	**HURRy** $\gamma = 0.5$	**HURRy** $\gamma = 0.95$
P_(A,C)	0.902714	0.183423	0.137867	0.1401
P_(B,C)	0.4572134	0.342112	0.147759	0.06143

- The γ parameter has taken three possible values: 0.05 (priority to contact duration); 0.95 (priority to contact frequency); and 0.5 (intermediate balance)
- The intervals predefined for the histogram of contact duration times have been configured with different granularity: (P.G) poor granularity (contact durations below 5t.u are not distinguished); and (H.G) high granularity (v_i of 0.5t.u, 1t.u and 5t.u are distinguished)

Table 3.1 shows the relevance of considering the contact duration in the routing decision. In this scenario, the packet size is considerably large, taking into account the transmission rate, so the frequency of direct contacts between nodes A and C forces PRoPHET to select the direct path as the best route, but in reality those direct contacts between A and C are too short for the messages to be successfully delivered. HURRy performs differently according to the balance configured for the priority associated to the frequency and the duration of contacts, but at least some of the transmission attempts are successful in all configurations. If $\gamma = 0.05$ HURRy is merely rating available contacts according to their duration, and so, node A is selecting node B as its best next hop to reach C. That is the reason why 9 out of 11 packets are delivered, even with poor granularity in the duration intervals. The opposite configuration with $\gamma = 0.95$ implies that HURRy is prioritizing the frequency of contacts, just like PRoPHET, but the difference in the results obeys to the fact that HURRy selects the upper path in Figure 3.6 until a number of encounters between A and C has occurred and then, the direct probability $P\_(A, C)$ increases its value. The intermediate configuration, $\gamma = 0.5$, shows the importance of defining an appropriate precision for the scenario considered. In this case, the number of packets delivered increases considerably if HURRy performs with high granularity in the range of duration intervals.

The comparison of the evolution experimented by $P\_(k, C)$ in the scenario during simulation time is also quite revealing. Table 3.2 presents the final status of the key probability values evaluated by node A when selecting a route towards C. All results in Table 3.2 referred to HURRy correspond to the case of high granularity (H.G).

It can be observed that node A will always choose the direct path towards C with PRoPHET, since $P\_(A, C) > P\_(B, C)$, and that is the reason why none of the transmission attempts succeeds, because the packet is too big to be delivered within the short duration of each contact between A and C. The probability values obtained with HURRy depend on γ, of course: the comparison between the two possible paths results in $P\_(A, C) < P\_(B, C)$ when $\gamma = 0.05$ and $\gamma = 0.5$; but if $\gamma = 0.95$ the final status ends with $P\_(A, C) > P\_(B, C)$, like in PRoPHET. The rating difference is much higher for the case $\gamma = 0.05$, which is the most opposed to PRoPHET. On the contrary, when HURRy uses a similar prioritization to PRoPHET, it is only in the beginning of the simulation that certain packets manage to reach node C.

Table 3.3: P_(k,C) values compared by node A with different granularity

	HURRy $\gamma = 0.05$		**HURRy** $\gamma = 0.5$	
	P.G	H.G	P.G	H.G
P_(A,C)	0.422297	0.183423	0.250865	0.137867
P_(B,C)	0.436876	0.342112	0.178831	0.147759

Finally, we would like to highlight some results associated with the precision defined for the intervals of the contact duration. Table 3.3 shows the different values obtained for $P\_(k, C)$ for configurations of high and poor granularity.

For $\gamma = 0.05$ the comparison in Table 3.3 results in $P\_(A, C) < P\_(B, C)$ both for poor and high granularity, although the difference is much bigger for the H.G case. When $\gamma = 0.5$, the comparison provides opposed results depending on the granularity defined. Provided that the intermediate case is trying to balance the rating parameters considered, an appropriate granularity to distinguish contact durations with high precision is the key factor influencing the final probability values. Thus, Table 3.1 showed that if $\gamma = 0.5$, the H.G case obtained a delivery ratio of 8/11, whereas the P.G case delivered only 5 packets out of 11. Table 3.3 states the reason for such a different performance.

3.2.3 OConS for NetInf - Benefits from the Application's Perspective

As it has been discussed before for the CloNe use case, the main advantage of the OConS framework is its ability to smartly combine some mechanisms so as to offer a better service to its end-user. In this section, we have discussed its applicability for NetInf.

First, it is worth highlighting the reusability approach which OConS fosters. Any developer of NetInf module would be able to reuse the OConS entities. The multi-path example which has been presented before, illustrates this benefit. In some cases (mobile access, for instance) a NetInf module would require to take decisions (e.g. on which accesses/interfaces to use); in this sense, the information which OConS could provide would certainly leverage a better decision.

In a tighter integration, OConS would be able to offer yet more benefits. We have discussed the use of multi-path scheme or a DTN routing. In both cases, if an advanced coding scheme (*Network Coding*) was available, the SOP would be able to orchestrate the two mechanisms, so that a better performance can be offered, transparently to the corresponding end-user.

The use of *Network Coding* has shown to bring about benefits for peer-to-peer communications, as well as for multicast. A study which was presented in [10, 38] showed, however, that its use for Wireless Mesh Networks, albeit potentially beneficial, might be severely affected by the presence of errors (wireless propagation environments). In this sense, it seems reasonable to assume that the information which might be provided by the corresponding OConS entities might be appropriate to modulate the operation of the *Network Coding* elements. This is just an illustrative example of the potential benefits that reusability with OConS offers.

On a more generic perspective, OConS would offer a NetInf developer the possibility to bypass any connectivity problem. NetInf deals with the optimum retrieval of content from the network (from the various sources where it is available). In order to be able to establish connectivity to such source(s), several mechanisms might be used, depending on several factors; as was described in [2], OConS aims at orchestrating the corresponding connectivity mechanisms, so as to offer the end-user a better service. In this section we have discussed two particular cases: multi-path and DTN. However OConS use goes far beyond these particular examples, provided that the corresponding mechanisms are integrated with the OConS architecture. In that sense, the NetInf application, by means of the OSAP interface would instruct the SOP to establish a connectivity with one (or more) destination(s)[2]. Depending on the corresponding manifest and the particular network state, OConS would establish the corresponding connectivity resources which will be then used to retrieve the content. Note that the NetInf application would not even require to be aware about whether a single or multiple sources are being used.

3.3 OConS in the Flash Crowd Scenario

Finally, we conclude this chapter by identifying some of the potential benefits which OConS can bring to the flash-crowd scenario. First, we discuss the potential advantages that the networking paradigm fostered by OConS architectural framework contribute to this scenario. Then, section 3.3.2 illustrates the particular benefits that the various OConS mechanisms and a combination of them can offer in this challenging scenario. Finally, Section 3.3.3 describes a specific OConS service that is highly adventurous to the scenario, integrating two OConS mechanisms and one legacy mechanism under the OConS architectural framework.

3.3.1 Enhancements Provided by the OConS Architecture

The concept of flash crowd designates an expected or unexpected large group of people with mobile devices in a location where there is an increased demand for communications and services. Their requirements for communications services and content are dynamically changing, but these have to be available for everybody and provided with the appropriate quality. The OConS architecture follows several design principles and presents several characteristics that enable the optimisation of connectivity in such a flash crowd scenario.

The OConS architecture enables the dynamic and flexible management of connectivity, rapidly reacting and adapting the network to changes (traffic, topology, application requirements) brought by a flash crowd. Its orchestration functionality provides an unified and abstract access to mechanisms, which can be combined and launched as an OConS service that exploits the particular conditions of the scenario. OConS is designed to coexist with legacy networking technologies. 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 which is controlled by an OConS service. Non-OConS nodes can also be upgraded to OConS nodes with OConS-related software. These are important characteristics in scenarios like a flash crowd, where heterogeneous nodes may join a network of enhanced capabilities and actively participate, by being assigned novel connectivity functions, if they can. The OConS architecture supports and explores the heterogeneity of flash crowd's nodes capabilities and resources. Within OConS nodes, and thanks to the modularity of the orchestration and the well-defined interfaces, it is easy to share and use both resources and capabilities to launch connectivity services adequate to the environment and network state.

The OConS architecture breaks the rigidness of classical architectures, supporting novel connectivity solutions, transport paradigms and communication protocols that exploit the connectivity conditions. A classical adverse condition can be turned into an advantageous connectivity condition, by launching adequate connectivity mechanisms. Thanks to the modularity of the OConS entities and its clear interfaces, OConS supports cross-layer optimisation, easily combining and exploring mechanisms that touch different layers of the classical protocol stack. Its well inter- and intra-node communication (INC) procedure and interfaces that are supported by any transport paradigm, enables an easy establishment of a control plane among all OConS nodes. This enables the discernment of the most appropriate solutions to launch, localised or global ones, brought by

[2] We assume that the locator which is provided has been already translated to a name which can be handled by OConS

link, network and flow level mechanisms. The combined support of both legacy and novel mechanisms is a strong characteristic, supporting solutions being built on existing Internet foundations, but enabling novel ones. The OConS architecture is easily scalable, supported by a light control plane that rapidly orchestrate, spread, launch and control adequate mechanisms in new-coming nodes. The open monitoring procedures brought by IEs, which can be re-used by multiple mechanisms, are capable of sharing network states between mechanisms, also helping in the scalability of the network, which is of most relevance for the flash-crowd scenario.

The OConS orchestration provides the possibility of distributed or centralised management of connectivity, choosing what better fits in each situation. In particular, in a flash crowd where most nodes have no orchestration capabilities, a centralised solution is the best. In the opposite, in a flash crowd of highly capable nodes, distributed and autonomous orchestration enables the launch of adequate localised solutions. In both cases, OConS takes the best of every node's capability, in terms of communication, but also processing (by implementation of orchestration functionalities) and storage (for storing monitoring information). OConS also supports opportunism, essential in flash crowd networks which are typically wireless challenged networks. Communication conditions are adverse, e.g., expectations of connectivity between certain nodes no longer holds, or congestion is experienced on some links because of the multiple simultaneous requests from the crowd. OConS supports innovative techniques that explore resources and communication conditions in the best way to create and sustain the connectivity.

Finally, by evaluating the performance of the various solutions proposed by OConS, it is clear that OConS provides enhanced and new connectivity mechanisms that are beneficial for the end-users and their applications, as well as for network operators. End-users enjoy better QoS or QoE with adapted connectivity, while network operators experience a more efficient usage of resources, higher throughput, and load balancing, which are collectively contributing to more satisfied users. It supports both self-organised community networks or global networks supported by service providers.

3.3.2 An Overview of the Enhancements Provided by the OConS Mechanisms

This section shows how individual OConS mechanisms and a combination of them are enhancing the performance and the user experience of the flash crowd scenario. The scenario naturally benefits from multi-homing access selection and multi-path mechanisms. However, it also benefit from less obvious mechanisms, addressing distributed mobility management, TCP-aware handovers, distributed datacentres, virtual radio resource allocation, and even network coding and CQI channel allocation, as elaborated for each of them.

Nowadays mobile devices support different access technologies in providing the best possible access to the internet. Smart phones and tablet PCs (e.g. Apple Ipad) have the capabilities to support different networks access technologies, such as 3G, 4G or WiFi networks. Currently these mobile devices can communicate using one access technology at a time. However there is a big potential for improving network capacity and enhancing user QoE if these access technologies are integrated together. Such integration would make access technologies cooperate and work simultaneously in a heterogeneous environment from which both the end users, as well as the mobile operators can benefit.

The first step usually embraces the need to select an access within a heterogeneous environment (like the flash-crowd), where the end-user is able to connect to a broad range of alternatives. The enhanced access selection mechanism reported in section 2.2.1 fosters a generic decision process. Various parameters of merit can be combined into a single utility function and the DE within the end-user is using this to select the access element to connect to. Depending on the particular characteristics of the scenario, this combination might be tuned so as to ensure that the end-user always select the most appropriate scenario.

Another Quality-aware access selection is addressed by the following mechanism. Though the Quality-Aware Multihomed Flow Management (QA-MFM) mechanism reported in section 2.2.5 is a user-centric approach, it can as well enhance the overall network performance in a flash crowd scenario where many users accessing the network simultaneously, in means of (i) reducing the network congestions by adequately adjusting the application parameters according to the available network capacity; (ii) utilizing the available networks in a balanced manner. The results presented in Sections 2.2.5 and 3.1.3 demonstrate that in case of large number of users in the network, compared to the legacy approach, the QA-MFM mechanism can avoid creating more congestions into the networks and thus keep lower packet losses and delays, and on the other hand it also distribute application flows more fairly over the available access networks and thus balance the load in different access networks. Therefore, by applying the QA-MFM mechanism in the flash crowd scenario, not only the user performances (*e.g.*, better QoE) can be improved, but also the network congestion can be reduced and network loads can be properly balanced among all networks.

In some cases, the access selection process can be combined (orchestrated) with other mechanisms. In this sense, the Multi-P mechanisms (section 2.2.3) is a very beneficial solution for the flash crowd scenario. With the help of the optimized resource allocation provided by the Multi-P mechanism in integrated networks, users can be provided with more reliable connectivity in congested flash crowd scenarios. The overwhelming number of users in the flash crowd can cause poor services or even outages for the mobile network operator in that region. However, with the Multi-P mechanism, the users can use multiple access technologies and the operator can do network load balancing. In this work, it is investigated how to tackle the simultaneous usage of wireless access technologies.

It is fair to assume that in the flash crowd scenario, significant portion of the traffic stays locally among the terminals of the participants; thus, on the mobility execution part, the Dynamic Distributed Mobility Management mechanism (section 2.3.4) can also bring benefits; e.g., no IP-level handover executions messages, and no encapsulation/de-capsulation operations are needed if a given terminal stays anchored to its initial access router. The mechanism still assures the IP-level session continuity for applicative flows that need this continuity (and which is not handled at the applicative level, e.g., SIP-sessions).

The evolution of the cellular LTE networks towards smaller cells increases the network efficiency and throughput. It is clear that in dense locations such as the flash crowd scenario, small cells are highly beneficial, and the network is capable to provide usable service to a larger number of subscribers. However, due to the small coverage area of each small cell, an increased number of handovers is expected, even within the flash crowd location. Adaptive video streaming is one of the popular application for the flash crowd scenario, where people share the on-site experience by means of streaming video. This is where the forward admission control mechanism, reported in section 2.2.2 is highly beneficial. Our mechanism selectively forwards the video packets that are destined to the departing subscriber (that moved to a different eNB), depending on the state of the TCP connection and the availability of wireless resources, thereby maintaining transport-layer session and minimizing the disruption of the video session.

Obviously, a significant part of flash-crowd scenario is the communication with datacentres. The OConS mechanism DDC-ARM presented in section 2.3.2 supports the 'flash crowd scenario' by concentrating the connectivity management in cloud datacentres, in which the required processing and storage resources can be increased and reduced, but also relocated between the datacenters according to the actual 'flash' traffic demand. Following this approach, DDC-ARM supports the mobility and migration of datacentre processes hosted on Virtual Machines (VMs) across distributed datacentres that are connected via WAN. By introducing an additional architectural entity at the OConS level, the ARP proxy, our evaluation shows that this ARP proxy is an effective method to reduce the ARP (control) traffic caused by the necessary address resolution procedure between those interconnected datacentres.

The Virtual Radio Resource Allocation mechanism presented in section 2.4.2 is helpful in flash crowd scenarios since they support isolation between several virtual resources deployed over a given area. In fact, by creating virtual resources with guaranteed capacity, the VRRA mechanism makes this capacity available for the end-users, granting use of the virtual resource independently of the traffic in the rest of the network ([19], [20]). Hence, even when a flash crowd occurs, the mechanism adapt the radio resource allocation to the virtual resources in order to maintain the serving rate contracted for the guaranteed ones, allowing end-users connected to those virtual resources to receive the requested service.

In some other cases, communications will be established between the terminals of the end-users, leading to multi-hop networking environments. The Network Coding mechanism which is presented in section 2.2.4 aims at providing better performance for the end-users (either by increasing the reliability, security or improving the capacity), by smartly combining the contents of various data packets at intermediate nodes, which only transmit these *'coded'* packets. The possibilities which are leveraged by this mechanism are broad, and it can be nicely orchestrated with other mechanisms (e.g. wireless mesh networks, routing, DTN, etc). As can be seen, most of these mechanisms are quite relevant for the flash-crowd scenario. The results which we obtained (see [10, 38] for further details) yield that the performance which might be expected from the application of this mechanism are severely degraded when the presence of errors due to the wireless propagation impairments are not negligible. This result sheds light to the goodness of the mechanism and can therefore be seen as a benchmarking for it.

OFDMA networks are expected to be deployed as one access technology at the edge of the flash crowd scenario. Supporting the QoS requirements of each flash crowd application despite varying channel condition is of prime importance. Our CQI channel allocation mechanism reported in section 2.4.1, implements a highly effective scheme to allocate CQI channels that dynamically support the most appropriate MCS setting for each frame, thereby providing the best-possible phy-level support for QoS demand of the real-time flash crowd applications.

Finally, the policy-based routing enhancement benchmarking algorithm reported in section 2.5.4 ensures, among other things, routing along the shortest path within the AS-AS core network, and it can be thus used to enhance the performance of the communication between DCs. The flash crowd scenario includes real-time communication of voice and video applications, where end-to-end packet delay is of prime importance. Shortest path routing reduces the number of hops and consequently the packet delay, thereby accommodating a possible larger packet delay at the highly populated access, thus making real-time communication more feasible in the flash-crowd scenario.

Ins summary, we showed in this section the rich set of OConS mechanisms that enhances the flash crowd scenario. The performance of some of these mechanisms is orthogonal (i.e. independent from other mechanisms). The Orchestration functionality can combine the appropriate set of the available mechanisms, to provide the best possible OConS service in this scenario.

3.3.3 An OConS Service for a Flash Crowd Scenario

Spontaneous community-based WMNs networks, of random deployment and distributed management, present many connectivity and coverage challenges. When a flash crowd of end-users with heterogeneous devices intends to access such a network these limitations are evidenced even more. A WMN organizational framework is explored, which exploits conditions and properties of the network as well as capabilities and specificities of each node. Following the OConS architecture, an opportunistic WMN resource management service is proposed. It is able to orchestrate, when possible, networking functionalities such as end-users access provisioning, Internet gateway connectivity, and mesh forwarding connectivity. It is shown through simulation how this OConS service, when offered by the community-based WMN, can be orchestrated in nodes willing to join and cooperate in the community WMN. The presence of an end-users flash crowd in a community-based WMN scenario can be beneficial in the improvement of coverage, capacity, and connectivity, when the proposed OConS service is offered by the community to joining members.

We now present the OWROS service. Then we will evaluate its impact as related to the flash crowd scenario.

3.3.3.1 OWROS OConS Service

An OWROS is proposed [26] for spontaneous community-based WMNs, based on an organisational framework [39] for the flexible and opportunistic formation and maintenance of WMNs and supported by the OConS architecture. This framework relies on the concept of self-organization and collaboration. The traditional WMN architecture nodes role separation constraint is relaxed. Nodes can flexibly act as Wireless Mesh Router (WMR) or Wireless Mesh Client (WMC). The case of spontaneous network formation is considered, relying on the concept of self-organization. The main idea is to make the network take advantage of the specific resources and characteristics of heterogeneous nodes in an opportunistic fashion. In our vision, any wireless node (WMR and WMC) can perform any network functionality, if they *can* and if they *wish*. According to WMCs characteristics and resources — communication capabilities (types and number of radios or fixed connection), energy autonomy, computational capabilities, storage space, surrounding environment mobility pattern, etc. — these nodes may collaborate in the network in which they self-organize and share duties by taking networking tasks and becoming so-called Super Wireless Mesh Client (SuperWMC).

Being a community network, all nodes have one common objective: make the network working as efficient as possible. By introducing this flexibility, spontaneous networks are likely to respond better to the expected services. An opportunistic resources manager gathers information relative to the different network functionalities and task requirements. It then suggests specific functionalities and/or tasks that the nodes are free to accept or not.

OWROS is available in the community-based WMN, an OConS domain, being offered to every node of the community. This is a distributed and self-organized service. Members willing to collaborate take tasks in the network according to their capabilities. In fact, the heterogeneity of multi-radio nodes' characteristics opens a wide set of connectivity possibilities, which go much beyond the classical two-tier WMN architecture, where nodes are either routers or clients. The OWROS OConS service explores the capabilities of each node and the network functional needs, suggesting one or multiple of the following OConS mechanisms:

- **Internet gateway provisioning mechanism**: this is a legacy mechanism, where the node has an interface (radio or Ethernet) with connectivity to the Internet. This mechanism enables the node to provide gateway connectivity to the WMN.
- **Client access provisioning mechanism**: this is a classical mechanism, where the node acts as a classical radio access network, covering a region where it offers connectivity to WMCs. It optimizes the operating bit rate, power and channel resources. The node needs to have an available radio interface capable to offer such service.
- **Mesh forwarding connectivity mechanism**: this is a novel mechanism named Fair and Efficient Resources Allocation mechanism (FERA), described in [25], for multi-radio nodes that optimizes rate, power and channel radio resources, guaranteeing a max-min fair capacity to all aggregating nodes. The node must have one or more radio interfaces, capable of forwarding traffic. The implementation of this OConS mechanism is described in detail in [25] and [2]. In this section we only consider its orchestration with the other two mechanisms to build the OWROS OConS service.

Next, the orchestration of the proposed OWROS service is described. Every WMR of the OConS domain has the knowledge of the functional networking needs within its neighbourhood. It has also registered, in its OR (so-called domain OR) the three above mechanisms that may compose the

OWROS connectivity service (gateway, access, and forwarding), with the associated requirements. The service is offered to members of the community. Depending on the node capabilities and functional network needs, the service suggests specific network functionalities that the node is free to accept or not. This is clearly different from the classical WMN architecture, where WMCs would never implement such functionalities.

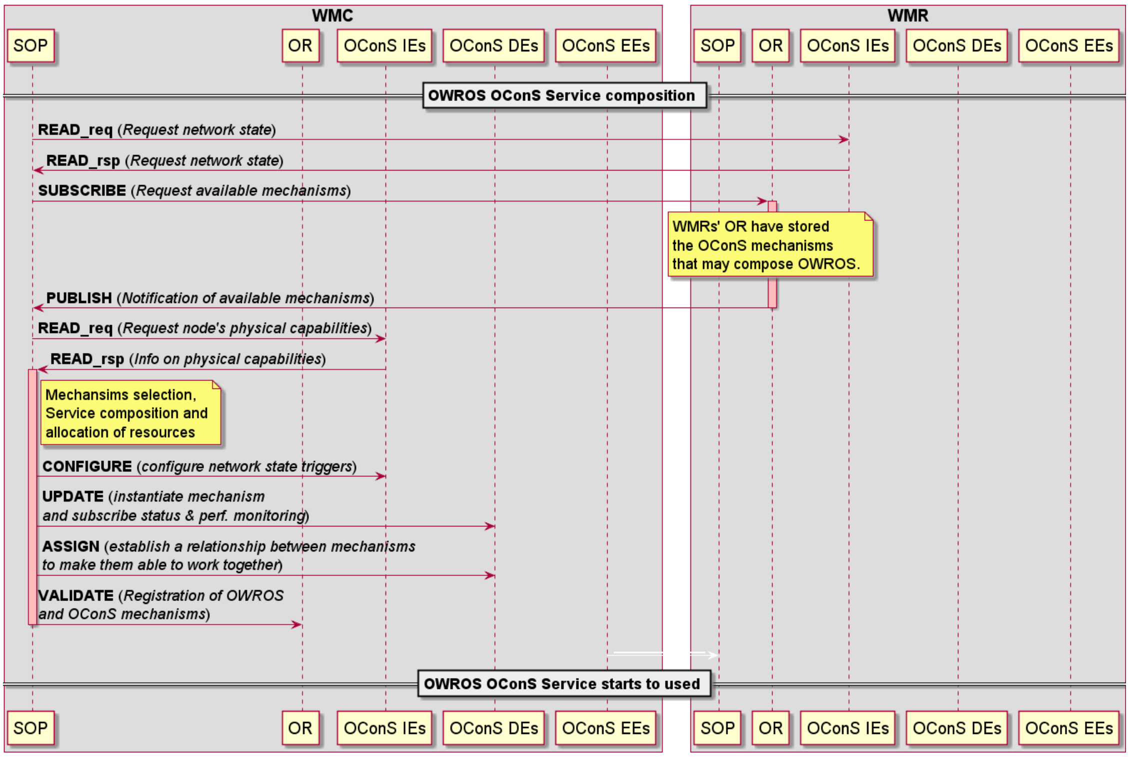


Figure 3.7: Orchestration of OWROS OConS Service in a Wireless Mesh Client.

For a new member of the community, two phases exist: bootstrapping, and orchestration of the OWROS service. It is assumed that under normal circumstances, the bootstrapping phase has been successful. A new member of the community that is not an OConS node is upgraded with OConS-related software by the SOP of a neighbouring WMR. Each node is aware of the available OConS entities. To implicitly orchestrate the OWROS service in a new member of the community, various interaction procedures are needed, as depicted in Figure 3.7:

- **Monitoring of functional needs**: a neighbouring WMR informs the node about the state of the network and the functional networking needs. Subscription of candidate OConS mechanisms: subscription, by the node's SOP towards a domain OR, of the OConS mechanisms available in the OConS domain. The domain OR publishes in the node's SOP the available mechanisms (gateway, access, and forwarding ones).
- **Physical space characteristics request**: The network state information is subscribed by the SOP towards IEs, as well as OConS entities capabilities, to specify the node's physical space characteristics, namely communication capabilities (one or several radio interfaces, of similar or different standards, Ethernet connection), surrounding environment (e.g., geographic position, propagation environment, and nodes in communication range), mobility pattern, persistence, energy autonomy, and computational capabilities.
- **Validation of OConS mechanisms**: the physical space characteristics and available OConS entities will validate which candidate OConS mechanisms are supported by the node.
- **Selection of OConS mechanisms**: Within these mechanisms, based on a set of rules for

mapping connectivity requirements into mechanisms capabilities, the ones that satisfy the connectivity requirements are selected.

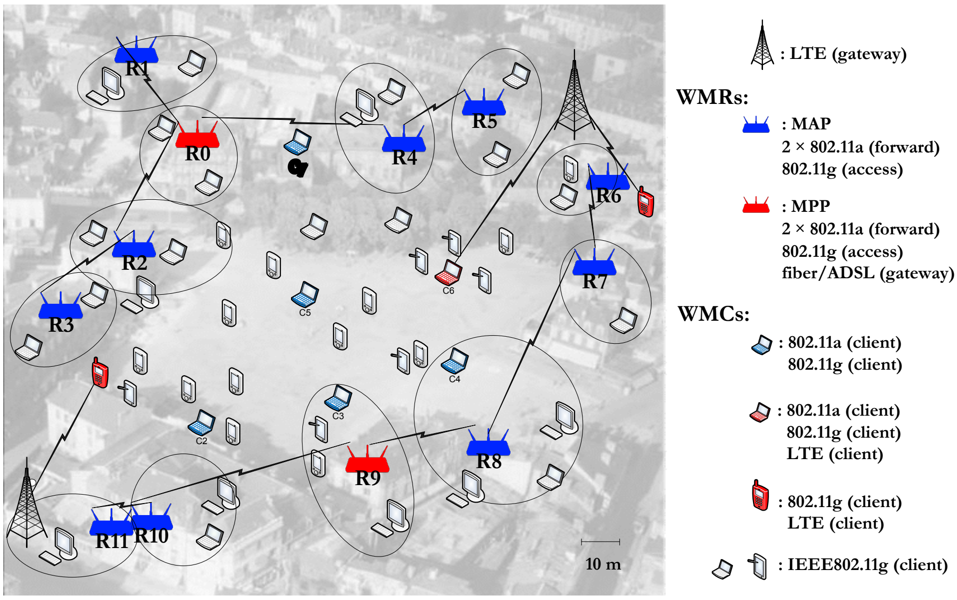
- **Composition and configuration of OConS service**: SOP instantiates the DEs of OConS mechanisms to be created, launched and put in relation to work together. Resources and some mechanisms may need to be configured. The registration of instantiated OConS mechanisms and OConS service in the OR is also needed.

These orchestration steps enable the implementation of adequate OWROS service connectivity mechanisms in any OConS node.

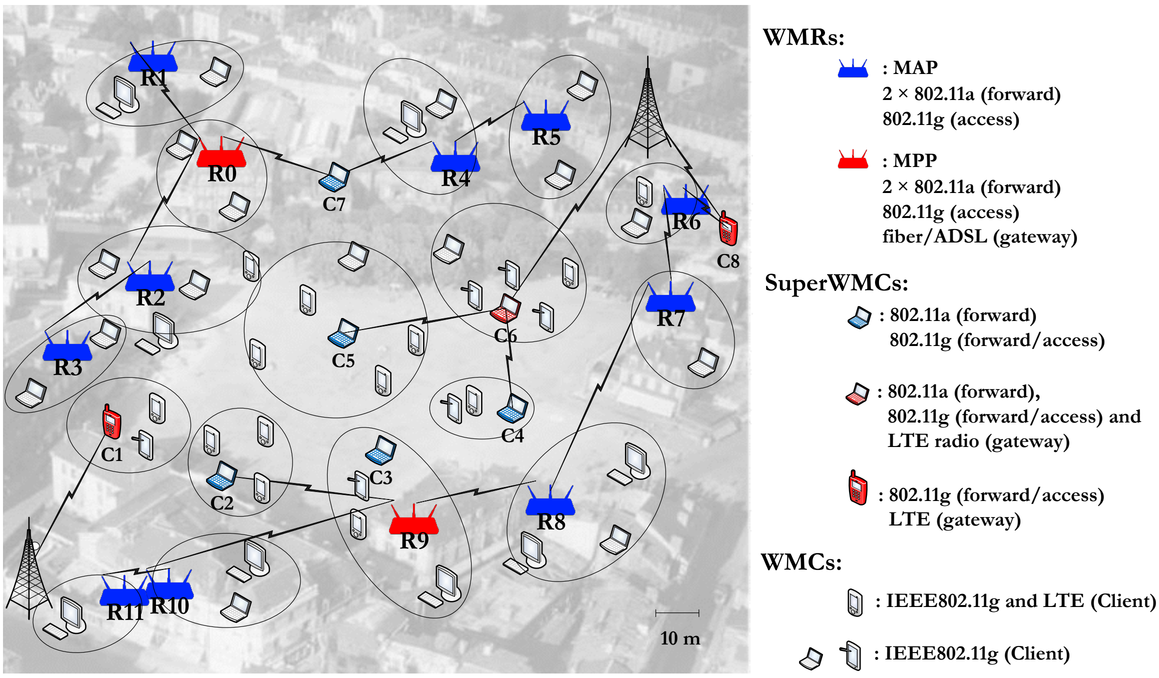
3.3.3.2 OWROS in a Flash Crowd Scenario

For the evaluation of the proposed OWROS service, a residential neighbourhood scenario of 300 x 200 m^2 area is considered with a community-based WMN (Figure 3.8a). Details on the scenario are available in [26]. A spontaneous community-based WMN of 12 randomly-deployed multi-radio WMRs provides connectivity to WMCs, having two WMRs gateways to the Internet. A flash crowd situation is considered, where 28 end-users equipped with heterogeneous multi-radio WMCs congregate, all willing to simultaneously access the Internet.

Without OWROS, this sudden increase of end-users represents more network congestion and less resources per node, thus many WMCs not having Internet access. Thanks to the OWROS service, some WMCs may become SuperWMCs, also providing access, forwarding or gateway functionalities. The offered capacity increases for many WMRs, as several SuperWMCs launch the Internet gateway provisioning mechanism. Other SuperWMCs reduce the ranges between many forwarders enabling forwarding links of higher throughput, thanks to the Mesh forwarding connectivity mechanism. SuperWMCs also provide coverage extension to many WMCs that originally were not covered, thanks to the Client access provisioning mechanism. It is concluded that with OWROS, the flash crowd of heterogeneous WMCs brings large benefits in terms of connectivity, coverage and capacity to the overall WMN performance.



(a) Without the OWROS OConS service many WMCs are not served.



(b) With OWROS OConS service some WMCs become SuperWMCs, enhancing coverage, connectivity and capacity.

Figure 3.8: Spontaneous community-based WMN with a flash crowd of heterogeneous WMCs.

The OPNET Modeler simulation platform [40] is used to evaluate OWROS, having been developed multi-radio nodes with the capability of launching OWROS with a specific set of mechanisms. The community-based WMN scenario is evaluated in terms of coverage and capacity. It is first considered that the WMN operates according to the classical architecture, where the crowd of WMCs gets Internet access via WMRs, a multi-hop wireless backhaul with some gateways to the Internet. Although many WMCs have multiple radio interfaces, the rigid architecture only exploits the 802.11g one. The resulting simulated max-min fair aggregated throughput, R_{fair}, and gateway-to-WMR delay, per WMR, is depicted in Figure 3.9 in grey (without OWROS). This corresponds to the maximum supported aggregated throughput per WMR. The gateway WMRs naturally achieve the maximum throughput, as they are directly connected to the gateway. For the remaining nodes, the announced capacity R_{fair} could be offered in the access network using an 18 Mbps bit rate in the 802.11g. With a range up to 86 m for IEEE 802.11g at 18 Mbps, every WMC of the crowd is within communication range of at least one WMR. Still, the limited number of orthogonal channels (only 3 orthogonal IEEE 802.11g channels) to be used by all WMRs and the associated interference range (2 times the communication distance, for 18 Mbps), do not allow to have the entire capacity of all WMRs used simultaneously, as nodes would interfere. To reduce interference ranges, lower power levels have to be used, reducing also the coverage range of each WMR, a large number of WMCs of the flash crow remaining uncovered, as represented in Figure 3.8a.

The performance of the same scenario is evaluated when the OWROS OConS service is offered to WMCs willing to join the community-based WMN. In this case, besides WMRs, some WMCs opportunistically assume network functionalities thanks to the proposed OWROS service, becoming SuperWMCs, as shown in Figure 3.8b. SuperWMCs capabilities are explored by the service in the most advantageous way. Many provide access and forwarding network functionalities, enabling to provide coverage to all WMCs and increase the offered capacity.

An analysis of various improvements brought by SuperWMCs, depicted in Figure 3.8b, is presented next. SuperWMC C1 has both an LTE and an 802.11g interface. After accepting an invitation to join the community, it is upgraded with OConS-related software, and starts the orchestration process described in Figure 3.8b. It receives the conditions of the network, and the available OWROS OConS service mechanisms. Based on its capabilities, the SOP validates two mechanisms it can assume: gateway and access provisioning, with the 802.11g and LTE interfaces, respectively, accepting to become a SuperWMC and launch the OWROS OConS service, starting to provide Internet access to end-users over the 802.11g interface. As C2 has both 802.11g and 802.11a interfaces, the SOP suggests to launch an OWROS service with access and forwarding mechanisms, providing access to end-users via its 802.11g, while the 802.11a forwards traffic to R9. For C3, SOP does not suggest to become SuperWMC as it is not necessary for the network. For C5, SOP launches and OWROS service, using its LTE connection as a gateway to the Internet, its 802.11g to provide access to uncovered end-users, and its 802.11a interface to forward traffic to C5 and C4. C7 will forward the traffic between R0 and R4, reducing the hop distance, thus, enabling to use a higher bit rate. C8 will provide a nearer and faster access to the Internet to R6 and R7.

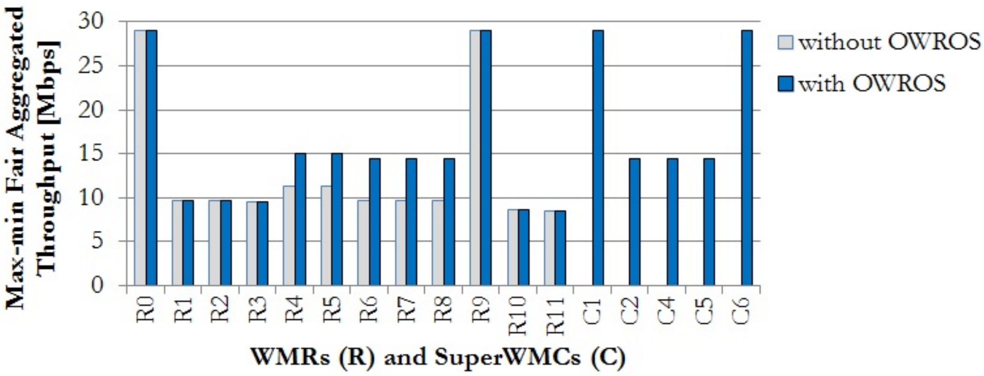


Figure 3.9: Comparison of performance without and with OWROS OConS service.

The simulation results for the WMN with the OWROS OConS service are depicted in Figure 3.9, in blue (with OWROS). It can be seen that offered capacity has increased for many WMRs, as more gateways to the Internet are available and the ranges between many forwarders is shorter, thanks to the SuperWMCs. SuperWMCs also provide coverage extension to many WMCs that originally were not covered. It is concluded that, in contrast to the classical WMN, where more WMCs represent more congestion and reduced resources for all, when the OWROS service is used, a flash crowd of heterogeneous WMCs brings large benefits in terms of connectivity, coverage and capacity to the overall WMN.

4 Assessment of Results

Reaching the end of the SAIL project, this chapter evaluates the OConS results achieved. Clearly, OConS research and development activity is not complete, OConS is not ready for full standardization and commercial deployment. Nevertheless, we were able to significantly advance the state of the art beyond SDN, PCE and OpenFlow. We have developed an impressive set of OConS mechanisms that adhere to OConS architectural design requirements, mechanisms that can be combined to provide OConS services. Such services are considering user and application demands as well as a wider and more appropriate view of the network state, and thus are executed more efficiently and reliably. We were able to demonstrate a proof of concept with some prototyping activity.

This chapter starts with an OConS architecture assessment section. We evaluate our work in the context of an architecture framework for the description of (functional) architectures established within a specific application domain. In contrast to an architecture framework, an architecture defines how a concrete (distributed) system is build in terms of functions (OConS mechanisms) and resources to accomplish a required service. Then in chapter 4.2 we perform a detailed assessment of the OConS mechanisms and OConS benchmarking algorithms conceived so far. In chapter 4.3 we perform a short assessment of the OConS demonstrations and prototypes we have completed, which is further elaborated in D.C.5 deliverable [3].

4.1 OConS Architecture Assessment

Due to the separation of control and forwarding planes, the design of software for the control plane is getting more into focus. The way this software for the control plane is build can be complex. OConS approach to manage this complexity is enabled by means of its OConS components and OConS nodes. Further, OConS facilitates the management of existing mechanisms along with the new OConS mechanisms in a structured manner.

The approach taken by OConS regarding service control is specific and can be considered a basic paradigm: there is a continuous feedback control loop - monitor, make a decision and enforce the decision. Certainly, this paradigm is not a dogma: service control is not hindered to be build without that feedback control loop, if deemed necessary. However, the OConS components are means that support this paradigm, and, at the same time, are models of parts of the service control. OConS components take the form of a kind of standard components that support the design and development of service control solutions. This is different than other approaches, suggesting, for example, programming languages to program networks [41, 42].

One of the potential misconception with OConS is that it is considered an architecture, where in reality, OConS is an architectural framework, according to system and software engineering standards. OConS is in fact an architecture framework, which defines how to organize the structure and views associated with an enterprise architecture. This is fully inline with ISO/IEC/IEEE 42010 that establishes common terminology for architecture frameworks, and specifies requirements for standardization of frameworks.

An architecture framework is defined here as: conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders. In the contrary, an architecture defines how a concrete (distributed) system is build in terms of functions used and how resources are used, to result in the desired services that are aimed to be achieved.

Such conceptual difference concerning architectural frameworks and architectures is well known in software engineering, but it also exists in the networking realm: "One can think of frameworks and architectures as being akin to microkernels and kernels, with the former being a simple and minimal design that enables greater flexibility than the latter" [43]. In OConS the architectural framework is concerned about the OConS components, OConS nodes and how these can in principle be distributed and interact to support the orchestration of mechanisms. Whereas an OConS architecture is defining how the mechanisms are orchestrated on the service control plane in a concrete manner, such that the forwarding plane is properly controlled by the orchestrated mechanisms. As such the OConS approach contributes to the way service control can be appropriately designed, considering the separation of control and forwarding planes and taking on board some software engineering concepts.

The discussion that follows now builds on this definition of architectural framework and architecture and validates the achieved results of our work in this respect. The architectural framework was not the main topic but rather the necessary task in the research reported here. Consequently, one may say that we have followed a "good enough" approach for building the architectural framework. This is an approach sometimes used on agile software development with limited resources, when achieving results quickly is more important than achieving generality.

In section 4.1.2 we validate the OConS architecture framework. We then consider a few possible implementation examples of OConS architectures, conceptually validating them in section 4.1.3. In section 4.1.4 we provide a comparison of an OConS architecture approach with other architectures that are currently under consideration for the Future Internet. Finally, we self assess our architectural results.

4.1.1 Architectural Validation and Criteria

Because of the different concerns of architectural frameworks and architectures, we first have to identify relevant validation criteria. These criteria will be different from those that typically help to quantify the engineering result (such as mechanisms used in a transport infrastructure that would address properties of performance, feasibility or scalability, for example). The criteria used here are concerned with questions such as: *How well the architecture is describing the solution? Is it useful in reasoning about the orchestration of mechanisms? Can ideas be communicated clearly? Is it open for evolutions?*. And, of course, we want to know how specific OConS architectures look like, as compared with other architectural concepts (Section 4.1.4). Therefore, these validation criteria are of relevance for the design, development and software engineering of service offerings when based on OConS.

The assessment discussion of Chapters 2 and Chapter 3 are different. The former is addressing the quality of results in concrete OConS architectures, when orchestrating particular combinations of mechanisms. Then, Chapter 3 is concerned about the quality of achieved support, when OConS is used by technical systems that make use of specific OConS architectures when orchestrating particular combinations of mechanisms.

4.1.2 Validation of OConS Architectural Framework Properties

The following discussion is concerned about the support of the OConS Architectural Framework to derive specific OConS architectures and to build and implement concrete, orchestrated combinations of mechanisms. We do not make any assumptions regarding the existence of such mechanisms.

Flexibility i.e. the ability to replace the design of orchestrations or (combinations of) mechanisms, in specific OConS architectures.

The OConS Orchestration is discussed in depth in [2, Section 3.7], elaborating on the properties of SOPs being used according to the OConS architectural framework. This covers particularly the OConS Service Orchestration at runtime, where a SOP considers the connectivity requirements and demands, as well as the network state changes, and it selects, according to a defined strategy, the appropriate OConS Services provided by the available mechanisms. The strategies that can be defined for the mechanisms selection allow to handle the trade off between elaborate mechanisms selection that can encompass the ever changing network state, and, more simpler (and faster) selection of mechanisms that are well known to the designer of an OConS architecture.

Thus, the SOP is an architectural invariant to all OConS architectures and it is not foreseen to be exchanged. Note that the four options discussed as the strategies for how to select the mechanisms (see [2, Section 3.7.4]), is another concept of the OConS architectural framework; likewise, these options may be more easily enhanced with further options. The SOP, however, remains as the continuous feedback control loop to monitor the changes, it makes decisions and then it sends the appropriate messages to the enforcement components.

Moreover, by using the mechanism manifest defined in [2, Section 3.2.2], we have a common way to represent each mechanism characteristics (what it provides, what are the needed resources, what are its constraints), and thus the SOP functionality can more easily incorporate (select, combine, instantiate) new mechanisms.

Complexity i.e. the additional complexity introduced by OConS for the design of solutions (expressed in specific OConS architectures), namely (i) the concepts and the functional entities that are used for wrapping access to the (existing) mechanisms [2, Section 3.2.1] and (ii) the four options for how to select mechanisms and adapt to changes in the runtime environment [2, Section 3.7.4].

Thus, the three generic functional entities of types IE, DE and EE should be considered as homogenization of parts of the service control. Almost all mechanisms encompass ways to gather information, based on which decisions are taken and eventually enforced. Here, the generic functional entities make the complexity more manageable, lifting details of (existing) mechanisms on a level where they become comparable, and they allow to reason about these mechanisms on a comparable abstraction level.

As discussed above, the flexibility is not introduced by the SOP, but rather by those strategies applied for mechanisms selection. Whereas the SOP, as architectural invariant, makes complexity more manageable and handy, the mechanism selection options increase the complexity within each OConS architecture. In turn, however, this approach allows more flexibility to adapt the mechanisms for service control at runtime, which can be beneficial for more optimal use of network resources. The obvious trade off here can certainly be debated, i.e., to see to what extent it is worthwhile, and, thus, we reckon that more research is required.

Extensibility, versatility and evolvability i.e. the ability of OConS architectural frameworks to accommodate additional concepts or views that affect the way specific OConS architectures can be build.

This property would allow to extend the architectural framework we have defined with further architectural concepts. As a result, the OConS architecture that can be build would express (in addition) other aspects. Since we have taken a “good enough” approach to define the OConS architectural framework, its extensibility can not be thoroughly assessed at this point in time, as there are no proofs to say that this is achievable, and under what circumstances. However, we have made this effort being convinced that most of the service control details can be expressed, although we do not have a formal proof for such a statement.

Sustainability i.e. the maintainability of specific OConS architectures that continue to be used over time.

We have investigated and specified the OConS Orchestration functionalities needed when combining several mechanisms in order to provide the optimal OConS service. 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 fully specification of a complete orchestration process that is sufficiently flexible, scalable and powerful to encompass current and new networking mechanisms. As this is an open research challenge, the OConS architectural framework, as well as OConS architectures, will change in the future, though the main underlying principles will remain.

Costs i.e. the costs at design and development time, introduced by making use of the OConS concepts, namely the OConS components, nodes and orchestration in order to derive a specific OConS architecture.

Without any prejudgement that the described OConS approach is enabling designers and network engineers to use the adequate concepts, it is clear that, due to the separation of control and forwarding, the design of software for the control plane is getting more and more into focus. In our approach we have abstracted the functionality that we would like to take for granted in networks, i.e., the basic components and entities, the OConS nodes and the SOP. From our point of view, they are presenting suitable models, though we also know that further research is needed, especially to show that the level of abstraction, the functionality modelled and the inherent process to design service control, are all cost efficient.

The above is not considered complete but sufficient for a discussion here. Since the development of the OConS Architectural Framework was not a central research topic it can be assumed that further studies are necessary.

4.1.3 Validation of OConS Architecture Properties

The following discussion addresses properties and capabilities that are considered to hold for each and every OConS architecture, understood as functional system architecture, and addressing the generalization over all (assumed) specific OConS architectures with their orchestrated mechanisms.

SW complexity i.e. the additional effort introduced by the architectural components each and every OConS architecture is made of.

The functional entities of types IE, DE and EE and the SOP, as concepts introduced into each specific OConS architecture have been discussed above. For functional systems architectures, these entities may appear in a multiplicity depending on the case-specific demand that comes with the number of mechanisms to be orchestrated, and these specific mechanisms. E.g., to achieve the orchestration, there will be at minimum two mechanisms (and at least one instance of type DE).

However, there may be OConS architectures that orchestrate more than two mechanisms and there will certainly be mechanisms encompassing more than one DE only. The concrete number of IE, DE or EE instances is case-specific. The advantage of this approach is a homogenization of control over (existing) mechanisms, and, in addition, in some cases the advantage to share information, i.e., when different mechanisms may benefit from commonly used information via shared IEs.

Achieving all that is to add into the control space the OConS Nodes and the capability to allow the communication between OConS nodes via Intra-/Inter- Node Communication (INC).

Thus, this one-time investment establishes a homogeneous environment within which the orchestration can build on the supporting functionalities that can be taken for granted in each OConS functional architecture.

Feasibility i.e. can a designed solution, which is expressed in a specific OConS architecture, be implemented in software running eventually within an infrastructure?

Naturally, it is not possible to make an assertion concerning this property for all possible OConS architectures. However, a proof of concept has been built in the project and this is extensively discussed in the deliverable D.C.2 [2, Section 7.2] to which the reader is referred to. That discussion focuses on the SOP and the INC and further addresses the orchestration encompassing the *Enhanced Access Selection* and the *Dynamic Distributed Mobility Management* mechanisms.

Also the reader is referred to D.C.5 [3], where the realization of prototype components and the demonstrator activities for OConS are reported. This also comprises prototype realizations of OConS architectural concepts such as orchestration or abstract IEs, DEs and DEs for several mechanisms.

Adaptability and self configuration i.e. the ability to adjust the orchestration of mechanisms at runtime according to monitored network characteristics and available capabilities of a (potentially changing) set of registered mechanisms.

Certainly, this capability, which is envisaged and depends on the registered mechanisms, supports in each OConS architecture the adaptability at runtime. It is described in details in deliverable D.C.2 [2, Section 3.7.4]. Yet, to depict the trade off between the efficiency and the complexity, four complementary ways for selecting the appropriate mechanisms are foreseen. These differ in their complexity and the computational effort required to make those decisions. The simplest form is a selection based on profiles that are identified and known at design time. The knowledge about the identified profiles is used at runtime when the selection of mechanisms respects the actual demand profile and the network state considerations, as the selection space.

The ability to adapt and improve the configuration of mechanisms can be further enhanced to enable different types of mechanism selection. E.g., this may encompass hierarchical structure where part of a selection mechanism could be changed during runtime. It may also comprise more complex forms. However, such enhanced capabilities will always come with the cost (e.g. simplicity), which may be a desirable architectural property to be kept.

Scalability i.e. whether orchestrated mechanisms expressed by means of an OConS architecture can scale, despite the OConS specific concepts like the functional entities, the SOP, the OConS Nodes and the INC.

There is certainly a limit given by natural resources that will describe the scalability boundaries of each and every OConS functional architecture. Experiments and tests to identify these limits have not been made; nonetheless, with the scalability in mind, we have suggested the use of a distributed and hierarchical orchestration, which needs nonetheless further investigations.

Such limitation will, however, also depend on aspects that are not part of OConS. Should for example virtualization mechanisms be used as runtime environment for the control plane that OConS Nodes will be implemented in, then physical limitation have minor impacts, if any. Further research will be fruitful, e.g., see here (e.g., see [44]).

Performance i.e. the performance talking into account the additional effort due to the wrapping of OConS components, the computational effort required for orchestration, interaction across

OConS nodes (if any), the network characteristics monitoring and adaptations (if any), and finally the OR registration procedures.

In general the amount of additional computation introduced for each OConS architecture is limited. The different phases the design, the deployment and the runtime execution of orchestrated mechanisms will require is discussed in details in Deliverable D.C.2 [2, Section 3.7]. Furthermore, a number of examples of orchestrations have been looked at also in prototyping activities (Deliverable D.C.5 [3]). General pitfalls were not identified, though they may exist in individual solutions that can be build according to the OConS architectural framework.

Costs i.e. the costs (at design time, development time and runtime), introduced by using an OConS architecture of orchestrated mechanisms, as compared with an approach that is not structured by means of an OConS architecture.

Many factors will actually determine the costs. Some of them have been addressed in the discussion above. Costs that are directly related to the OConS approach and that materialise at design time, are introduced by the OConS architectural framework, and are discussed above. Generally, each OConS architecture will have to cope with the additional costs to address and encompass software components that do the wrapping and the handling of IE, DE and EE, create the OR and the OConS Nodes. In addition, there will also be additional costs that are related to the implementation of the SOP. However, the architectural concepts are designed in a way that they are widely applicable, such that an economical approach is possible.

The main risk that eventually will be a potential cost factor is related to testing. It is assumed that each and every software module has to pass rigorous testing before it is released. It is obvious that testing is required and vendors will have to show if and how validation defined criteria can be shown and will be proven during testing. The fact that an orchestration process copes with adaptability and adjustments according to changing demand or network state will make the testing a challenging task, but creates a larger entropy to be covered in testing. However, one will not be hindered to design and create OConS architectures for orchestrated mechanisms that avoid unnecessary complexity.

There are additional criteria items that we can not use for validation, as it would require specific OConS architectures to look at. These criteria can, in fact be relevant in future assessments, such as: manageability, operability, sustainability, omni-accessibility, traceability, debugability and dependability (understood as the combination of security and reliability). When the separation of control and forwarding planes will be deployed, the main challenge would be to ensure reliable operation and system integrity; these may be the most critical topics for operators.

4.1.4 OConS Architecture Approach in Comparison

The following is a comparison of an OConS architecture approach with other architectures that are currently under consideration for the Future Internet. This comparison is meant to clarify the contribution of our OConS approach. Table 4.1 is inspired from [16] and relates some of the OConS architecture validation criteria with a concrete OConS architecture, i.e. the OConS architecture containing the DDC-WIM, whose results are briefly presented in 2.3.1, and evaluated in 4.2.2. As such it also provides a different view on the relevant OConS architectural concepts and other architectures.

Both the SDN and the OpenFlow approaches facilitate a flexible control of a network domain, e.g. the implementation of a new protocol and its integration into existing legacy networks. Both approaches manages the collection of the complete network topology and the network state concerning connectivity resources. Other resources, however, such as the state of processing resources

Architecture concept	**IP**	**PCE**	**OpenFlow**	**OConS**
Overview	Integrated control and data plane in one node type, fully distributed control algorithms	Separated control and data plane, centralized control algorithms	Separated control and forwarding nodes, centralized control algorithms	Separated control and forwarding nodes, centralized and decentralized control algorithms possible
Multi-layer capability	No	Yes	Yes	Yes, in a structured manner
Multi-domain capability	Yes, by exterior routing protocols (e.g. BGP-4)	Yes, algorithm Per Domain Path Computation (PDPC) and Backward Recursive PCE basedComputation (BRPC) via PCE protocol (PCEP)	No	Yes, by means of enhanced mechanisms and combinations of mechanisms
Reliability	Self healing, that's what it was introduced for	Single point of failure is the PCE	Single point of failure is the OpenFlow controller	Several OConS mechanisms running on different OConS nodes can be combined to ensure reliability
Market status	Widely deployed in current Internet	Field trials and early deployment	The most popular SDN implementation, primarily used of experimentation of new protocols in real networks	Proof of concept and initial research results

Table 4.1: Architecture Comparison Table

or storage usage, are not available from the OpenFlow protocol only. The OpenFlow approach provides resource feedback from the forwarding and switching resources only. Within OConS, however, a new mechanism can define new measurements - triggered on demand or event based - that enable new functionality that is beyond forwarding or switching capabilities. This wider scope enables the OConS orchestration to obey and enforce longer-term strategies.

In addition, the modular OConS architecture supports the combination of different mechanisms. This means that e.g. OpenFlow switch controllers, flexible PCE-like path computation algorithms and operator policy mechanisms, can all be combined to form a new OConS service that is otherwise not possible today. These differences are expressed in the rows of Multi-domain Capability and Reliability in Table 4.1. OConS enhanced properties and open interfaces leverage from IP, PCE and OpenFlow advancements, and demonstrate an advanced and powerful approach for managing network connectivity in future networks.

4.1.5 Self-Assessment of Result on Architectural Work

When the OConS research work started, the basic idea for improving connectivity services was to follow the simple *monitor – make decisions – enforce* strategy, aiming at providing flexibility and control beyond today's solutions. During our work, it became evident that the concepts followed can be generalized to become more widely applicable. This observation, combined with the separation of control and forwarding concerns, led our work to the architectural results discussed here. Keeping the different concepts apart can already be a challenge, i.e. viable definitions of notions for functional architecture, architectural framework, etc. The use of ISO/IEC/IEEE 42010 was helpful, but there might be other, possibly better foundations. We basically followed a "good enough" architectural framework approach, as elaborated in Section 4.1 , which indicates that more research is needed.

It turned out that we have introduced concepts that apply to each and every OConS architecture. It was only consequent to gather these concepts (e.g., OConS functional entities and SOP) in one architectural framework. This actually directed our work into fields typically governed by software engineering, with is only a natural consequence of opening the control plane for more software-centric solutions. This direction is by no means fully covered. The space into which future solutions can be designed has been clearly opened. We have shown preliminary feasibility analysis, as well as the additional complexity that has to be addressed at one point.

We showed in practical experiences that it is feasible to orchestrate new and existing mechanisms. However, we do not know what would take precisely to have orchestrate-able mechanisms or mechanisms that are not amenable to orchestration. This would show the limits of our approach concerning the combination of mechanisms.

Despite OConS clear benefits, orchestration is not necessarily making things less complex. The controlled scope can be reduced by means of virtualized environments, hiding physical resource availability and allocation from OConS. However, we do not have suitable models that make it easier to handle service control design and development; models that, by means of further virtualization, allow to safely hide technical details. Clearly, there is more research needed in order to better understand and mature our solutions.

4.2 OConS Mechanisms Assessment

In this section we assess the OConS mechanisms developed. We start with an overview, discussing how the evaluation of OConS mechanisms is performed in general. We then look specifically into each OConS mechanism and benchmarking algorithm we have conceived so far, providing self-assessment, lesson learned, and possible future work.

4.2.1 Overview

We have conducted extensive evaluations, aiming at validating and providing proof of concept for our innovations and enhancements. Accordingly, the evaluation scope spanned the specific OConS mechanisms, as well as a combination of some of the OConS mechanisms within the OConS architectural framework; furthermore, several of those mechanisms were validated throughout the SAIL-wide Flash Crowd Scenario, and most notably, the OConS prototypes and experimentation in support of CloNe and NetInf use-cases. We have developed an impressive set of OConS mechanisms, each one is addressing a specific networking problem or deficiency, enhancing a specific connectivity aspect with respect to the current state of the art.

All of those mechanisms were described in deliverable D.C.2 [2]. Accordingly, most of them have been evaluated in Chapter 2, thus presenting their performance and benefits. A few of them are still under development, did not reach final results and are therefore not reported in Ch.2.

Generally speaking, the evaluation of the OConS mechanisms was done according to the following methods:

- Simulation and/or lab test bed of the network environment and the mechanism. This task is considered as a unit test and validation, in which each mechanism demonstrates its per-

formance in an isolated and simulated manner. In some cases, real data was used for the simulation. To some extent, all completed OConS mechanisms were at least evaluated in this method. 7 OConS mechanisms were evaluated according this method only.
- Theoretical evaluation of selected mechanisms: linear programming and game theory, facilitating complex scenarios comprising several base stations, hundreds of nodes and heterogeneous networks, with a variety of services.
- Implementation of a mechanism on a real platform (such as an Android cell phone). Clearly there is less to simulate and more real aspects in this method of evaluation.
- Evaluation of one or more mechanisms in OConS simulated framework environment. In this method the applicability and the feasibility of the mechanism was studied in the context of selected features of the OConS architecture, thereby showing how the mechanism is performing in OConS architecture. In selected cases, a few OConS mechanisms that form an OConS service were involved, also showing some functionality of the orchestration, and the effect of simultaneous use of mechanisms. in other cases, OConS mechanisms were also combined with legacy mechanisms, evidencing a feasible migration. 8 mechanisms were evaluated following this method
- Evaluation of OConS mechanisms and selected features of the OConS architecture over commercially-available routers and switches. Two mechanisms where integrated with bootstrapping, discovery, signalling and orchestration functionality of the OConS architecture, all running on routers and switches.
- In addition to the specific performance evaluation that is appropriate for each mechanism, in some cases additional effects of the mechanisms were studied, including response time, delay and cost, also evidencing scalability capabilities.

In addition to the OConS mechanisms, we have also developed a few benchmarking algorithms, which were also reported in D.C.2 [2], and evaluated in chapter 2. 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. Clearly, the evaluation of these benchmarking algorithms was limited to the first method only, since they are not meant to run on the real time OConS environment.

4.2.2 OConS Mechanisms Self-Assessment

Access Selection and Decision (Section 2.2.1)

The enhanced access selection mechanism is fully integrated within the OConS architecture. In addition, it is a mechanisms which can be used in almost every use case or scenario, since its functionality is rather generic. We have assessed its feasibility by means of a prototype [4], in which we exploited the signalling capabilities between OConS entities; furthermore, we also orchestrated with another mechanism. In this sense, it is worth mentioning that this mechanism can be smartly combined (orchestrated) with many others (multi-path, different mobility protocols, etc). Since the prototype evaluation was limited (in terms of the number of networks, technologies and users), we conducted an in-depth study, using a event-driven simulator [5]; the results therein showed the relevance of the decision strategy, especially considering the impact of not penalizing handover events. All in all, results achieved thus far suggest that the Enhanced Access Selection Mechanism has proved to be very suited to the OConS architecture, which also provides the required flexibility to extend it so as to consider different parameters within the decision process. Going forward, we will foster the flexibility of the mCASE tool in order to conduct more analysis.

Handover with forward admission control for adaptive TCP streaming in LTE advanced with small cells (Section 2.2.2)

Our handover schemes seek to maintain the TCP throughput while minimizing the traffic exchange between BSs. We examined three handover algorithms and analyzed their performance. The first algorithm drops all the packets received by the old BS after the user has moved to a new cell. The second algorithm forwards all these packets to the new BS. The third algorithm uses an elastic forwarding buffer and makes an on-line decision which packets to drop and which to forward. We showed that the best performance is obtained by the third algorithm, and that this algorithm not only maximizes the total throughput but also minimizes its variance during handover, thus improving users' experience of streaming applications. However, the third algorithm is more complicated than the first two, because it requires the BS to estimate the TCP congestion window size (cwnd) for each connection. Therefore, we believe that more work is needed in order to find a scheme that obtains the performance gain of the third algorithm while keeping the simplicity of the first two algorithms.

Multi-P Decision and Transmission for the Interconnection of 3GPP and non-3GPP Systems (Section 2.2.3)

The proposed mechanism advocates the support of multihoming for QoE enhancement and overall network performance optimization. The concepts, mechanisms, and system architecture for network resource management presented in this work serve as a basis for further research in the area of user multihoming and heterogenous networks. The current work has focussed on radio interfaces of LTE and WLAN. A natural extension would be to perform the resource management of the transport / core network and radio access networks simultaneously. This will provide the resource management scheme with an overall picture of the access network so that performance of both networks is optimized. For example, in such a scenario it will be possible to determine if the transport network can support the user data rates which are being allocated at the radio interfaces during the resource management process. Similarly, it will be possible to determine the suitable transport links for certain application types based on the dynamic QoS characteristics of these transport links.

Though the simulation results provided the proof of concept for integrated networks of LTE and WLAN, the presented concepts should be equally valid for other access technologies like HSPA,WiMAX etc. However, such a validation is another future work item. Moreover, this work focuses on network resource management controlled by operators, while a quantization of achievable benefits when users manage their own bandwidth resources, remains an open work item.

Network Coding Layer within the OConS framework (Section 2.2.4)

Using a network simulator, we studied the performance gain which might be obtained by the use of *Network Coding* over Wireless Mesh Networks, for TCP connections. The complete mechanism was designed following the OConS architecture [10]. Despite the expected potential of this coding scheme, one of the most relevant results was the fact that its performance was severely degraded by the presence of errors at the wireless links [38]. Another aspect which is left for future work is the analysis of more complex network topologies, since we used simple ones, in which it was quite straightforward to establish the node(s) which needed to code the packets, and the combination of flows which was required. It is worth highlighting that one of the papers which were presented as a result of this study [10] received the best student paper award at the MONAMI'12 conference.

Mobile-driven QoE-aware Multihomed Flow Management (Section 2.2.5)

The proposed mechanism is used to solve the Multihomed flow management problem for the end users. The evaluation results have shown that this mechanism can support better QoE for all applications, while limiting the access costs and battery consumption in comparison to legacy techniques of only choosing one best network at a time. Further, the mechanism can enhance the overall network performances by avoiding network congestions and utilizing all available networks in a balanced manner. At the moment, the mechanism is implemented based on the linear programming (LP) formulation of the user-centric solution. Future work will investigate other potential optimization techniques and even heuristic approaches, which can be more feasible for implementation at the end devices. So far, we have compared the mechanism against one flow management technique of only choosing one best network at a time. In the future, we will also compare our approach to other multi-homing techniques in order to get better insight about the respective costs. In addition, in order to make proper decisions, our mechanism relies on information (*e.g.*, link QoS such as capacity, delay, loss) collected from the information elements of OConS. It is a challenge for the network operators as well as for the end-users to collect these information and share/distribute them properly in the network.

Price based load balancing for wireless access networks (Section 2.2.6)

We studied a distributed load balancing scheme which might be used by the BS to manage their resources in an optimal way. Their current load was used so as to establish the price they were offering to the end-users (who would also use such parameter within the corresponding decision process, orchestration between the two mechanisms). The results obtained [15] showed that pricing policies can be used so as to encourage or deter users from connecting to a particular BS in order to achieve load balancing. Furthermore, the study shows that under certain considerations, the operator revenue is not affected by this kind of pricing while end users experience a smaller number of handovers, thus improving their QoS.

Distributed Datacentre WAN Interconnectivity Mechanism (Section 2.3.1)

While today's networks provide a variety of interrelated services including definition of paths, traffic monitoring, load balancing and more, they do not easily support the setup and control procedures of such services. Unfortunately, the network programming possibilities and languages used are low level, complicated, error-prone and as such, difficult to handle. E.g. the much hyped OpenFlow concept is defined at a very low level of abstraction and directly targeted at the capabilities of the involved switch hardware. Moreover, with OpenFlow it difficult to accomplish sophisticated services beyond setting up a plain path, such as process placement, monitoring and flexible reaction upon changes in the network. This is why the OConS DDC-WIM mechanism was conceived 2.3.1. This OConS mechanism eases dynamic resource allocation for virtual resources including not only the calculation of the connectivity resources, but also the selection of the computing resources needed to fulfill a requested service. Furthermore, in order to maintain the requested service, the related resources are monitored and the allocations are renewed in case resource congestion is detected. Consequently, this OConS mechanism achieves datacentre elasticity over a wide area network.

Address Resolution for Distributed Datacentres (Section 2.3.2)

The OConS mechanism "DDC-ARM", described in section 2.3.2, manages the control traffic caused by address resolution between interconnected datacentres. We have shown that DDC-ARM considerably improves the scalability of the link layer ARP. Moreover, DDC-ARM also supports the datacentre mobility and migration of processes hosted on VMs across distributed datacentres connected via WAN. By introducing an additional architectural entity at the OConS level, the ARP proxy, our evaluation shows that this is an effective method to reduce the ARP (control) traffic caused by the necessary address resolution procedure between those interconnected datacentres. DDC-ARM is based on the basic mechanism DDC-WIM whose results are reported in section 2.3.1. The DDC-ARM extends the ususal ARP procedure and bridges distributed LAN islands across WANs based on OConS domains. DDC-ARM prevents flooding of the interconnecting WANs by introducing an intelligent address translation. Thus it improves the scalability of distributed datacentres.

Integration between OConS and CloNe (Section 2.3.3)

The integration between OConS and CloNe implemented in SAIL provides a proof-of-concept implementation for multi-domain SDNs. This topic is arising both in the Open Networking Foundation (ONF) and in the Network Function Virtualisation (NfV) study group at ETSI. The current implementation allows basic path stitching based on OpenFlow. The main future work is to understand the challenges when coordinating SDN controllers that implement QoE-based connectivity services. There needs to be a clean separation between the functions that OConS orchestration can deliver and those implemented using the DCP defined in CloNe. Additionally, duplicated functions in the implementation of the ISI in the pyOCNI code and the OConS orchestration need to be identified and rationalised.

Dynamic Distributed Mobility Management (Section 2.3.4)

Several approaches exist to distribute the IP-level mobility execution functions in a flatter way, notably within the access routers (e.g., see Fast Proxy Mobile IP in [45]). However, in these approaches, a centralized anchoring functional entity is needed, as well as encapsulation/de-capsulation operations for all the flows, regardless of the fact that the terminal is (or not) on the move from "home network". We have thus developed both a distributed and a dynamic mobility scheme, i.e. the mobility execution functions can be distributed at the access routers level, while, at the same time, the anchoring and the encapsulation/de-capsulation operations are done only for active traffic flows during the handovers. Likewise, the evaluations carried out with the OPNET simulator have validated our approach. As further improvements, we think that a proactive mobility execution can be also beneficial, especially when coupled with a selection and handover decision mechanism that is able to provides proactive decisions.

Network Coding for M-to-N Routing in DTNs (Section 2.3.5)

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 Annex A.2 suggest that the history of contacts could enhance the delivery ratio of multicast transmissions in challenged environments. In WMN or DTN topologies, where a local SOP can instantiate and combine several available OConS mechanisms, the combination of OConS-DTN routing with NC based on social routines would result in an enhanced performance and increased packet delivery ratio in such challenged environments.

CQI channel allocation in OFDMA networks (Section 2.4.1)

While the proposed scheme considerably increases the system's benefit from the allocated CQI bandwidth, one of the most important conclusions we draw from this research is that this benefit depends to a large extent on how we determine the profit function for each Mobile Subscriber (MS). After we tested several possible functions, we chose one that requires the BS to estimate how dynamic the downlink channel of each MS is. This estimation is translated into the average time window - w, during which the Channel State Information (CSI) value of each MS changes. The BS also calculates the average data packet rate - r for the MS , and sets E=w*r. Consequently, E represents the average number of consecutive packets transmitted to the MS using a correct CQI value. We believe that other functions can be found, some of them might better suit different cellular environments.

Dynamic Radio Resource Allocation for Virtual Connectivity (Section 2.4.2)

The allocation of radio resources to sharing operators based on RRM is already in place for Mobile Virtual Network Operators, e.g., for 3G Networks. Furthermore, RAN Sharing is also standardized for LTE networks, allowing to share the infrastructure as passive sharing, active sharing and geographical sharing [46]. Although several models for radio resource sharing have been proposed, the allocation of resources, e.g., spectrum or power, is more or less fixed and cannot be dynamically adapted to the network state. This may lead to an inefficient use of the available resources denying service to end-users. The novelty of the proposed VNet Radio Resource allocation mechanism is to address wireless virtualisation by sharing a set of physical resources from different RATs through the adaptive allocation of radio resources to virtual resources. Another relevant enhancement of the mechanism is to enable differentiation among several types of virtual resources, ensuring the amount of capacity (data rate) requested for guaranteed virtual access to networking services. In the scope of OConS two algorithms to implement the VRRA mechanism have been proposed, Section 4.4.1 of [2], and evaluated, Section 2.4.2. A first one is based on the pre-allocation of radio resources to each virtual resource as a function of the requested capacity. The second approach presented is an evolution of the first one, though the approach is slightly more flexible, since it considers only the allocation of radio resources to the virtual resources when end-users are requesting service. Both algorithms have been modelled under OConS and evaluated for different scenarios [19] and [20]. From the results, it can be said they can support the minimum bandwidth requirement for virtual access in a wireless cluster, composed of several physical BS from different RATs, providing isolation among virtual resources. The implementation within OConS takes advantage of OConS flexible approach, e.g., concerning the activation and (re)configuration during network operation as well as the utilisation of the integrated inter-node communication capabilities.

Cognitive Radio Systems through Spectrum Sensing Techniques (Section 2.4.3)

The work developed within OConS in the Cognitive Radio field has been focused on analysing and simulating cooperative spectrum sensing techniques following the OConS architecture. Experiments have been conducted towards the performance analysis of soft decision techniques that support the access selection decision in wireless environments. This mechanism could be applied in any of the use cases depicted, where several wireless nodes need to be assigned a radio channel and attachment. We have assessed its feasibility by means of a validation scenario through simulations (section 2.4.3), in which several wireless nodes associate their 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 individual nodes. This mechanism is specially tailored to be combined with OConS enhanced access selection (see 2.2.1, which may result in very specific channel allocation and sensing information based selection criteria. All in all, the Enhanced Access Selection Mechanism has proved to be very suited to the OConS architecture, which also provides the required flexibility to extend it so as to consider different parameters within the decision process.

Radio Resource Management Mechanism for Multi-Radio Wireless Mesh Networks (Section 2.4.4)

A mechanism for radio resource management of multi-radio WMNs connectivity has been proposed, its results are presented in Section 2.4.4. It implements the unified management of tightly interdependent radio resources, such as channels, bit rates and transmission power levels of multi-radio MAPs. It has been implemented according to the OConS mechanism-level architecture [22], compared with other strategies [23], evaluated for different types of scenarios such as hexagonal [24] and random deployments [25], with single or multiple gateways. It is concluded that the mechanism efficiently guarantees max-min fair capacity to every aggregating mesh node, overcoming the classical limitations of WMNs associated to its multi-hop environment and typical fat-tree traffic flow. The performance of WMNs is bounded by the system's characteristics. Possible improvements (MIMO, higher modulations and receiver sensitivities) result in higher system rates and ranges, enabling higher theoretical WMN capacity levels that can be reached using the proposed RRM mechanism.

This mesh forwarding connectivity mechanism has been combined with two other mechanisms, an end-users access provisioning and an Internet gateway connectivity mechanism, to build an OConS service, comprising and opportunistic WMN resource management. It is able to orchestrate, in nodes with adequate capabilities (mesh routers but also classical mesh client terminals) and networking functionalities. Section 3.3.3 shows how this OConS service, when offered by a community-based WMN, can be orchestrated in nodes willing to join and cooperated in the community WMN. It is evaluated how the presence of an end-users flash crowd can be beneficial in the improvement of coverage, capacity, and connectivity, when the proposed OConS service is offered by the community to joining members. As future work, the combination of this mechanism with even more refined mechanisms is envisaged, providing advanced OConS services that explore all capabilities of the networking elements as well as environment and application characteristics.

Resource Management within Heterogeneous Access Networks Benchmarking (Section 2.5.1)

When analyzing the goodness of the different mechanisms (in particular concerning the access selection and resource management in heterogeneous access environments), it is interesting to be able to have an optimum performance so as to establish the performance bound. In that sense, we used game theory techniques so as to study resource allocation and pricing strategies to be used by the access elements. The results were published in [27] and [28], respectively. However, the fact that we modeled the resources as discrete units, complicated the corresponding problem formulation, making it unfeasible for more complex scenarios. If the resources were modeled as continuous unit, we could probably use this approach to analyze environments with a larger number of technologies, access elements, and users. Another approach which will be fostered in the future is the use of other mathematical techniques, such as linear programming, extending the approach which we presented in [47].

Multi-Path Benchmarking: the Trade-off Between Control Plane Load and Data Plane Efficiency (Section 2.5.2)

Our work aims at minimizing the control load for delivering a given amount of bandwidth between pairs of ingress and egress nodes, while studying the trade-off between minimizing the cost of a network flow and minimizing the control load. We defined two optimization problems: the Decomposition with Minimum Control Load problem (DMCL), and the Routing with Minimum

Control Load problem (RMCL). We measured the control load by counting either the number of paths that carry the bandwidth or by the number of nodes they traverse. Both problems are NP hard for both control load measures. However, we presented approximation algorithms for all four problem variants. Furthermore, we presented efficient practical algorithms for RMCL. These algorithms first find an initial network flow and then decompose it while trying to minimize the number of paths or the number of nodes. The procedure for selecting the initial network flow was shown to have a critical impact on the performance of the algorithm. In this work we focused on the online version of the problem, where traffic flows are admitted one by one. While there are scenarios where the operator can admit many traffic flows at the same time, we believe that the ”one traffic flow at a time” scenario is very important for the following three reasons. First, in many relevant applications, traffic flows are admitted for a pre-specified duration. The starting times and due dates of the flows are usually independent. Thus, when a new traffic flow has to be admitted, previous traffic flows already use their own paths. Second, an operator may decide to set up a new set of paths between two nodes in order to respond better to periodic congestion. This is an on-line decision, which is captured by the ”one traffic flow at a time” approach. Third, when a link or a node fails, each path that crosses this link or node has to be re-routed, which is again an on-line problem. We therefore conclude that the handling one flow a time is very relevant. Future work should study the impact of admitting multiple flows.

Policy-based Routing Enhancements (Section 2.5.4)

While using overlay routing to improve network performance was studied in the past by many practical and theoretical works, very few of them consider the cost associated with the deployment of overlay infrastructure. We addressed this fundamental problem, developing an approximation algorithm. Rather than considering a customize algorithm for a specific application or scenario, we suggested a general framework that fits a large set of overlay applications. Considering three different practical scenarios, we evaluated the performance of the algorithm, showing that in practice the algorithm provides close to optimal results. Many issues are left for further research. One interesting direction is an analytical study of the vertex cut used in the algorithm. It would be interesting to find properties of the underlay and overlay routing that assure a bound on the size of the cut. It would also be interesting to study the performance of our framework for other routing scenarios, and to study the issue of an actual implementation of the scheme. In particular the connection between the cost in terms of overlay nodes and the performance achieved due to the improved routing is not trivial and it is interesting to investigate it in the different use cases.

OConS DTN service for content sharing with NetInf (Section 3.2.2)

OConS-DTN service results from the combination of applying the HURRy protocol together with BPQ extension over DTN topologies to support content sharing. On the one hand, the HURRy routing protocol takes a forwarding decision making use of a subset of human-related properties that characterise the expected connectivity opportunities to be established among people gathered in an EwLC scenario. On the second hand, OConS support for NetInf comprises the Orchestration of this service by selecting and combining both mechanisms (BPQ extension block and HURRy protocol) when available in each single node. In [2] a representation of how DTN nodes act as NetInf caches and process BPQ requests and responses was depicted. OConS-DTN service has been implemented for a proof-of-concept validation, and the open source project has been published under [36]. The implementation details of this prototype are described in [3], whereas some evaluation results of HURRy performance have been reported in 3.2.2.

4.3 Assessment of OConS Demonstrations and Prototypes

The practical evaluation work in OConS started with the design and realization of a first experimental project phase, driven by early demonstration and experimentation activities of the partners accompanying the OConS technical framework definition, which resulted in intermediate demonstration show cases. These results were presented at the project-internal workshop in January 2012 (project month 18).

Based on these components and the gained experience, D.C.3 [48] described a further step of OConS prototype and demonstration activities which focused in second phase (project month 19-30) on the updated use case scenarios (from D.C.1-Addendum [49]), the work package internal cooperation based on the progress of the OConS architectural framework, and the cross-WP cooperation with CloNe and NetInf.

The final state of OConS prototyping activities as described in D.C.5 [3] comprises:

- OConS reusable components for architecture and orchestration, such as the Generic OConS Protocol Library;
- OConS support for CloNe with the datacentre interconnectivity, elastic networking features based on performance monitoring, path and flow management; distributed dynamic mobility with tunnelling Management
- OConS support for Netinf with the multi-path content delivery and the DTN service for retrieving multimedia content with NetInf.

One of the main objectives of the prototyping activities which were carried out during the SAIL project was to assess the feasibility of the proposed framework and mechanisms, and their contribution to the overall project-wide scenario of 'flash crowds'. The reader might refer to [3] for a more in-depth discussion of the various activities which were undertaken and their main results. Below we highlight the most relevant aspects.

One demonstration [4] particularly focused on the OConS architecture, assessing the feasibility of the orchestration procedure. It included the barebones of the OConS architecture ([2]), specifically the signalling protocols, the INC, the SOP and the OR. In addition, it integrated two illustrative mechanisms which were used so as to show the potential, which OConS, in general, and orchestration, in particular, might bring about.

Using a multi-homed flow management scheme, the demonstration showed how a specific DE (in this case belonging to the *Enhanced Access Selection* mechanism, see Section 2.2.1) was automatically notified upon the registration of any entity which might bring benefits to the mechanism. Furthermore, the demonstration also showed the orchestration of two different mechanisms, which albeit originally conceived to work in isolation, were smartly combined so as to offer a better service to the end-user.

Thus, as a proof-of-concept, the *Enhanced Access Selection* mechanism works jointly with the *Dynamic Distributed Mobility* mechanism to provide an OConS service; moreover, this dynamic tunnelling mechanism can be used by other decision entities from any other mechanism in order to achieve the IP-session continuity for a given flow. From a prototyping point of view, the development and the experimentation work on the mobility execution has allowed us to achieve a better understanding of the basic functional entities (i.e., EE), as well as to validate our approach and to demonstrate its feasibility.

The second scenario was grouped around the distributed datacentre connectivity as required to support the network service needs of CloNe. The combination of mechanisms comprised of the consistent creation of virtual networks within and between distributed datacentres as the network resources of CloNe FNSs. The realization tested the feasibility of an OpenFlow-based approach in an SDN-like set-up. We could show that the CloNe interface definitions for requesting the connecting network resources (encoded in OCNI format) are indeed matching the requirements for the OConS northbound Application Programming Interface (API) as defined in D.C.2 [2]. Both OConS and

CloNe use a common RESTful approach for the interface, which facilitate easy communication at the control and management levels, either by means of a Graphical User Interface (GUI) or a machine interface emulating this mimic by HTML GET and PUT messages. Further discussions of the CloNe-OConS relations can be found in D.A.3 [33], chapter 5.

The realization of an OConS domain controller (DCU) with the capability of orchestrating multiple network mechanisms is described in [16]. The principal functions of network and link state discovery, advertising and monitoring (IE functionality) were combined with intra and inter-domain path and flow computation (DE functionality), establishment and maintenance (EE functionality). The orchestration via the flexible Open Service Gateway Initiative (OSGi) SW framework in the OConS domain controller (typical SOP functionality) was able to register and integrate different path computation mechanisms (PCE alike) and store the network state in the TED.

For the 'elastic networking' functionality of OConS, it turned out that the 'delegation' principle of the resource management in the CloNe distributed cloud layer/manager had to be extended with a callback capability to the CloNe fault management, in order to close the feedback loop when the OConS-managed network is no longer able to provide the requested network and computing resources autonomously between the involved attachment points.

The other demonstration scenario dealt with the interaction of OConS and NetInf. The relationship between NetInf and OConS appears to be straightforward. OConS is, essentially, just a packet transport mechanism. However NetInf is, by virtue of the convergence layer concept, able to use specific packet transport mechanisms of different capabilities for the operation of NetInf over OConS. At the top end of a NetInf convergence layer, a service access point for the NetInf protocol has to be realised (for details, refer to Section 2.2 of D.B.3 as the up-to-date version). At the bottom end, in order to have OConS and NetInf complement each other, all that is needed is to develop a corresponding convergence layer.

OConS Multi-path Network of Information (OMPNetInf) mechanism extends the NetInf architecture to provide efficient multi-path retrieval of content through the use of the OConS Framework; it introduces a number of forwarding strategies (mechanisms) to retrieve content: splitting, replication and distribution. OMPNetInf utilizes the OConS Framework elements (such as orchestration to perform discovery and selection of OConS mechanisms and the coordination of operations) to select and use the appropriate multi-path strategy by means of Information Elements (IE), collecting and providing information for the multi-path strategy, Decision Elements (DE) selecting and configuring of the forwarding strategy and the Execution Elements (EE) enforcing the decisions made by DEs.

A first solution has been demonstrated in the OConS prototyping activity. It turned out that the OConS multi-path capable Convergence Layer (CL) performs the task of identifying an appropriate forwarding strategy to retrieve the content required by the NetInf based applications. The CL uses chunk based request pipelining to request and retrieve the required content. By obtaining information related to the quality of the paths under the control of this CL, it was able to control how the requests for content are distributed to these different paths.

Further CL variations exploiting other OConS capabilities (e.g., in the wireless domain) seem possible and not hard to develop. However, that would not really add too many scientific insights, and hence, we contended ourselves with this proof of concept.

5 Conclusion

This deliverable discusses the evaluation of OConS results and the application for them. Together with D.C.2 [2] and D.C.5 [3], we reach the end of the OConS research and development activities within the SAIL project.

Regarding the applications of OConS results, we elaborated on how OConS provides connectivity solutions to CloNe and NetInf. We also showed how the challenging flash crowd scenario can benefit from the OConS approach. We assessed the OConS results, evaluating the OConS architectural framework, giving examples of the OConS architectures for the framework, and comparing our OConS architectural framework with other architectures under consideration for future networks. Finally, we evaluated the demonstration and prototyping activities that were reported in D.C.5 [3].

Clearly, a lot of significant collaborative work and research results have been achieved. We are proud of these achievements, but acknowledge that the research in the OConS-related topic is by no means complete. We have pursued the separation between the control and forwarding planes, taking it to a challenging and promising arena beyond what SDN, OpenFlow, and PCE can offer, by supporting any network functionality in an abstracted and modular manner. We hope that we and other professionals will have the opportunity to further pursue these ideas, in order to eventually mature them into standards and commercial products and deployments.

A Detailed Results

This Annex describes in more detail the different mechanisms referenced in this deliverable, together with their results and evaluation not reported elsewhere.

A.1 Address resolution mechanism for distributed data centers (DDC-ARM)

Introduction. The OConS mechanism DDC-ARM presented here has been briefly introduced in Section 2.3.2. For the description of the mechanism, we refer to D.C.2 [2] Annex A.8 .

DDC-ARM manages the control traffic caused by address resolution for interconnected data centers. An analytical model for the ARP traffic between data center locations is introduced that takes into account the number of hosts and connected sites. This model can then be used to quantify the ARP traffic for a data center interconnect solution, e.g. based on VPLS.

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 VPLS switches.

Further details of the proposed mechanism and of its performance evaluation model have been given in [17].

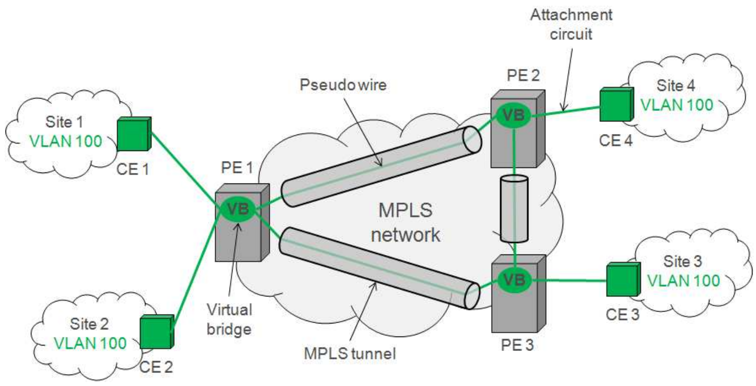


Figure A.1: VPLS architecture scenario as OConS domain

DDC-ARM consists of several procedural phases:

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

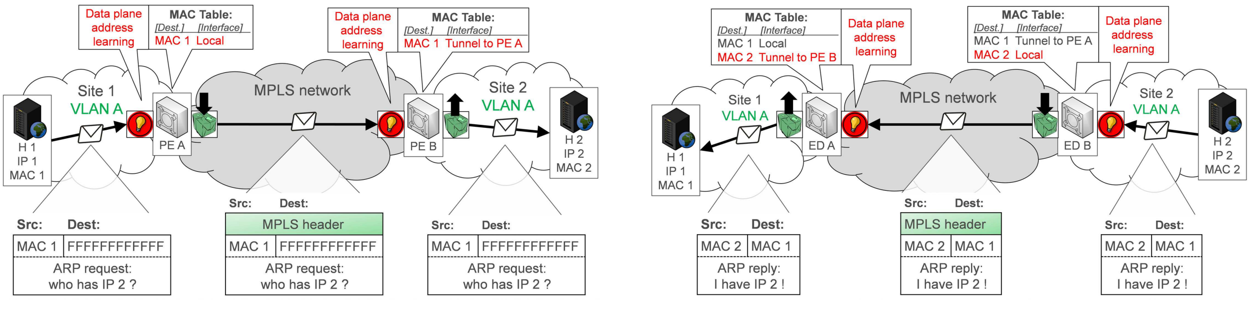


Figure A.2: OconS - VPLS scenario: (a) flooding and address learning, (b) packet forwarding

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.

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 [50].

Performance evaluation. For the performance evaluation, we first study the ARP handling for VPLS as explained before to quantify the amount of total address resolution traffic for a VPLS switch and the transmitted ARP requests per VPLS switch.

We assume a geographically dispersed data center with D locations and a maximum number of N connected nodes per entire VLAN. For the numerical evaluations, N is set to 10000. This results in N/D nodes per data center location. Furthermore, we suppose that each node initiates flows to random destinations which leads to a certain rate of outgoing flows per node. This workload model is of course simple and it neglects many details about the complex load distribution mechanisms inside a data center, but it is difficult to provide a better model that is still generally applicable.

Figure 2 in [51] confirms that a random workload scenario is a reasonable assumption. We vary the rate of outgoing flows per node between 10^{-4} and 10^{+4} flows per second, in order to consider a wide range of possible load situations. Regarding the flow duration, we assume short flows that are

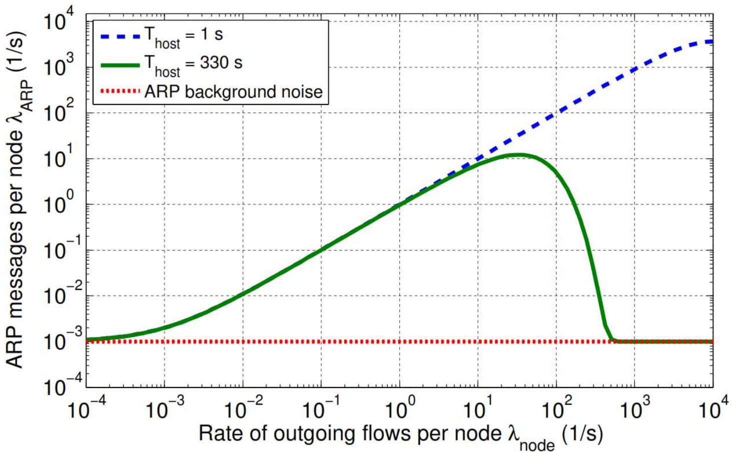
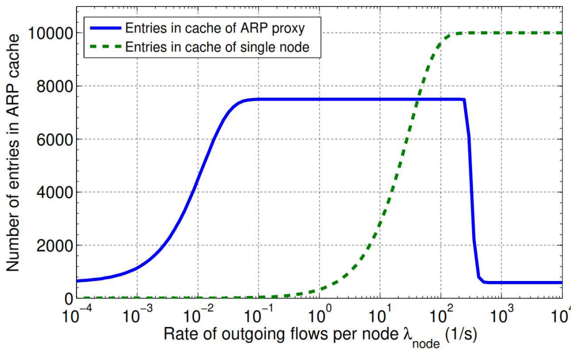


Figure A.3: Rate of outgoing ARP messages per node

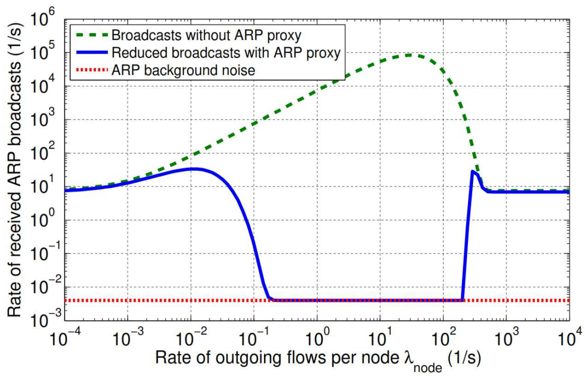
at least one order of magnitude smaller than the cache timeouts in hosts. This assumption is valid because according to [52], 99% of the flows in a data center transmit less than 100 MB, which results in a flow duration in the order of seconds. The cache timeout depends on the configuration but a typical value is 300 s. For connection-oriented protocols like TCP, the above described behavioor results in a timeout for cache entries of about 330 s.

In the following, we assume a benign environment in which ARP traffic is mainly triggered by the steady-state communication between the nodes inside the data center. We thereby neglect specific situations such as broadcast storms caused by failures, which are more difficult to model.

We model the ARP rate per node λ_{ARP} dependent on the rate of outgoing flows per node λ_{node}. We assume that the time between two consecutive flows a node initiates towards another node is exponentially distributed with rate λ_{node}. According to [53], this is a reasonable approximation for the traffic within a data center. As there are $N-1$ other nodes in the VLAN, the considered node contacts a specific other node with a rate of $\lambda_{node}/(N-1)$. If the node does not find an entry in the ARP cache, it broadcasts an ARP request in the VLAN. This happens if the node initiated no other flow to that same destination for more than T_{host} time.



(a)



(b)

Figure A.4: (a) Number of entries inside ARP proxy cache and server ARP cache.
(b) Rate of received ARP broadcasts with and without proxy ARP mechanism.

In Figure A.3, we plot λ_{ARP} against the rate of outgoing flows λ_{node} for $T_{host} = 330s$ and $T_{host} = 1s$.

The ARP rate per node increases about linearly with the rate of outgoing flows per node λ_{node} for values below $\lambda_{node} < 10$ flows per second. It is almost identical because in that range, the node finds almost never a matching entry in its ARP cache as entries are deleted from the cache before the same destination is contacted again. For larger rates of outgoing flows per node, the ARP rate depends on the value for the timeout interval T_{host} . For $T_{host} = 330s$, the rate of ARP broadcasts rapidly decreases for rates λ_{node} larger than 10 flows per second. The smaller the caching duration T_{host} , the later the ARP rate drops to the base rate.

Figure A.4 (a) shows the number of entries in an ARP proxy cache, depending on the rate of outgoing flows per node: we plot the number of mapping entries inside the proxy ARP cache for $D = 4$ domains. As a reference, we also depict the number of entries in the ARP cache of one individual node.

For a low load, i.e., for rates λ_{node} smaller than 0.1 flows per seconds, the number of entries inside the ARP proxy cache slowly increase until they reach the maximum of $N/D * (D - 1)$ entries. In this parameter range, the increasing rate λ_{node} leads to an increasing rate λ_{dest} at the ARP proxy which then leads to an increasing number of cache entries in the ARP proxy.

For a medium load, i.e., for rates λ_{node} larger than 1 but smaller than 100 flows per second, the number of cache entries inside the node cache increase because of the increasing rate λ_{node}. Hence, the number of sent ARP broadcasts per node decreases and the cache entries for external nodes shift from the ARP proxy cache to the node cache. For a high load, i.e., for rates λ_{node} larger than 100 flows per second, the ARP caches inside the nodes nearly contain all possible N destination addresses and hence no ARP broadcasts are sent anymore. The small number of cache entries for rates larger than 100 flows per second come from the assumed ARP background noise lbase.

Result: ARP Broadcast Reduction with ARP Proxy. Now, we apply the model for the ARP proxy to the receiving rate of incoming ARP requests at a VPLS edge switch.

The outgoing rate of ARP requests sent from internal nodes cannot be reduced with an ARP proxy because these messages originate within the local domain and always arrive at the VPLS switch. Hence in the following, these messages are not shown. The ARP proxy of a site broadcasts an ARP request to the other sites only if it cannot find a matching entry for that request in its ARP cache.

In Figure A.4 (b), we plot the receiving rate with and without ARP proxy mechanism for $D = 4$ domains. The dashed line shows the received rate without ARP proxy and the solid line shows the received rate with ARP proxy. Again, we assume some kind of ARP background noise per domain. According to Figure A.4 (b), there is a significant reduction of the ARP traffic for values of l between 0.01 and 100 flows per second.

In this interval, nearly all external destination mappings are stored in the ARP cache and hence no inter-domain ARP broadcasts are necessary, except for ARP background traffic. For lower and higher rates, not all entries are cached in the ARP proxy and hence the probability to broadcast an ARP request increases. For these rates, the ARP proxy is not as efficient as for the interval between 0.01 and 100 flows per second.

In summary, our analytical model confirms that an ARP proxy is an effective method to reduce the ARP traffic between data center sites, and that it may make sense to deploy such proxies in VPLS edge switches.

Summary. We introduced an OConS mechanism for the network layer address resolution in connected distributed data centers, and we investigated the scalability of this mechanism in terms of the resource consumption re. to cache sizes and message rates for broadcast

The evaluation was based on an analytical model for the ARP traffic between data center locations that takes the number of nodes and connected sites into account. As an application scenario, this model was used to quantify the amount of ARP request and ARP reply messages received at a VPLS switch. 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 could be adapted also for other data center interconnect solutions or for ARP traffic prediction within a single data center.

A.2 Network Coding supported by social metrics in DTN

A practical experiment was carried out by using 86 Bluetooth-based wearable devices in an office-like environment during 27 non-consecutive days. It should be emphasized that this particular setup fosters seasonality and periodicity in the historical encounters between different nodes, which are expected to be reflected in the results for the above metrics. Indeed, this information will lay the basis for the improved probabilistic RLNC scheme proposed in this research line.

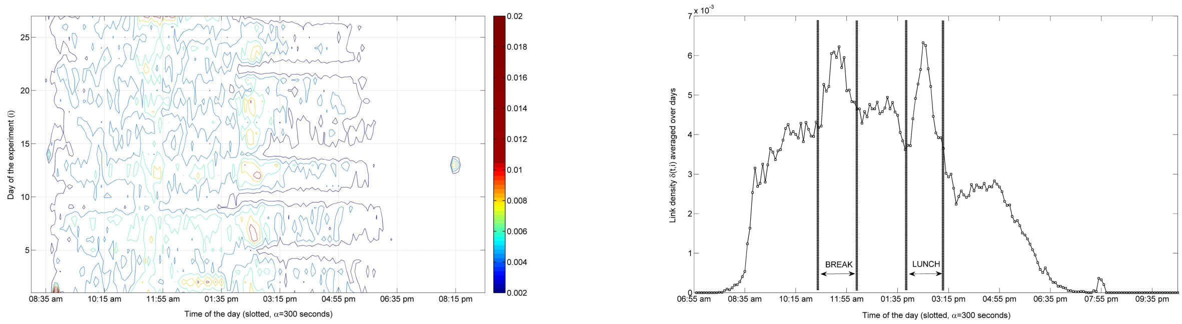


Figure A.5: (a) Link density of the experiment; (b) Link density averaged over days of the experiment.

To begin with, we first focus on the link density of the graph depicted in Figures A.5.a and A.5.b, which have been computed once the connection traces have been windowed in slots of $\alpha = 300$ seconds. On the one hand, Figure A.5.a depicts $\delta(t, i)$ in a contour plot, where it is checked that the activity 1) follows a regular pattern later investigated in Figure A.5.b; and 2) traces recorded on Fridays must be tackled with care, since the work schedule in the considered setup do not consider Friday evenings. On the other hand, in Figure A.5.b $\delta(t, i)$ has been averaged over i to illustrate the overall connectivity behaviour of the experiment through the different times of the day. Clearly, noticeable peaks appear in the vicinity of coffee and lunch breaks; at these times a given node (individual) connects to many other nodes, which is successfully assessed by these peaks. Also interesting is to note that in the afternoon the link density decreases with respect to the morning, possibly due to a reduced mobility of the nodes after lunch time.

We now turn the scope onto the analysis of the closeness, betweenness and eigenvector centrality of the experiment.The scores under consideration for the adaptive schemes have merely been: the closeness centrality $\mu(t, i, n)$ and the betweenness centrality $\rho(t, i, n)$, i.e. the higher the centrality score of a given node at a given time instant in a given day is, the higher the probability of forwarding a given – encoded or not – packet will be. It should be emphasized that the proposed approach is probabilistic, i.e. it may occur – with low probability, though – that a packet is sent to the neighboring node characterized by the worst (least) value of the score. In order to ease the understanding of the results, the closeness centrality metric $\mu(t, i, n)$ has been averaged over i (i.e. the 27 days of the experiment). This permits to identify whether there are potentially nodes in the experiment whose distance to the other nodes becomes particularly low at some times of the day. The rationale behind this analysis is to effectively check that forwarding network-encoded packets to

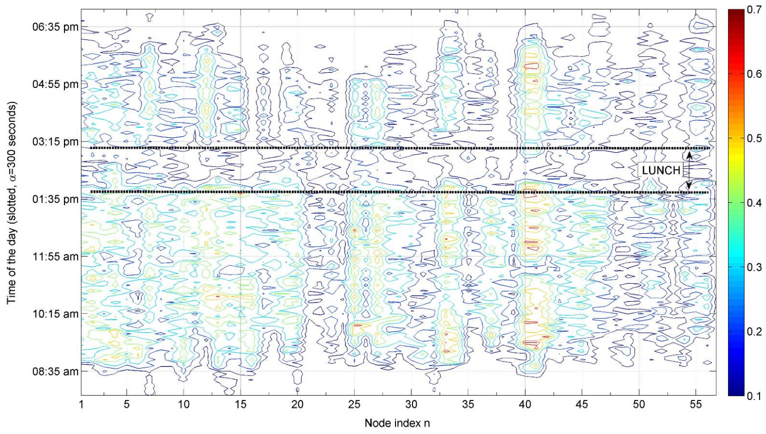


Figure A.6: Contour plot of the closeness centrality $\mu(t, i, n)$ averaged over the day index variable i.

these "central" nodes makes sense a priori based on expected asymmetries in the centrality scores of all nodes. In this regard, Figure A.6 depicts the contour plot corresponding to the closeness centrality of the 86 nodes with respect to the time of the day. First it is of utmost importance to notice that some nodes (e.g. 25, 40 and 41) are particularly central (in terms of $\mu(t, i, n)$ for $n \in \{25, 40, 41\}$) in the morning hours, as opposed to other nodes (e.g. 22, 23, 49 or 50) whose centrality – and thus, their importance in the network flow dynamics – is relatively low. In this plot is also interesting to observe that surprisingly the overall closeness score of the compounding nodes of the experiment decreases dramatically at lunch time. The reason being that whereas the centrality score increases significantly by virtue of the gathering of the node at hand with some other nodes in a small lunch room, the value of the score becomes biased due to the long shortest path between the node at hand and other nodes not coinciding physically in the same place. Side computations not shown for the sake of brevity have verified this fact by removing these "outliers" from the computation of $\mu(t, i, n)$.

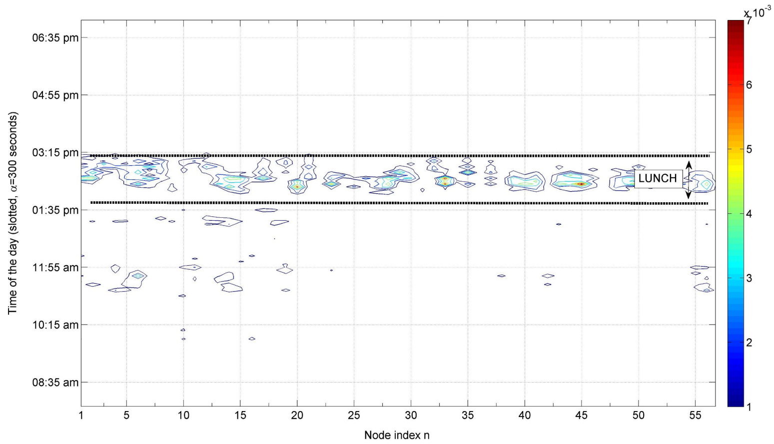


Figure A.7: Countour plot of the betweenness centrality $\rho(t, i, n)$ averaged over the day index variable i.

A close look at the results of the betweenness centrality $\rho(t, i, n)$ (Figure A.7) reveals that at lunch time, some nodes act as "connection bridges" for all shortest paths existing in the network. These nodes may act as central network "buffers" (not necessarily closely connected to other nodes, but participating in relevant paths through the network) to which to forward network-encoded packets. This metric is deemed to be of special importance in challenged environments, where connectivity may be scarce and delay-permissible optimised packet forwarding schemes are to be derived. Similar plots and conclusions hold for the eigenvector centrality score $\zeta(t, i, n)$, hence in what follows we will omit this indicator for clarity.

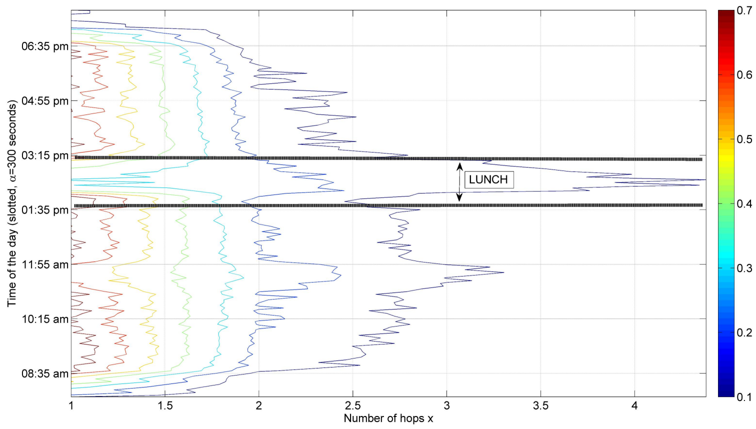


Figure A.8: Contour plot of the distance distribution $\mathcal{D}(t, i, x)$ averaged over the day index variable i.

Finally, we briefly comment on the results for the distance distribution $\mathcal{D}(t, i, x)$, which is depicted as a contour plot in Figure A.8. Notice that again the results have been averaged over the days of the experiment for a better understanding. Observe that the results elucidate a clear tail increase of this distribution at break and lunch times, in clear connection with the above rationale for the surprising decrease of the closeness centrality $\mu(t, i, n)$ at these times.

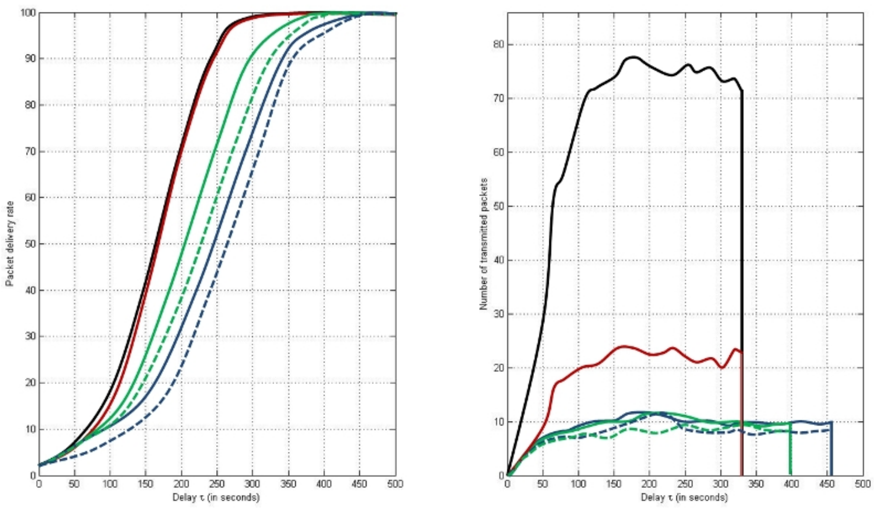


Figure A.9: (a) Packet delivery rate; (b) number of transmitted packets averaged over the days of the experiment when instantiated at 10:00 am. Black solid lines correspond to naive ER; red solid lines to naive RLNC; green solid lines to closeness-based adaptive RLNC; green dashed lines to closeness-based adaptive ER; blue solid line to betweenness-based adaptive RLNC; and blue dashed lines to betweenness-based adaptive ER.

Figure A.9(left) depicts the delivery ratio versus the transmission delay τ for the aforementioned 100 original packets sent at a value of t_s equivalent to 10:00 a.m. Note that the plot has been averaged over the day index i. First observe that the difference between naive ER and naive RLNC is minimal, mainly due to the fact that in absence of bandwidth constraints, simply flooding the network results to be optimal to speed up the successful broadcasting of all the original source packets in the network. RLNC also attains a optimal dissemination of this information through the network provided that complexity considerations are overridden. However, as show in Figure A.9 (right) the number of transmitted packets through the network is decreased with respect to its ER counterpart. As for the adaptive version of these network communication mechanisms, the performance gap widens with respect to the corresponding naive approach when the betweenness centrality is selected to trigger transmissions, the reason being that at this time of the day the value of $\rho(t, i, n)$ when averaged over i results to be extremely uniform (see Figure A.7). Therefore, the roulette-wheel selection of the destination for every forwarded packet degrades the delivery rate performance dramatically. On the contrary, when the closeness centrality is used instead, a performance improvement is noticed mainly due to the existence of "more sociable" nodes in the network; this information is more valuable for the adaptive probabilistic forwarding mechanism. Observe that this trend is detected in both ER and RLNC mechanism, with a slightly better overall performance increase in the case of RLNC. As intuitively expected, the consumed network bandwidth in terms of transmitted packets per transmission delay decreases for the adaptive schemes, not that significantly due to the fact that the number of connections between peers is relatively low at this time of the day.

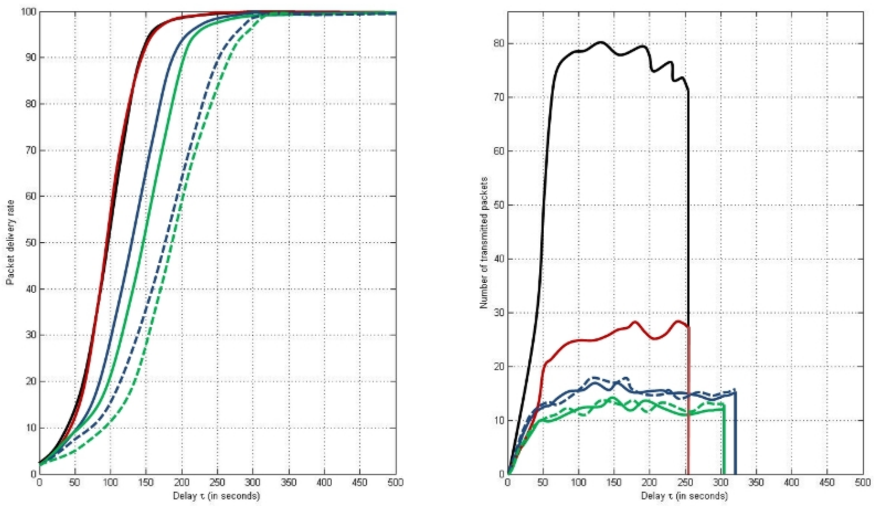


Figure A.10: (a) Packet delivery rate; (b) number of transmitted packets averaged over the days of the experiment when instantiated at 2:15 p.m. Again, black solid lines correspond to naive ER; red solid lines to naive RLNC; green solid lines to closeness-based adaptive RLNC; green dashed lines to closeness-based adaptive ER; blue solid line to betweenness-based adaptive RLNC; and blue dashed lines to betweenness-based adaptive ER

On the other hand, when the time tick for the transmission of the original source packets is changed to 2:15 p.m. (lunch time), the results depicted in Figures A.10 (left) and A.10 (right) go all the way around: the relative behavior between naive schemes keeps the same (with a remarkable boost of the delivery rate due to the higher number of interconnections between peers), but in this second set of simulations the betweenness centrality results to be a better triggering option for any of the proposed adaptive schemes. The rationale can be found again in Figures A.6 and A.7, where it is seen that at this time of the day the betweenness indicator reflects much more clearly the asymmetry of the social network dynamics between the compounding nodes of the network. However, the gap in performance between both scores $\mu(t,i,n)$ and $\rho(t,i,n)$ narrows with respect to the previous experiment, possibly due to the fact that the effectiveness of the betweenness centrality score is defined roughly on the integrity of the shortest paths between nodes during successive relay transmissions of a given packet, which cannot be guaranteed given the particular assumptions (one packet transmitted per node with a probabilistic choice of its destination) of the analysed adaptive network schemes.

In conclusion, the closeness centrality has been found to be an effective network metric candidate to trigger both uncoded (epidemic) and network-encoded packet transmissions in real networks, specially when dealing with applications where the delivery rate is not deemed critical, but where the transmission opportunities result to be very scarce due to challenging communication constraints.

A.3 Spectrum sensing supporting access selection

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.

Applying an energy detector as spectrum sensing technique, a single sensing node has a weak performance under low SNR conditions (SNR=-10 deciBel (dB)). According to figure A.11, a single sensing node presents an optimum point of detection probability (P_d) below the 70% (for a probability of false alarm, P_fa , below the 30%), which is totally unacceptable.

The resulting figures A.12 and A.13 show the performance of several data fusion techniques applied. In one hand, hard decision techniques (AND, OR and Voting decision rules), which are interesting from an implementation point of view due to their low complexity. In the other hand, a soft decision technique based on Linear Combining fusion rule. The improvement of Linear Combining compared to Hard Decision rules is significantly better under AWGN channel conditions A.11. Nevertheless, for a more realistic Rayleigh or Shadowing propagation scenarios the performance difference between Soft and Hard decision rules (above all, OR and Voting) is severely reduced. For example, under Shadowing conditions, A.13 shows an identical performance for Linear Combining and OR decision rules.

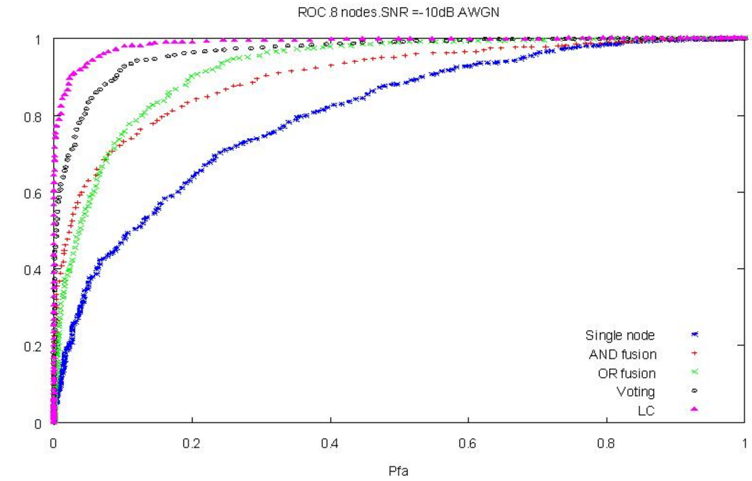


Figure A.11: ROC curves for 8-node cooperative spectrum sensing. AWGN channel

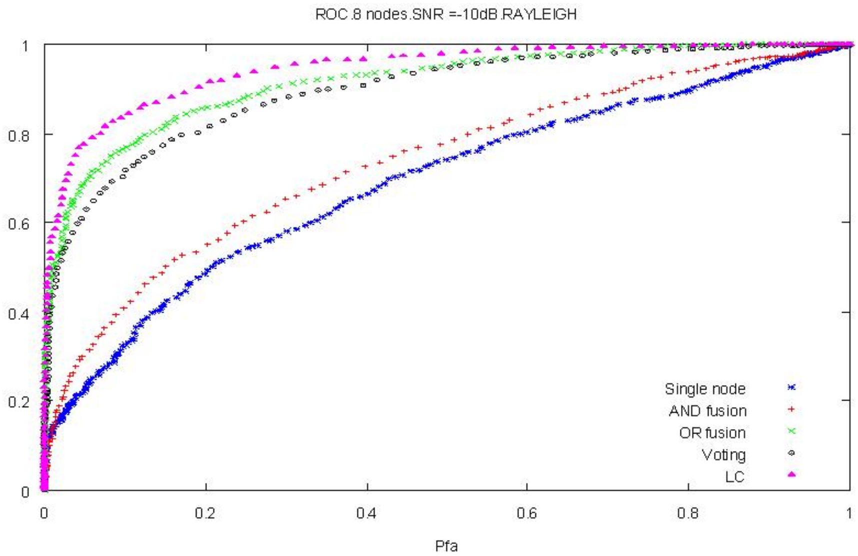


Figure A.12: ROC curves for 8-node cooperative spectrum sensing. Rayleigh flat fading channel

Both techniques achieve a P_d above the 95% and a P_fa below the 0,05%, which is considered as acceptable for Dynamic Spectrum Access architectures. As conclusion, although Soft Combining techniques are optimum for cooperative spectrum sensing schemes, flat fading channels (both Rayleigh and Shadowing-type distributions) provide an advantage to Hard Decision techniques, which could suppose a much lower implementation costs than Soft Decision ones.

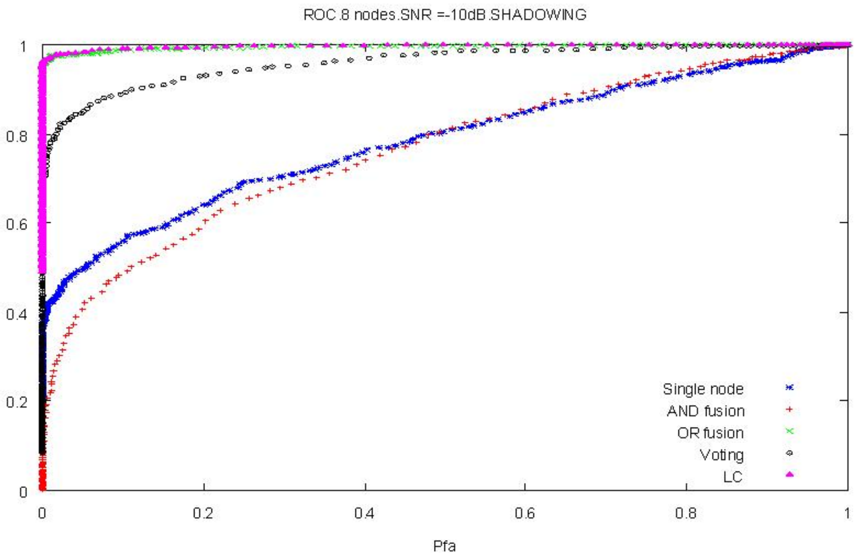


Figure A.13: ROC curves for 8-node cooperative spectrum sensing. Slow flat fading (Shadowing)

List of Abbreviations, Acronyms, and Definitions

3GPP 3rd Generation Partnership Project

API Application Programming Interface

ARP Address Resolution Protocol

AS Autonomous Systems

AWGN Additive White Gaussian Noise

BE Best Effort

BGP-4 Border Gateway Protocol

BPQ Bundle Protocol Query

BS Base Station

BRPC Backward Recursive PCE basedComputation

CC Cloud Controller

CE Customer Edge

CloNe Cloud Networking

CL Convergence Layer

CQI Channel Quality Information

CSI Channel State Information

dB deciBel

DC Datacentre

DCM Distributed Cloud Manager

DCP Distributed Cloud (Control) Plane

DCP-LNP Distributed Cloud Plane - Link Negotiation Protocol

DDC-ARM Distributed Datacentre Address Resolution Mechanism

DDC-WIM Distributed Datacentre WAN Interconnectivity Mechanism

DCU Domain Control Unit

DE Decision Making Entity

DTN Delay Tolerant Networking

EE Execution and Enforcement Entity

eNB evolved Node B

ER Epidemic Routing

ETSI European Technical Standards Institute

EwLC Event with Large Crowd

FER Frame Error Rate

FERA Fair and Efficient Resources Allocation mechanism

FNS Flash Network Slice

GMPLS Generalized Multi-Protocol Label Switching

GRT Guaranteed

GUI Graphical User Interface

HSPA High-Speed Packet Access

HURRy HUman Routines optimise Routing

ICN Information Centric Networking

IE Information Management Entity

IETF Internet Engineering Task Force

INC Intra-/Inter- Node Communication

ISI Infrastructure Service Interface

LAN Local Area Network

LSP Label Switched Path

LSP Link State Packet

LTE Long Term Evolution

MAC Media Access Control

MAP Mesh Access Point

mCASE multi-Constraint Access Selection in heterogeneous Environments

MCS Modulation and Coding Scheme

MPLS Multi-Protocol Label Switching

MS Mobile Station

NC Network Coding

NEP Nash Equilibrium Point

NetInf	Network of Information
NfV	Network Function Virtualisation
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
ONF	Open Networking Foundation
OR	Orchestration Registry
OSAP	Orchestration Service Access Point
OSGi	Open Service Gateway Initiative
OWROS	Opportunistic WMN Resources Management OConS service
PCE	Path Computation Element
PCEP	PCE protocol
PDPC	Per Domain Path Computation
PE	Provider Edge
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
pyOCNI	Python OCNI
QoE	Quality of Experience
QoS	Quality of Service
RAT	Radio Access Technology
RLNC	Random Linear Network Coding
ROC	Receiver Operating Characteristic
RRM	Radio Resource Management
RTT	round-trip time
SACK	selective acknowledgements
SAIL	Scalable and Adaptive Internet Solutions
SDN	Software Defined Networking
SNR	Signal-to-noise ratio

SOP Service Orchestration Process

SOP Service Orchestration Process

SRA Single Router Abstraction

SuperWMC Super Wireless Mesh Client

TCP Transmission Control Protocol

TED Traffic Engineering Database

VBS Virtual Base Station

VM Virtual Machine

VNet Virtual Network

VPLS Virtual Private LAN Services

VRRA VNet Radio Resource Allocation

WAN Wide Area Network

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless LAN

WMC Wireless Mesh Client

WMN Wireless Mesh Network

WMR Wireless Mesh Router

WP work package

List of Figures

List of Tables

References

[1] SAIL. The SAIL project web site. `http://www.sail-project.eu/`.

[2] SAIL. Architecture and Mechanisms for Connectivity Services. Deliverable FP7-ICT-2009-5-257448-SAIL/D.C.2, SAIL project, February 2013. Available online from http://www.sail-project.eu.

[3] 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.

[4] Luis Diez, Olivier Mehani, Lucian Suciu, and Ramón Agüero. Design and implementation of the open connectivity services frameqork. In *MONAMI 2012, 4th International Conference on Mobile Networks and Management*, 2012.

[5] Johnny Choque, Ramon Aguero, and Luis Munoz. Simulation framework for the evaluation of access selection algorithms over heterogeneous wireless networks. In *MONAMI 2012, 4th International Conference on Mobile Networks and Management*, 2011.

[6] L. Zhao, Y. Zaki, A. Udugama, U. Toseef, C. Görg, and A. Timm-Giel. Open connectivity services for future networks. In *8th International Conference Expo on Emerging Technologies for a Smarter World (CEWIT)*, pages 1–4, November 2011.

[7] U. Toseef, Y. Zaki, L. Zhao, A. Timm-Giel, and C. Görg. QoS aware multi-homing in integrated 3GPP and non-3GPP future networks. In *ICSNC 2012, 7th International Conference on Systems and Networks Communications*, Red Hook, NY, USA, November 2012. IARIA, Curran Associates.

[8] U. Toseef, Y. Zaki, C. Görg, and A. Timm-Giel. Uplink QoS aware multi-homing in integrated 3GPP and non-3GPP future networks. In *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. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[9] U. Toseef, Y. Zaki, A. Timm-Giel, and C. Görg. Optimized flow management using linear programming in integrated heterogeneous networks. In *ICSNC 2012, 7th International Conference on Systems and Networks Communications*, Red Hook, NY, USA, November 2012. IARIA, Curran Associates.

[10] 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 *MONAMI 2012, 4th International Conference on Mobile Networks and Management*, 2012.

[11] 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.

[12] 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).

[13] Yasir Zaki, Thushara Weerawardane, Carmelita Görg, and Andreas Timm-Giel. Long term evolution (LTE) model development within OPNET simulation environment. In *OPNETWORK 2011*, Bethesda, MD, USA, August 2011. OPNET Technologies, OPNET Technologies.

[14] Umar Toseef, Yasir Zaki, Andreas Timm-Giel, and Carmelita Görg. Uplink QoS aware multihoming 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.

[15] Johnny Choque, Ramón Agüero, Joan Serrat, and Luis Muñoz. A price based load balancing scheme for multi-operator wireless access networks. In *MONAMI 2012, 4th International Conference on Mobile Networks and Management*, 2012.

[16] 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.

[17] 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.

[18] Pierrick Louin and Philippe Bertin. Network and host based distributed mobility. In *WPMC2011: Wireless Personal Multimedia Communications*, Brest, France, October 2011.

[19] F.D. Cardoso L. Caeiro and L.M. Correia. Adaptive allocation of virtual radio resources over heterogeneous wireless networks. In *EW 2012, 18th Europeen Wireless Conference*, April 2012.

[20] F.D. Cardoso L. Caeiro and L.M. Correia. Ocons supported on demand radio resource allocation for virtual connectivity. In *MONAMI 2012, 4th International Conference on Mobile Networks and Management*, September 2012.

[21] 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.

[22] R. Aguero, L. Caeiro, L.M. Correia, L.S. Ferreira, M. Garcïœa-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.

[23] 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.

[24] 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.

[25] 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.

[26] L.S. Ferreira and L.M. Correia. Service for management of connectivity in spontaneous community-based wireless mesh networks. In *MONAMI 2012: 4th International Conference on Mobile Networks and Management*, Sep. 2012.

[27] 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.

[28] 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.

[29] 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.

[30] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In *In Proceedings of MASCOTS*, 2001.

[31] N. Spring R. Mahajan and D. Wetherall. Measuring isp topologies with rocketfuel. In *In Proceedings of the ACM SIGCOMM*, August 2002.

[32] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972.

[33] SAIL. Final Harmonised SAIL Architecture. Deliverable FP7-ICT-2009-5-257448-SAIL/D.A.3, SAIL project, February 2013. Available online from http://www.sail-project.eu.

[34] SAIL. Refined CloNe Architecture. Deliverable FP7-ICT-2009-5-257448-SAIL/D.D.3, SAIL project, October 2012. Available online from `http://www.sail-project.eu`.

[35] D. Kutcher S. Farrell, A. Lynch and A. Lindgren. Bundle protocol query extension block. Active Internet-Draft, May 2012.

[36] Telecom Business Area. Android implementation of hurry routing protocol and bpq extension block for dtn2 taking the bytewalla project as basis, 2013.

[37] Aalto University. The Opportunistic Network Environment (ONE) simulator, (Last accessed July 2012). `http://www.netlab.tkk.fi/tutkimus/dtn/theone/`.

[38] David Gómez, Sofiane Hassayoun, Arnaldo Herrero, Ramón Agüero, and David Ros. Impact of Network Coding on TCP performance in wireless mesh networks. In *IEEE PIMRC 2012: 23rd IEEE Symposium on Personal, Indoor, Mobile and Radio Communications*, 2012.

[39] L.S. Ferreira, M.D. de Amorim, L. Iannone, L. Berlemann, and L.M. Correia. Opportunistic management of spontaneous and heterogeneous wireless mesh networks [accepted from open call]. *Wireless Communications, IEEE*, 17(2):41 –46, april 2010.

[40] Opnet modeler wireless suite.

[41] Jennifer Rexford. Programming languages for programmable networks. *SIGPLAN Not.*, 47(1):215–216, January 2012.

[42] Nate Foster, Rob Harrison, Michael J. Freedman, Christopher Monsanto, Jennifer Rexford, Alec Story, and David Walker. Frenetic: a network programming language. In *Proceedings of the 16th ACM SIGPLAN international conference on Functional programming*, ICFP '11, pages 279–291, New York, NY, USA, 2011. ACM.

[43] Teemu Koponen, Scott Shenker, Hari Balakrishnan, Nick Feamster, Igor Ganichev, Ali Ghodsi, P. Brighten Godfrey, Nick McKeown, Guru Parulkar, Barath Raghavan, Jennifer Rexford, Somaya Arianfar, and Dmitriy Kuptsov. Architecting for innovation. *SIGCOMM Comput. Commun. Rev.*, 41:24–36, 2011.

[44] ETSI. Industry Specification Group on Network Functions Virtualisation (NfV). First meeting held in January 2013.

[45] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia. Fast Handovers for Proxy Mobile IPv6. RFC 5949(Standard), 2010.

[46] 3GPP. 3GPP Technical Specification Group Services and System Aspects - Network Sharing; Architecture and functional description *Release*9. Technical Report TS 23.251, 3GPP, March 2010.

[47] Johnny Choque, Ramon Aguero, and Luis Munoz. Optimum selection of access networks within heterogeneous wireless environments based on linear programming techniques. Aug 2011.

[48] 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.

[49] 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.

[50] K. Elmeleegy and A.L. Cox. Etherproxy: Scaling ethernet by suppressing broadcast traffic. In *INFOCOM 2009, IEEE*, pages 1584 –1592, april 2009.

[51] Srikanth Kandula, Sudipta Sengupta, Albert Greenberg, Parveen Patel, and Ronnie Chaiken. The nature of data center traffic: measurements & analysis. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 202–208, New York, NY, USA, 2009. ACM.

[52] Albert G. Greenberg, James R. Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta. Vl2: a scalable and flexible data center network. In Pablo Rodriguez, Ernst W. Biersack, Konstantina Papagiannaki, and Luigi Rizzo, editors, *SIGCOMM*, pages 51–62. ACM, 2009.

[53] Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. Data center tcp (dctcp). *SIGCOMM Comput. Commun. Rev.*, 40(4):63–74, August 2010.